



UNIVERSITÀ DEGLI STUDI DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

DOTTORATO DI RICERCA IN  
INGEGNERIA ELETTRONICA E DELLE TELECOMUNICAZIONI  
CICLO XX

EFFICIENT DATA DISSEMINATION PROTOCOLS  
IN PERVASIVE WIRELESS NETWORKS

**Direttore** Ch.mo Prof. Silvano Pupolin

**Supervisore** Andrea Zanella

**Dottorando** Elena Fasolo

Padova, 31 Gennaio 2008



*To Massimo*



## ACKNOWLEDGMENTS

The Ph.D. and the material collected in this thesis represent for me a lot of experiences and emotions which enriched my life in these last years. Most of this would not have been possible without the very special people that I met in my way and encouraged me to never give up.

Thanks to Andrea Zanella, that suggested me to take up with the Ph.D. and forced me to believe in my capabilities. Thanks to Michele Rossi: he taught me to be precise, professional and to always try to overcome my limits. Thanks to Michele Zorzi who wisely directed most of my research activities. Thanks to Joerg Widmer for accepting me at NTT DoCoMo EuroLabs in Munich: he gave me the opportunity to work within his lively research group. What I learned is due to these clever and kind men and to the fruitful discussions we had. They showed me what the actual research is and in my work I have done my best to try to imitate them.

Thanks to all the guys of the SIGNET group and the colleagues I knew at NTT DoCoMo EuroLabs. They have been unique travel mates to share joys and pains, happiness and strain, in a word: life. Thanks to Anna, Simone and Paolo; we started this adventure together: now we will go on different directions but they will be always in my heart.

Thanks to who supported me in these years, especially to my family which has most of the merit of my achievements. Thanks to all the new friends I knew within these offices at the department, they make my life glad and lighten the difficult work of a Ph.D. student. Thanks to Petra: when I was in Munich I felt at home in her house and she made my staying there delightful. Thanks to Michael because he showed me a lot of Munich, the life of the german Ph.D. student and he is still teaching me German.

Finally, thanks to Massimo: the most amazing finding of my Ph.D. Thanks simply for everything.

Thanks a lot.



# Contents

<b>Sommario</b>	V
<b>Abstract</b>	VII
<b>1 Introduction</b>	1
1.1 Structure of the Thesis . . . . .	3
<b>2 Efficient Data Dissemination in Wireless Networks</b>	7
2.1 The Broadcast Storm Problem . . . . .	7
2.2 Related Work . . . . .	10
2.2.1 Flooding . . . . .	10
2.2.2 The MCDS-based Scheme . . . . .	10
2.2.3 Probability-based Schemes . . . . .	11
2.2.4 Position-based Schemes . . . . .	12
2.2.5 Neighbor-based Schemes . . . . .	13
2.2.6 Performance Comparison . . . . .	13
2.2.7 Cluster-based Schemes . . . . .	14
2.2.8 Epidemic Protocols . . . . .	15
2.3 Two Case Studies . . . . .	18
<b>I Efficient Data Dissemination in Inter-vehicular Networks</b>	21
<b>3 Dissemination of Alert Messages in Inter-vehicular Networks</b>	23
<b>4 Position-based Data Dissemination Protocols</b>	25
4.1 Urban Multi-hop Broadcast Protocol (UMBP) . . . . .	25
4.2 Geographic Random Forwarding (GeRaF) . . . . .	27

<b>5</b>	<b>Smart Broadcast Protocol</b>	<b>29</b>
5.1	Reference Scenario . . . . .	29
5.2	The Smart Broadcast Protocol Description . . . . .	31
5.3	Theoretical Analysis . . . . .	33
5.3.1	One-hop Latency . . . . .	33
5.3.2	One-hop Message Progress . . . . .	35
5.3.3	Message Propagation Speed . . . . .	36
5.4	Optimal Parameters Setting . . . . .	36
5.5	Validation of the Theoretical Analysis . . . . .	37
5.6	Performance Metrics, Simulations and Results . . . . .	40
5.7	Robustness Analysis of Smart Broadcast Protocol . . . . .	44
5.7.1	Validity Interval . . . . .	45
5.7.2	Optimal Contention Windows Size on Average . . . . .	46
5.7.3	Performance Evaluation . . . . .	47
5.8	Discussion and Conclusions . . . . .	50
<b>II</b>	<b>Efficient Data Dissemination via Network Coding</b>	<b>51</b>
<b>6</b>	<b>Data Dissemination via Network Coding</b>	<b>53</b>
<b>7</b>	<b>Network Coding Background</b>	<b>55</b>
7.1	Linear Network Coding . . . . .	58
7.2	Random Network Coding . . . . .	60
7.2.1	Data Dissemination via Network Coding . . . . .	61
7.3	More Complex Approaches . . . . .	62
<b>8</b>	<b>MAC Layers and Packet Combination Strategies on Network Coding Data Dissemination</b>	<b>65</b>
8.1	Problem Description . . . . .	66
8.2	MAC Protocols . . . . .	67
8.2.1	IEEE 802.11b . . . . .	68
8.2.2	IEEE 802.11b with Pseudo Broadcast [1] . . . . .	68
8.2.3	IEEE 802.11 with Pseudo Broadcast and RTS/CTS Handshaking . . . . .	68
8.2.4	Ideal MAC . . . . .	68
8.3	Network Coding Strategies . . . . .	69
8.3.1	Probabilistic Network Coding . . . . .	69



---

8.3.2	Semi-deterministic Network Coding . . . . .	69
8.3.3	Timed Network Coding . . . . .	70
8.4	Simulation Results . . . . .	70
8.4.1	The Impact of MAC Protocols . . . . .	71
8.4.2	Different Packet Combination Strategies . . . . .	76
8.5	Analysis of Adaptive Packet Combination Strategies . . . . .	81
8.5.1	Adaptive Send-count Approaches . . . . .	81
A	Performance Evaluation . . . . .	84
B	Observations . . . . .	85
8.6	Innovation-based Network Coding . . . . .	88
8.6.1	Description of the Scheme . . . . .	88
8.6.2	Simulation Results . . . . .	89
8.6.3	Observations . . . . .	92
8.7	Conclusions . . . . .	94
<b>9</b>	<b>Proactive Network Coding</b> . . . . .	<b>95</b>
9.1	Proactive Network Coding (ProNC) . . . . .	95
9.1.1	Basic Rules for ProNC . . . . .	96
9.1.2	Stopping Conditions . . . . .	97
9.1.3	Rate Adaptation Heuristic . . . . .	100
9.1.4	Implementation Notes . . . . .	101
9.2	Simulation Results . . . . .	101
9.2.1	Reference Scenario and Performance Metrics . . . . .	102
9.2.2	Evaluation of Strong and Weak Stopping Policies . . . . .	102
9.2.3	A Possible Rate Adaptation Heuristic . . . . .	106
9.2.4	ProNC vs Reactive Network Coding Schemes . . . . .	108
9.3	Discussions and Conclusions . . . . .	109
<b>10</b>	<b>The MIMO_NC Approach</b> . . . . .	<b>111</b>
10.1	Description of MIMO_NC . . . . .	113
10.2	Performance Analysis . . . . .	116
10.2.1	Classical NC Performance . . . . .	117
10.2.2	MIMO_NC Performance . . . . .	117
10.3	Simulation Results . . . . .	120
10.4	From MIMO_NC to Super MIMO_NC . . . . .	123

---

10.5 Super MIMO_NC . . . . .	123
10.5.1 Performance Analysis . . . . .	125
10.5.2 Adaptive MIMO_NC . . . . .	128
10.6 Performance Results . . . . .	129
10.6.1 Star Topology . . . . .	129
10.6.2 Butterfly Topology . . . . .	131
10.7 Conclusions . . . . .	132
<b>11 Conclusions</b>	<b>135</b>
<b>12 Author's Publications</b>	<b>139</b>
<b>Bibliography</b>	<b>141</b>

## SOMMARIO

Quando parliamo di disseminazione di informazioni intendiamo quel processo per cui una grande quantità di dati viene distribuita a tutti (o molti) utenti di una rete. Le specifiche caratteristiche del sistema in uso possono rendere questo obiettivo complesso e difficile. Sviluppare algoritmi per la disseminazione *efficiente* di informazioni in ambienti wireless ad hoc, ad esempio, è tuttora un problema aperto a causa delle caratteristiche del mezzo di comunicazione e perchè tutte le trasmissioni devono essere gestite in modo distribuito. Da un lato, il canale wireless comporta una notevole quantità di problemi legati alla contesa del mezzo, alla gestione delle collisioni e dell'interferenza. Quando la quantità di dati da trasmettere e il numero di nodi coinvolti nelle comunicazioni sono elevati, questi aspetti sono ulteriormente accentuati. Dall'altro lato, i protocolli dovrebbero utilizzare solo informazioni locali facilmente reperibili per poter rendere la rete scalabile e robusta alla mobilità.

Questi sono solo alcuni esempi delle difficoltà che si incontrano nel progettare meccanismi per la distribuzione di dati in una rete wireless. L'obiettivo di questa tesi è investigare questi e altri aspetti al fine di sviluppare algoritmi di disseminazione di dati che siano efficienti. Naturalmente un algoritmo è efficiente in relazione agli obiettivi per cui è stato progettato. In generale, i requisiti richiesti a questo tipo di schemi sono affidabilità, bassa latenza, ridotto consumo energetico, robustezza, limitata complessità computazionale e così via.

Tuttavia, il problema della disseminazione di dati, così definito, è ancora troppo vasto e generale per poter progettare schemi che abbiano anche una valenza pratica. Per questo motivo, in questa tesi, ci concentreremo su due casi studio. Definiremo due scenari applicativi che ci permetteranno di mettere in luce le problematiche legate al problema della disseminazione. Nella prima parte della tesi ci occuperemo di *disseminazione di messaggi di allerta in reti inter-veicolari* mentre nella seconda parte tratteremo la *disseminazione di dati per mezzo del network coding*. Questi scenari sono stati scelti perchè da un lato sono sufficientemente specifici da poter definire le condizioni iniziali e gli obiettivi del problema in modo preciso. Dall'altro lato sono sufficientemente generali da consentire che le soluzioni proposte possano essere applicate in contesti diversi. Lo studio di questi scenari ci darà l'opportunità di comprendere a fondo il problema della disseminazione di dati in reti wireless distribuite.



## ABSTRACT

Data Dissemination consists on spreading a large amount of information to all nodes belonging to a network. The peculiar characteristics of the system in use make this goal particularly interesting and challenging. Developing efficient data dissemination schemes for wireless ad hoc networks, for instance, is still an open issue due to the broadcast nature of the channel and to the need of managing all data transmissions in a distributed way. The former leads to a lot of problems related to the channel contention, collisions and interference. The latter requires to define algorithms which exploit only local information of the network and which are scalable and robust to the node mobility.

The focus of this thesis is to investigate such wireless ad hoc networks by defining and developing data dissemination schemes which can be efficient. The efficiency of an algorithm mainly depends on the requirements imposed by the application scenario of that scheme. In general, they can be reliability, low latency, limited energy consumption and computational complexity and so on.

Thus, the problem of efficiently disseminate data, as defined right now, is too wide and general. For these reasons, in this thesis, we will focus on two case studies. We will define two application scenarios in order to point out all the peculiarities and issues related to the data dissemination. In the first part we will focus on *dissemination of alert messages in inter-vehicular networks* while in the second part we will deal with the *data dissemination problem in pervasive systems*. We choose these two scenarios as they are specific, i.e., we can precisely define the initial requirements, constraints and objective. But they are also general, i.e., the solutions we will find, could be implemented in different contexts. Thus, the analysis of such case studies will give us a wide and detailed view of the data dissemination problem.



# 1

## Introduction

---

*Well begun is half done.*  
(Aristotle)

In this thesis we deal with the problem of data dissemination in wireless distributed systems. Our main aims are two. First, we would like to understand the more interesting and challenging problems related to this topic, the basic mechanisms which regulate data dissemination schemes, the major requirements of such protocols and so on. Second, we want to propose practical and feasible solutions to disseminate data in realistic environments.

The concept of data dissemination is wide and meaningful. In this context,

we refer to data dissemination whenever there is some amount of data which has to be spread over a wireless distributed network.

The data can be generated by a single node or by different sources, the destinations usually are many (even all nodes belonging to the networks) and they are interested in retrieving all or a part of the generated information. We observe that the concept of data dissemination can be applied at different layers of the protocol stack and it can be useful for different purposes.

Looking at the network layer, data dissemination schemes can be used to spread routing data such as *Hello* messages or topology information. ARP messages are usually disseminated all over the network to associate each IP address to the corresponding MAC address for each device. Normally, this kind of data dissemination is referred to as broadcast or multicast.<sup>1</sup>

At the application layer, we can be interested in transmitting the same amount of data to multiple users. As examples, we can consider the file sharing applications, the broadcasting of multimedia files and so on. In sensor networks, data dissemination is applied whenever the sink node has to query the network to gather some useful information from the sensors. Moreover, some attention has been recently devoted to new challenging scenarios where different technologies meet together to offer new services [2]. It is the case of the so called *Internet of Things* where heterogeneous systems offer the possibility to gather a lot of information stored in different devices, to find objects, to control actuators and so on. In this scenarios a lot of users interact with each other by exchanging a great amount of information.

---

<sup>1</sup>Data dissemination schemes and broadcast protocols are used interchangeably in this thesis.

Thus, referring to all the possible applications of data dissemination, we can identify several requirements that good protocols should satisfy.

- *Reliability*: the protocol has to guarantee that all nodes interested in collecting the information successfully receive all the packets. This could be more important for some applications such as the broadcasting of alert/disaster/hazard messages, the spreading of information vital for the existence and maintenance of the network, and so on. On the contrary, reliability could be not required when the data to be spread represents extra information such as the distribution of advertisement messages.
- *Time-constraints*: the scheme has to respect some constraints on the packet delivery delay. In some situations, data dissemination has to be performed as fast as possible because the information to be spread is particularly important. Real time applications represent only one of the most interesting cases. In other cases, instead, the service is delay tolerant and data dissemination scheme has no constraint on the delivery time as in the case of file downloading.
- *Feasibility*: the data dissemination scheme has to be feasible, i.e., implementable on practical networks. Thus, it has to deal with specific constraints such as limited resources, heterogeneous devices, limited computational capabilities and so on.
- *Energy-Efficiency*: there are a lot of specific networks formed by devices with limited energy resources. Wireless Sensor Networks (WSN) are only one example but also PDA networks and, in general, networks formed by battery-based devices are included in this class. Data transmission is particularly expensive and it has to be limited to guarantee long lifetime to such networks. Thus, the data dissemination scheme has to be developed in order to save energy resources. This could be achieved in different ways such as reducing the number of transmissions required to deliver data, implementing duty cycles on nodes, and so on. In other situations, where network has unlimited resources, these aspects can be of minor importance.
- *Bandwidth-Efficiency*: usually, in a network, different types of traffic coexist. They can be generated by different applications or also by different network layers. In addition, data dissemination schemes can be very expensive in terms of bandwidth as they involve a lot of devices and multiple transmissions. Thus, to limit the impact of data dissemination over other applications is needed that the protocol to spread data saves network resources as much as possible. It is the case, for instance, of the data dissemination schemes which spread control, topology or maintenance information. They represent a background network traffic which constantly limit the bandwidth available for the applications. Reducing as much as possible the number of transmissions required to deliver data could be also in these situations a good strategy to guarantee better performance.



It is usually very hard to achieve all the goals at the same time and by the use of the same strategy. While there are common features which all data dissemination schemes have to implement, others are particularly related to a specific application. Thus, a lot of different algorithms have been proposed in literature to efficiently disseminate data in a wireless network with a major focus on one of the different characteristics. In the rest of the thesis we also focus on specific data dissemination applications by defining two reference scenarios. Our main aim is to point out and solve the issues related to *Time-Constrained* and *Energy-Efficient* applications because we believe that they represent the most challenging scenarios for future network architectures. However, the scenarios we define are sufficiently general such that the solutions we develop can be applied also in different contexts.

Finally, we mention here that the data dissemination in wireless environments leads to a lot of challenging problems due to the nature of the medium in use and to the specific features of the transmission protocols usually applied for wireless communications, i.e., interference, collisions, contention, random access mechanisms and so on. In addition, we need to take into account that, in general, all the operations in wireless networks have to be carried out in a distributed way without coordination/synchronization among nodes. Indeed, each form of centralized management of the nodes' transmissions is usually expensive in terms of network resources and it can strongly reduce the system performance. On the other hand, the same applications can be easier to offer in a wired context where a centralized management of the nodes' transmissions is usually easier to implement. For all these reasons, the wireless scenario represents a challenging and interesting environment to develop efficient data dissemination schemes. In particular, we focus on all those systems that use the CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) paradigm as the basic transmission strategy because it is the most applied in wireless networks. A detailed description of this kind of approach is beyond the scope of this thesis. For the reader who intends to obtain a deeper knowledge into this field, we suggest to see [3–5].

## 1.1 Structure of the Thesis

To give some insights on this research field we organize the topics of the present thesis in two *Parts* consisting of different *Chapters*. Each Part discusses a particular topic within the thesis subject and the Chapters explain in detail some aspects about the topic under investigation. We briefly indicate here the contents of each Part and Chapter.

**Chapter 2** introduces the problem of efficiently disseminate data in wireless networks by presenting the existing solutions and by introducing our motivations for deeper researches on this areas. In addition, we present here the two case studies we take as reference scenarios in this thesis. We decide

to focus on realistic applications to give a more practical direction to our investigations.

## **Part I – Efficient Data Dissemination in Inter-Vehicular Networks**

In this Part we focus on a specific practical application of data dissemination schemes related to the delivery of alert messages in inter-vehicular networks. These kind of systems are gaining a lot of interest among the scientific community as they involve a lot of new applications. In addition, their specific features lead to challenging research activities. Defining a data dissemination scheme in this context is particularly interesting as the peculiar characteristics of the messages to be spread require tight time constraints. Thus, this practical application gives us the motivations to investigate data dissemination schemes which are efficient in terms of packet delivery delay.

**Chapter 3** introduces the inter-vehicular scenario, the motivations and the main issues related to the data dissemination in this context.

**Chapter 4** reviews the most interesting solutions proposed to disseminate data in such networks. We present here the details of the two protocols we consider to compare our algorithms.

**Chapter 5** is focused on the development of an efficient data dissemination scheme for inter-vehicular networks. The name of such scheme is *Smart Broadcast Protocol (SB)* and it is based on the minimization of the time required to deliver data to all the interested nodes. In this Chapter, after the protocol description, we present both the theoretical analysis of its performance and an extensive simulation campaign. We evaluate different optimization aspects of our proposal and we compare SB with other existing schemes.

## **Part II- Efficient Data Dissemination via Network Coding**

In Part II, we move towards a different topic related to the efficient data dissemination in pervasive systems. This context leads to different protocol requirements. In particular, we focus on such applications based on the utilization of devices with limited resources. The main aim here is to develop data dissemination schemes that are efficient in terms of energy resources, bandwidth usage and number of required transmissions. This is the case of wireless sensor networks but not only. Also networks with high traffic loads, with limited bandwidth, and so on, can be interested in disseminating data saving network resources. To achieve this goal we use an emerging network paradigm named *network coding*.

**Chapter 6** gives a more detailed vision of the second reference scenario by introducing the main concepts and terminology used in the following Chapters.

**Chapter 7** introduces some background related to the network coding paradigm. It is a recent technique used to increase the throughput of the network. Classical dissemination schemes are based on the *store and forward* approach, where nodes keep the received packets in a buffer and, when it is possible, they forward those messages to the other nodes. Network coding, on the contrary, is based on the *store, code and forward* paradigm where nodes usually forward coded packets instead of simple data. The main peculiarity is that these packets are obtained as linear combination of simple original data. Applying this kind of coding process we can drastically reduce the number of transmissions required to disseminate data with respect to the classical schemes. For these reasons, network coding techniques are applied to reduce the traffic load, to increase the network performance or to save energy resources.

**Chapter 8** discusses practical problems related to the application of network coding in wireless systems. Most of the work done up to now about network coding is theoretical and based on the graph theory. Also proposed practical schemes are mostly analyzed in simplified environments where channel errors, interference, MAC protocols, multi-rate physical layers, and so on, are not considered. Our main contribution is the analysis of network coding performance taking into account all these aspects in more realistic environments. The main outcomes are two. First, we better understand the mechanism of network coding and how this strategy react to the collisions or to different MAC protocols. This clarify aspects that the theoretical analysis can not explain. Second, we find the way to develop a scheme which can efficiently perform also in realistic environments.

**Chapter 9** describes the novel data dissemination scheme we propose to alleviate the effect of realistic environments on network coding performance. It is named *ProNC* and it is based on a proactive approach rather than on the reactive one used up to now in existing solutions. The main advantages of such a scheme are its robustness to packet losses, its reliability and its promptness. We analyze ProNC performance via simulations and compare them with the reactive solutions.

**Chapter 10** proposes a different solution to make network coding more robust to the packet losses. Our focus here is the study of fading environments and we propose a novel approach which jointly combine network coding paradigm and MIMO techniques in order to provide an integrated system. This kind of topic is particularly interesting due to its novelty and originality. We develop a sophisticated communication system which offers the same benefits of network coding but is able to exploit *spatial diversity* as MIMO techniques in order to increase the system reliability. We name such an approach

MIMO\_NC. We test its performance in different scenarios both theoretically and via simulations. We find that MIMO\_NC can represent a promising approach for future radio communication systems and it can improve the performance of data dissemination schemes based on network coding.

**Chapter 11** summarizes the main outcomes of the this thesis by looking at the main topics and the main results obtained.

# 2

## Efficient Data Dissemination in Wireless Networks

---

*The more alternatives,  
the more difficult the choice.*  
(Abbé D'Allanival)

In order to understand how to face the problem of efficiently disseminate data in wireless ad hoc networks we need, first, to point out which are the main features, constrains and problems related with this goal. Thus, in this Chapter, we describe the so called *Broadcast Storm Problem* which defines the main issues related to the data dissemination in a wireless environment. Then, we review the existing literature on this topic pointing out the pro and cons of the different solutions. Finally, we introduce the two case studies we will analyze in detail in the rest of the thesis.

### 2.1 The Broadcast Storm Problem

The main objective of data dissemination schemes can be summarized as follows:

- *Guarantee reliability*: the protocols should aim to guarantee a reliable service. In particular, all the information should be delivered to all the interested nodes. It is clear that, when the destination nodes are several and the network is organized in a multi hop fashion this task could be particularly complex due to the lack of coordination among nodes.
- *Guarantee low latency*: this goal is mainly related to real time application. However, developing protocols which can fast deliver data helps in increase network performance.
- *Save network resources*: data dissemination schemes are in general very expensive in terms of network resources such as bandwidth, energy, and so on. This is due to the fact that a great amount of data has to be transmitted. One of the main aims of these protocols is to limit the usage of network resources hence leading to better performance in terms of throughput or increasing the network lifetime (if we refer, for instance, to networks with limited energy resources).

Unfortunately, in order to achieve these goals we have to face a wide set of problems referred to as the *Broadcast Storm Problem* [6]. They are specially related to the fact that, to reach a lot of nodes

of a network, a great amount of transmissions is required and several nodes usually try to access the channel to send their data. This is particularly emphasized in wireless environments where the channel is usually shared among nodes. In addition, multi-hop scenarios introduce further complexity as data to be delivered has to flow along long paths. To better explain this concept, it is important to remember that all schemes developed to disseminate data are based on the *store and forward* approach and they substantially differ from the unicast or multicast communication paradigms.

In unicast situations, to deliver a packet from a node A to a node B, a routing path is usually established and the packet is forwarded by intermediate nodes until it reaches its destination. In multicast cases, the protocols follow a similar approach, i.e., multiple routing paths are established from the source node to the different destinations.

In a broadcast scenario, where all nodes are both sources and destinations, we should establish a routing path from any node to any other node of the network. It is clear that this might be unfeasible. Thus, in general, data dissemination schemes are based on the assumption that all nodes participate to the distribution of the data. A node usually stores the received packet and, if possible, it forwards it to the interested destinations. The way a node contributes to the dissemination depends on the strategy in use and it is what characterizes the different protocols.

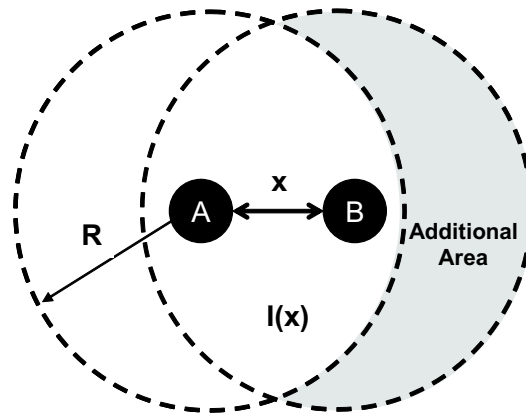
However, as long as all protocols are based on the same approach, they are also affected by the same problems. We can group the main aspects of the Broadcast Storm Problem in three main classes briefly described in the following.

1. *Redundancy*: it occurs whenever nodes transmit unuseful information, i.e., all other nodes already have that data. Due to the broadcast nature of the wireless channel, any transmitted message can be received by a set of nodes in the transmitter's coverage area. Then, each of these receivers could be charged to forward the message thus leading to a lot of redundant retransmissions.

The redundancy level can be quantified by means of the *coverage gain* metric. To better explain this concept we refer to Fig. 2.1 where node A and B decide to forward the same message. Assume that, at first, node A sends the packet and all its neighboring nodes receive it. When node B transmits, its transmission can be useful only for the nodes placed in the gray area of Fig. 2.1. This additional area reached by the transmission of B is given by:

$$\int_0^R \frac{2\pi x[\pi R^2 - I(x)]}{\pi R^2} dx \simeq 0.41\pi R^2 \quad (2.1)$$

where  $I(x)$  is the intersection region of the coverage areas of A and B and  $R$  is the coverage range of any node. When the retransmitting nodes are more than two this gain is further reduced towards 0 thus increasing the redundancy.



**Figure 2.1** Additional area covered by a second retransmissions.

One important goal is the reduction of the redundancy as much as possible by limiting the number of nodes which forward the packets on behalf of other nodes.

2. *Contention:* Data dissemination protocols, usually, produce a high channel contention. A single message is received by several nodes which immediately could try to forward it; this means that they simultaneously try to access the channel in the same area. The consequences are lower performance, i.e., higher packet delivery delay and lower throughput due to the longer backoff procedures usually implemented to solve the channel contention. The reduction of the number of nodes which act as forwarding nodes has some beneficial effects also in this situation as it reduces the number of contending nodes [7].
3. *Collisions:* Data dissemination performance could also be affected by collisions. A collision occurs when two (or more nodes) in the same coverage area simultaneously transmit. This leads to an interference level such high that receiving nodes are not able to successfully receive any packet. Due to the collisions, packets are lost and they need to be retransmitted thus wasting network resources. In addition, retransmitting a packet usually means longer waiting time to access the channel (due to the increasing of the contention window size used to select the backoff time [7]) and consequently high packet delivery delay. As in the previous cases, the problem can be partially solved by allowing to retransmit to a smaller number of nodes.

There are many ways to alleviate the presented problems. The reduction of the number of forwarding nodes is not the only possible solution; also the definition of a more complex MAC layer could increase the data dissemination performance. However, maintaining the protocol as simple as possible has a lot of advantages both in terms of computational resources and network management.

In the next Section we investigate the possible approaches to alleviate the broadcast storm problem.

## 2.2 Related Work

In this Section, we overview the most interesting solutions which better comply with the peculiarities of wireless and even mobile networks.

We start our analysis from two particular schemes, namely the flooding and the MCDS-based protocol. The first one represents the simplest data dissemination scheme, while the second one represents the most efficient. The former is widely used in practice whereas the latter is only a theoretical approach. We introduce them as they represent two benchmarks for all the other schemes.

Between these two boundaries, several broadcast algorithms have been proposed to cut the tradeoff between robustness and redundancy, in particular in the context of wireless ad-hoc networks [8]. All the protocols we consider are based on a CSMA/CA scheme to access the channel and they are classified mainly on the basis of the strategy adopted for electing the forwarding nodes. We can identify five main classes of protocols: (i) *probability-based*, (ii) *location-based*, (iii) *neighbor-based*, (iv) *cluster-based* and (v) *epidemic*.

### 2.2.1 Flooding

The simplest way to spread data to all nodes of a network is the *flooding* approach where each node is charged to retransmit its own message and the packets from all the other nodes that it receives. All the nodes in the coverage area of the transmitter (*neighbor nodes*) can receive the sent packets as in the packet header the destination address is set to  $-1$  (i.e., BROADCAST transmission). This kind of approach has several disadvantages when it is applied in CSMA/CA environments as the broadcast storm problem is particularly evident in the flooding scheme.

First, the flooding scheme introduces a lot of redundancy. As an example, we consider Fig. 2.2 which represents data distribution using flooding. To reach all nodes, 24 transmissions are required but only some of them are really necessary.

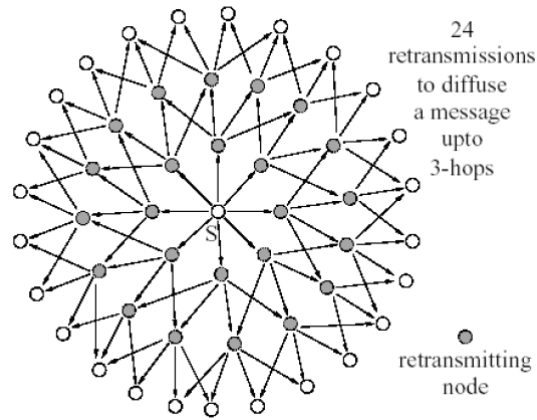
Second, according to the protocol, each node which receives a new packet immediately forwards it. This produces two bad consequences: high contention to access the channel in a specific area and high collision probability.

Thus, transmitting a lot of redundant packets is not only useless but also harmful as most of the transmitted packets are unsuccessfully received by the interested nodes.

### 2.2.2 The MCDS-based Scheme

Theoretically, the data dissemination protocol which guarantees the best performance is the one based on the *Minimum Connected Dominating Set* (MCDS).





**Figure 2.2** Flooding redundancy

MCDS is defined as the minimum cardinality set of connected nodes, such that each other node in the network is connected to a node of the MCDS set.

According to this structure, each message to be disseminated is propagated only by the nodes in the MCDS. A performance analysis of such an ideal algorithm under simplifying hypotheses is presented in [9]. However, the creation and maintenance of the MCDS structure in a distributed network could be not feasible. It needs a centralized entity which selects the nodes belonging to the MCDS, updates the MCDS when nodes move or turn off. In addition, each node needs to know if it belongs or not to the MCDS and achieving this knowledge requires the exchanging of a lot of control messages. Nevertheless, the performance reached by the data dissemination scheme based on the MCDS can be considered as an upper bound for our analysis.

### 2.2.3 Probability-based Schemes

In the algorithms belonging to this category, the forwarding nodes are picked up according to a given probability distribution function.

The simplest case is provided by the algorithm proposed in [6], where a node receiving the broadcast message forwards it with a probability  $p$  while refrains from rebroadcasting with a probability  $1 - p$  ( $p = 1$  means pure flooding scheme). The best selection of parameter  $p$  is delicate and depends on the network scenario. In the remaining, we refer to this scheme as *Probabilistic Flooding*.

Another approach, named *counter-based* scheme [6], relies on the following reasoning: the larger the number of duplicate broadcast messages a node receives, the smaller the additional area covered

by a new broadcast retransmission by that node. According to this, nodes that receive more than  $C_{max}$  copies of the same message in a given time window are prevented from retransmitting it. This mechanism reduces redundancy to the detriment of robustness and promptness in the message propagation. The value of  $C_{max}$  has a strong influence on the algorithm performance. Unfortunately, its optimal setting is strictly dependent on the node density which may undergo rapid fluctuations in mobile scenarios. To alleviate this problem, several schemes propose adaptive strategies to dynamically adjust the threshold value according to variation of the network topology conditions [10].

## 2.2.4 Position-based Schemes

The main advantage of the *position-based* algorithms is that they do not require exchange of topology information, thus reducing control traffic overhead.

One of the simplest position-based schemes, named *location-based*, is introduced in [11]. It reduces the broadcast redundancy by having a node rebroadcast depending on the additional coverage area it provides. In this scheme the additional coverage area is estimated from the location information of the nodes. Therefore each node must determine its own location, e.g. by using Global Positioning System (GPS). Each node adds its location to the header of the packet before broadcasting or rebroadcasting it. Upon receiving a packet a node learns the location of the sender and calculates the additional coverage area provided if it rebroadcasts the packet. If the additional coverage area is less than a threshold value, then the node does not rebroadcast the packet.

Another interesting algorithm in this category is the *Urban Multi-hop Broadcast Protocol* (UMB) [12], designed to disseminate data in urban areas. It is based on a contention mechanism whose aim is to select as relays the nodes belonging to the MCDS. The contending nodes send a black-burst signal, whose duration is proportional to their own distance from the broadcast source. The longer black-burst is transmitted by the furthest node, thus winning the contention and becoming the next broadcast relay node. An improvement is obtained by taking into account both position and movement direction. As we compare some of our solutions with UMB, in Chapter 4 we give a detailed description of its main procedures.

Other two interesting proposals are the *Vector based TRack DEtection* (V-TRADE) and the *History enhanced Vector TRack DEtection* (HV-TRADE) protocols [13]. They are specifically designed to take into account nodes mobility and they require that each node is equipped with a positioning system device. Each node keeps updated information about current and past positions of its neighbors and uses it to predict their future position. When a node has a message to broadcast, it selects the best broadcast relay node from the so called *border nodes* set. The border nodes are neighbor nodes with two features: they are placed in such a way to guarantee the maximum advancement of the broadcast message and

they have never received a copy of the broadcast message. The drawback of such approach is that it is not suitable for resource-limited nodes as it includes the possibility to use very expensive devices such as the GPS.

### 2.2.5 Neighbor-based Schemes

When a *Neighbor-based scheme* is used, nodes base the decision whether rebroadcasting the message or not on the status of their neighbors. For instance, according to *Flooding with Self Pruning* [14], a node  $A$  retransmits a broadcast message only if it can be received by “isolated” nodes, i.e., nodes that are not reachable by any other. More sophisticated methods to elect the relays, aiming at reducing redundancy and increasing reliability, are described in [15, 16]. Such algorithms determine the set of possible relay nodes on the basis of the position of both the one-hop and two-hop neighbors. Other examples are the *Scalable Broadcast Protocol* (SBA) and the *Ad-Hoc Broadcast Protocol* (AHBP). However, in general, such schemes require local topology knowledge and they could worsen performance in case of mobility.

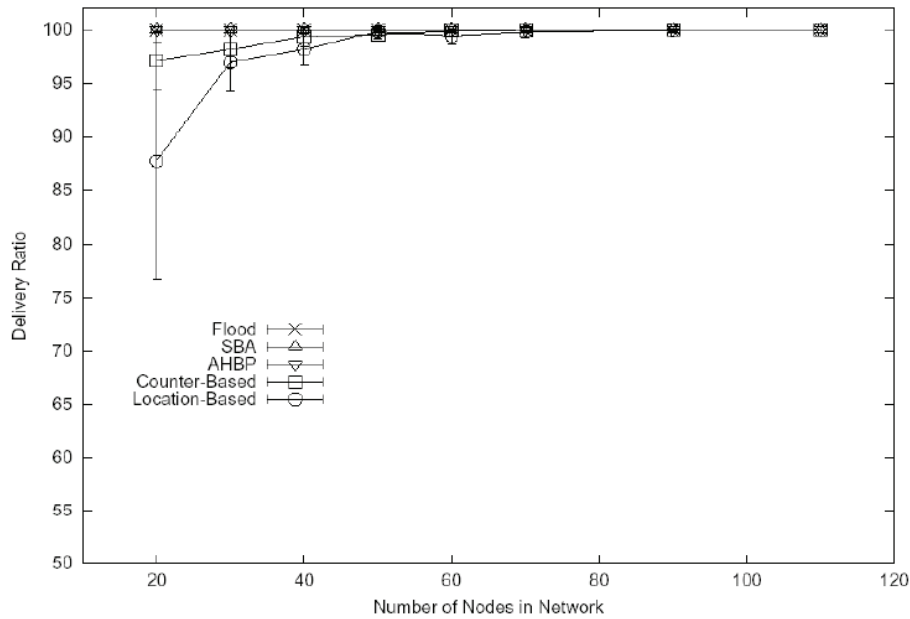
### 2.2.6 Performance Comparison

We report here the results presented in [17]. They compare the performance of different schemes selected as representative of the classes introduced above. These results are related to scenarios where ideal MAC and physical layer are used. This is due to point out only the data dissemination characteristics of the algorithms.

In Fig. 2.3, the packet delivery ratio versus the number of nodes in the network is plotted. We note that basic flooding and the protocols representing neighbor-based methods, namely SBA and AHBP, perform better in sparse networks as they introduce more redundancy. However, in dense network all the schemes achieve the same performance.

Fig. 2.4 shows the number of retransmitting nodes versus the number of nodes. This represents a more interesting metric as it gives an idea of the protocol efficiency. We have seen in Fig. 2.3 that all schemes guarantees similar performance in terms of packet delivery ratio, but they differ on the strategy adopted to achieve that performance.

Protocols that have more complicated algorithms have fewer retransmitting nodes than the others. AHBP approximates theoretical the best-case. The Figure also shows that the location-based scheme has fewer retransmitting nodes than probability-based schemes. The threshold values for the counter-based and location-based schemes are held constant in the graphs of Fig. 2.3 and Fig. 2.4. For the counter-based scheme the threshold is 3, and for the location-based scheme it is 45 meters, but the threshold values affects the results. In the counter-based scheme, a higher threshold value in sparse



**Figure 2.3** Protocols Comparison: Packet Delivery Ratio versus number of nodes.

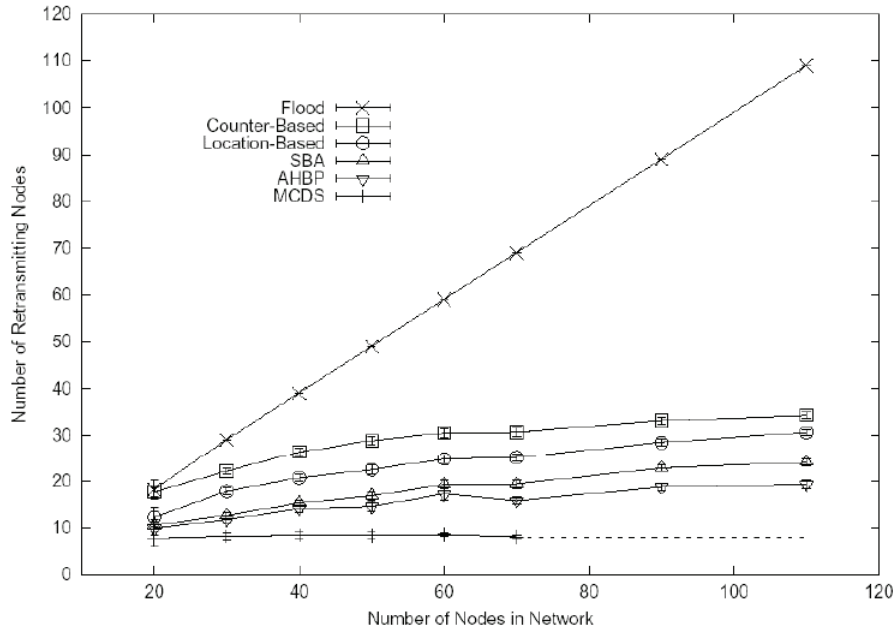
networks and a lower threshold value in dense networks increases the delivery ratio. In location-based scheme, a lower threshold value is used to maintain a high delivery ratio in sparse networks and a higher threshold value is used in dense networks.

Finally, the MCDS schemes achieves as expected the best performance as it can guarantee the full reliability with the lowest redundancy thus saving network resources.

### 2.2.7 Cluster-based Schemes

The *Cluster-based schemes* require the organization of the network in clusters. By exploiting neighbors knowledge, the nodes are grouped into small clusters, each one managed by a particular node, which is elected *cluster-head*. The nodes in the same cluster share some common features, such as relative position or energy level. Once clusters are formed, dissemination is usually performed by entrusting the cluster-heads with the task of retransmitting broadcast messages.

Cluster-based protocols can be divided on the basis of the strategy adopted to elect the cluster-heads. In ad hoc environments, clustering schemes related to mobility are of particular interest. One solution is to group the nodes according to their relative speed, in order to decrease the re-clustering cost. MOBIC technique [18], for instance, selects the nodes with the lowest speed variance as cluster-heads. Similarly, the *Distributed Dynamic Clustering Algorithm* (DDCA) [19] defines the so called  $(\alpha, t)$  rule to build a cluster and to select the cluster-head. A node belongs to a cluster if its probability



**Figure 2.4** Performance Comparison: Number of retransmitting nodes versus total number of nodes.

to reach another node in the cluster in a time  $t$  is larger than  $\alpha$ . Moreover, a node can be added to the cluster if the  $(\alpha, t)$  rule among itself and the cluster-head is satisfied. This guarantees that nodes approximatively close to each other belong to the same cluster.

The schemes proposed in [20, 21] are further cluster-based approaches suitable for mobile networks. They aim at minimizing the clustering maintenance due to re-clustering and re-affiliation. The basic idea is to re-organize the clustering structure only on-demand, when it is really necessary, rather than periodically.

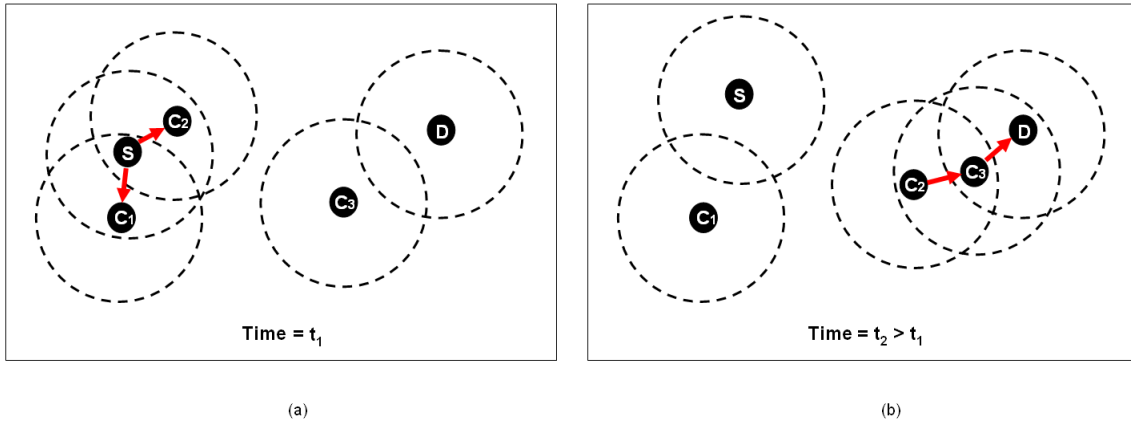
## 2.2.8 Epidemic Protocols

The basic idea of the *Epidemic protocols* is completely different from the other schemes already mentioned (also because they are design for a different purpose). The main aim here is to guarantee a reliable data dissemination scheme also when the networks remain unconnected for an undefined period of time. The basic idea of such approaches is the gossiping. Rather than aiming at the reduction of the number of transmissions they try to increase the robustness of packet delivery and the reliability of the system.

Well known techniques to deal with message delivery without location information are *epidemic* [22] and *probabilistic protocols* [23].

In [22] the authors present an overview of the basic concepts of epidemic routing. Epidemic algorithms have recently gained popularity as effective solutions for disseminating information in large scale distributed systems. In an epidemic algorithm, all nodes are potentially involved in the information dissemination (see Fig. 2.5). Basically, each node buffers every message it receives until it reaches a certain buffer capacity  $b$ . Moreover, each message is forwarded a limited number of times  $t$  and each time to a randomly selected subset of neighboring nodes, whose size is also limited and equal to  $f$ . The parameters  $b$ ,  $t$  and  $f$  are tunable and may affect the performance of the system. Epidemic routing, in its basic version, roughly works as follows: a source initially sends a message to be disseminated in a system of  $n$  nodes. Each infected node (each node that receives a copy of the message) forwards the message to a randomly chosen subset of nodes of size  $O(\log(n))$ . Eventually, the message will reach all the destinations in the system with high probability after  $O(\log(n))$  transmissions. The failure of one or more communication links does not significantly affect the message delivery. This is due to the redundancy which is inherently introduced by the algorithm in forwarding multiple copies of a single message.

The main problem of epidemic routing is that every node that receives a message is entitled to forward it only to already known nodes. The original epidemic algorithm [24], for instance, assumes global knowledge, i.e., every node knows every other node in the network. It is easy to understand that this aspect is in *sharp contrast with the idea of a dynamic and distributed system*. [25] addresses epidemic routing in partially-connected ad hoc networks where a connected path from the source to the destination does not always exist. The main goal of this study is to delivery messages to arbitrary destinations with minimal assumptions regarding the underlying network topology. For instance, one might subdivide the network into two subnetworks, one of which is connected. The connected portion of the network could be subsequently exploited as a backbone to quickly spread the information to all its nodes. Messages are delivered to the nodes in the connected subnetwork first and are finally delivered to the users within the unconnected subnetwork (these are users on the move, which continuously enter and leave the network) as soon as these come in touch with a node belonging to the “backbone”. As in the original epidemic routing, each node maintains a buffer which is used to record the history of the most recently originated/received messages. In addition, to facilitate the exchange of histories between nodes, these are encoded by means of hash tables into *summary vectors*. When two arbitrary nodes are located within the communication range of each other, the node with the lowest identifier initiates the information exchange. The two nodes subsequently exchange their summary vectors and each node may request a copy of the messages that it has not yet seen. The receiving node maintains total autonomy in deciding whether it will accept a message. For example, it may determine that it is unwilling to carry messages larger than a given size or destined to certain nodes. The main drawback of such an approach consists of the high overhead involved in the communication between nodes

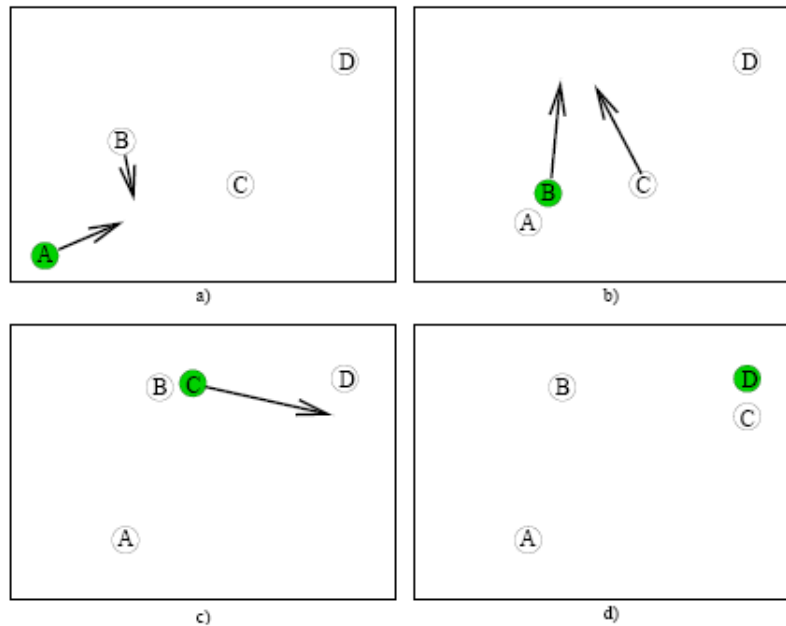


**Figure 2.5** (a) A source (S) wishes to transmit a message to a destination (D) but no connected path is available. It then transmits a copy of the message to its two neighboring nodes C1 and C3. (b) C1-C3 are leveraged to transitively deliver the message to its destination at some later point in time.

(exchange of summary vectors and subsequent handshake for the actual packet transmission).

An alternative to epidemic routing is represented by *probabilistic routing* [23]. Probabilistic and epidemic routing are built on the same rationale: messages may have to be buffered for a certain time at intermediate nodes, whose mobility is exploited to bring messages closer to their destinations by exchanging them between nodes as they come in touch. However, according to probabilistic routing, as two nodes meet, a message is transferred from one node to a second node only if the *delivery predictability* is higher at the second node (Fig. 2.6). The *delivery predictability* is a probabilistic metric defined for any source ( $s$ ) and destination ( $d$ ) pair,  $P_{(s,d)} \in [0, 1]$ , which represents the likelihood that node  $s$  will be able to deliver a message to  $d$ . This metric is exchanged between nodes by means of *summary vectors*, similarly to what discussed previously for epidemic routing.

We finally cite the recent work on optimal gossiping and routing in [26]. The authors of [26] present for the first time the *gossip network model* where travelers can obtain information about the state of dynamic networks by gossiping with peer travelers using ad hoc communication. Travelers can subsequently use the acquired information to recourse their path and find the lowest cost route to their destination. While the paper introduces a rather simple model for the characterization of the network and learning probabilities, its contribution is very valuable for what concerns the concept of optimal routes. In fact, in [26] it is shown that sometimes it is better for a packet to take a detour, which will surely increase the path length but, at the same time, will give a chance to the traveler to acquire additional knowledge about the state of the network. As a consequence, travelers can exploit their new knowledge to refine (and likely improve) their future routing decisions. The routing problem is



**Figure 2.6** Transitive communication. A message (shown in the figure by the node carrying the message being green) is passed from node A to node D via nodes B and C through the mobility of nodes.

formulated according to a dynamic programming approach and the optimal routing policy is devised by accounting for information states and learning probabilities. These are used to model the likelihood to retrieve useful information by taking a detour. Overall, the work is about getting an optimized balance between the path cost and the cost of gathering information. We note that these methodologies may be applied to our problem as well. In the information dissemination phase, for instance, we may decide to take detours in order to disseminate the information more evenly and therefore decrease the acquisition time for future queries. A drawback of the model in [26] is that the network topology is assumed to be known to a large extent and only link weights may be unknown. This, however, does not fit our scenarios where connectivity structures will likely vary in time.

## 2.3 Two Case Studies

The problem of efficiently disseminate data in wireless ad-hoc networks is complex due to the several reasons investigated in the previous Sections. They can be summarized as follows.

- *The channel* - The broadcast medium used for the transmissions leads to a lot of different problems such as collisions, interference, contention. They arise whenever multiple nodes want



to simultaneously access the channel to transmit data. These situations, by nature, occur in data dissemination and their effects increase with specific medium access schemes such as the CSMA/CA.

- *Data Dissemination Goals* - Data dissemination schemes usually have to guarantee performance in terms of packet delivery ratio, latency, energy saving and so on. Achieving all these goals simultaneously is in general complex, especially in case of large networks.
- *Wide Range of Applications* - Data dissemination schemes can be applied in different contexts and for different purposes. Thus, the definition of a general set of initial assumptions, requirements and constraints could be difficult.

Thus, to better define the problem, it could be useful to have in mind some practical situations where the data dissemination schemes can be applied. This practical perspective can give several advantages: i) it facilitates the definition of the initial assumptions and constraints; ii) it specifies the requirements and iii) it helps in the definition of the main objectives. Defining an efficient data dissemination scheme in this more defined context could be easier and it could become of practical utility.

On the contrary, the definition of such a specific scenario could lead to some limitations. In particular, the risk is that the developed data dissemination scheme is too much specific and it can be properly applied only to a specific scenario. To avoid this drawback we need to find some application scenarios such that they are:

- *Realistic* in order to help us in the definition of the problem and to give a practical meaning to our outcomes;
- *General* such that the obtained results can be applied also to different situations with similar requirements or aims.

According to this perspective, in this thesis we focus on two specific case studies, namely *Data Dissemination of alert messages in inter-vehicular networks* and *Data Dissemination in pervasive systems*. They represent two possible interesting applications of data dissemination schemes.

We choose such scenarios as they are quite different in terms of initial assumptions and requirements. Thus, the solutions we find can be applied to a wide range of applications, assuming a general validity. In addition, they give us the motivations to analyze and apply some of the most interesting techniques related to the wireless ad-hoc networks.

Hence, before starting the detailed treatment, we briefly introduce here our two case studies.

1. *Data Dissemination of alert messages in inter-vehicular networks (CARNETs)*. This kind of applications requires strong constraints on the packet delivery delay, so they are the suitable en-

vironments to develop time-efficient data dissemination schemes. In this context, we focus on the definition of a proper MAC protocol capable to optimize the channel contention and the relay selection phase. To speed up the data dissemination we need to maximize the information progress towards the destinations at each retransmission. To achieve this aim, the selection of the retransmitting nodes plays an important role and it is the focus of our analysis.

2. *Data Dissemination in pervasive systems.* Pervasive systems as wireless sensor networks are usually characterized by energy limited devices, thus they represent a good scenario to define and to study energy-efficient data dissemination schemes. As mentioned before, some kind of data processing can be useful to reduce the amount of data to be spread over the network (or they can make the dissemination more efficient). For this reason, we add to the classical protocols some intelligence. The natural choice is the use of the network coding paradigm which is a recent data processing approach applied in order to increase the network throughput. We develop different approaches to apply network coding to the data dissemination problem mainly focusing on practical aspects.

We want to underline that these two reference applications represent for us only two good scenarios to make more concrete our proposals. The algorithms we define can be applied in different contexts and for different purposes but we believe that having in mind some interesting applications simplifies the understanding of the crucial steps of our analysis. Finally, we observe that practical considerations and aspects may contribute to increase the interest on the academical research activities for the industry and pave the way for actual implementations of our ideas.

The rest of the thesis is organized in two parts. In Part I, we focus on the inter-vehicular network scenario by giving all the details of our proposal. In Part II, we consider the data dissemination schemes based on network coding by discussing several approaches that apply this novel paradigm to disseminate data.

Finally, we conclude our dissertation by summarizing and discussing the obtained results and also giving some hints for future researches on these topics.

## **Part I**

# **Efficient Data Dissemination in Inter-vehicular Networks**



# 3

## Dissemination of Alert Messages in Inter-vehicular Networks

---

*Imagination is more important than knowledge...*

(Albert Einstein)

One interesting research trend of recent years regards inter-vehicle communication systems (IVS), also known as Car Networks (CARNETs). These systems are intended for a broad range of applications, including primary services such as emergency notification in cases of accidents, but also more advanced applications as cooperative driving assistance, car-to-car audio/video communications, nomadic Internet access, and so on.

Thus, IVC represents a challenging scenario for ad hoc based communications. On the one hand, the presence of car batteries looses the constraints for energy-aware communications which characterize, for instance, wireless sensor networks. On the other hand, services related to the car-mobility require new design paradigms at most layers of the OSI model. Hence, the peculiarities of the IVC scenario ought to be exploited in order to gain advantages in the design of physical, MAC and routing layer solutions. In particular, such peculiarities include the availability of timing and localization information provided by the Global Positioning System (GPS).

Recently, some European/International projects have started research activities on this topic. Car 2 Car Communication Consortium [27] and SafeSpot Integrated Project [28] are only two of the most interesting ones. They are dedicated to the objective of further increasing road traffic safety and efficiency by means of inter-vehicle communications. In [29] an analysis of the current trends in inter-vehicular networks is drawn.

At the physical layer, the main trends are on using the UTRA-TDD and the IEEE 802.11 standards, though the latter one is getting more and more attention. The research effort has been mainly addressed to the medium access mechanisms, routing protocols, network management strategies and applications provisioning. In [30] an extensive survey of the most interesting MAC and physical layers suitable for inter-vehicular networks is presented. Particular attention is devoted to the IEEE 802.11 family.

However, little attention has been devoted to the design and analysis of efficient and reliable broadcast propagation mechanisms, which are of primary importance in the IVC scenario. Indeed, the impressive social and economical cost of road accidents makes the research of proactive safety services an important task. Thus a fundamental application in this category is the fast and reliable propagation

of warning messages to upcoming vehicles in case of hazardous driving situations, such as dangerous road surface conditions, accidents events, unexpected fog banks, and so on [31]. This type of applications requires the definition of suitable data dissemination mechanisms, capable of delivering (alert) messages to the highest number of upstream vehicles in the shortest time possible. In order to meet these requirements, the design of broadcast protocols should exploit the peculiar features that differentiate CARNETs from classical wireless ad hoc networks.

In this scenario, we propose and analyze a position-based broadcast algorithm, named *Smart Broadcast* (SB), which permits fast and reliable message propagation in a inter-vehicular scenario. We consider a CARNET that relies upon MAC and physical layers derived by the IEEE 802.11 specifications. Furthermore, we assume that nodes are capable of determining their own position, by means of a suitable localization system. As observed in Chapter 2, many problems arise when packets have to be spread over wireless ad hoc networks. Our aim is to find a distributed, efficient and effective data dissemination mechanism which is able to perform closely to the MCDS ideal scheme. Note that, in this case, efficiency of the data dissemination algorithms has to be intended in terms of latency. The goal, indeed, is to find a way to reduce as much as possible the time required to deliver the alert messages to all the possible interested users. For these reasons, the core of the Smart Broadcast protocol is the contention-resolution phase that determines the next relay node<sup>1</sup> at each hop.

The effectiveness of SB approach will be proved by theoretical analysis and simulations. We also compare the SB performance with other existing position-based data dissemination schemes.

The rest of the Part is organized as follows. In Chapter 4 we overview the related work by briefly describing the data dissemination protocols we consider as comparison. In Chapter 5 we describe the *Smart Broadcast Protocol* in detail by providing the theoretical analysis of the protocol performance and by deriving the equations for the optimal parameters setting. We also validate the theoretical analysis by means of simulations and compare the SB performance with the other position-based algorithms.

---

<sup>1</sup>A relay node is a node entrusted to rebroadcast the message.

# 4

## Position-based Data Dissemination Protocols

---

*If knowledge can create problems,  
it is not through ignorance that we can solve them.*  
(Isaac Asimov)

Some of the existing data dissemination protocols have been summarized in Chapter 2. Here, we focus on two specific position-based schemes, namely *Urban Multi-hop Broadcast (UMB)* and the *Geographic Random Forwarding (GeRaF)* which are particularly suitable for inter-vehicular scenarios.

Note that we are mainly considering the position-based schemes. The reasons for that are manifold. First, we assume that, in our scenario, devices are equipped with a localization tool, that is reasonable in CARNETs. Thus, using position-based protocols is an immediate consequence. Second, these kind of strategies are more suitable to be implemented in a distributed way thus favoring the scalability. Third, some information about the nodes' position makes possible to optimize the data dissemination scheme in terms of propagation speed.

### 4.1 Urban Multi-hop Broadcast Protocol (UMBP)

The *Urban Multi-hop Broadcast Protocol (UMBP)* [32] protocol is explicitly designed for broadcast propagation in vehicular networks.

It is based on the IEEE 802.11 Distributed Coordination Function [7]. Accordingly, the transmission of any packet is preceded by the exchanging of two control packets between the transmitter and the receiver (usually referred to as RTS, i.e., *Request-To-Send* and CTS, i.e. *Clear-To-Send*, respectively). They are useful to prevent collisions. The core of the algorithm is the contention scheme used to select the next relay node and it can be summarized as follows.

- i) The coverage area of a node is equally partitioned in a given number of sectors,  $N_{max}$ . The relay node is selected in the furthest non-empty sector, so that the message progress is maximized.
- ii) The node that holds the broadcast message (source) transmits a MAC-broadcast<sup>1</sup> control packet, called *Request-to-Broadcast (RTB)* that is very similar to the RTS, which contains the geographical position of the source and the sector size.

---

<sup>1</sup>We use the term *MAC-broadcast* to denote one-hop broadcast transmissions. MAC-broadcast packets are never retransmitted by the receiving nodes.

iii) Upon receiving the RTB packet, nodes compute their distance from the source,  $\hat{d}$ . Then, nodes transmit a channel jamming signal, called *black\_burst*, that covers a number of time-slot equal to their distance from the source (in number of sectors): the further the distance, the longer the *black\_burst*. The expression to compute the *black\_burst* time for the first transmission attempt is given by:

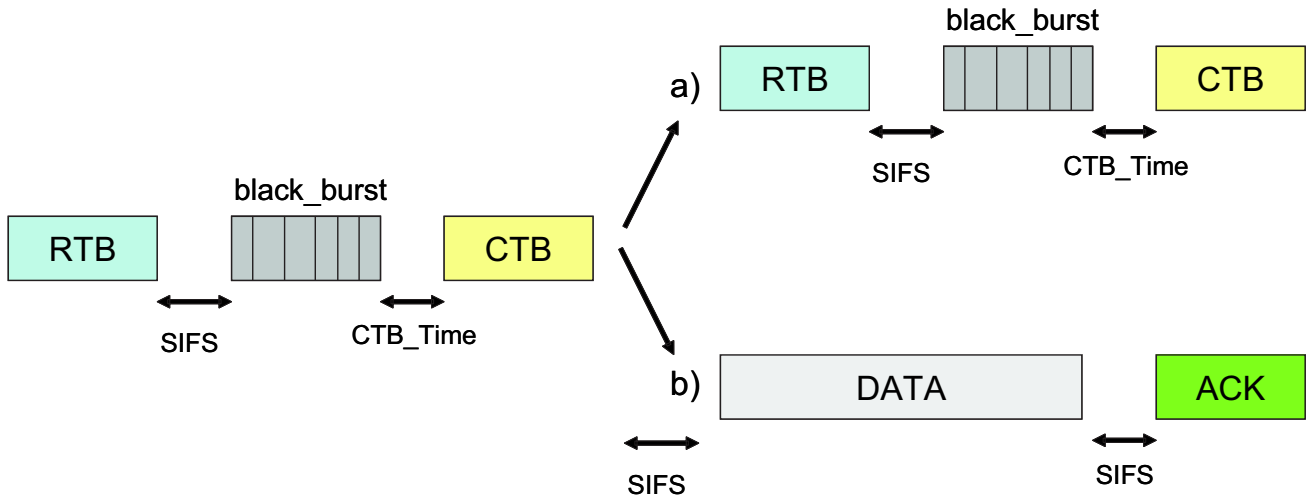
$$L_1 = \lfloor \frac{\hat{d}}{R} N_{max} \rfloor Time_{slot} \quad (4.1)$$

where  $\hat{d}$  is the distance between the transmitter and the receiver node,  $R$  is the coverage range of the transmitter,  $N_{max}$  is the number of sectors and  $Time_{slot}$  is the duration of a slot.

iv) Once a node has exhausted its *black-burst* transmission, it checks the channel status. If there are still ongoing transmissions, the node exits the contention phase. Conversely, if the channel is sensed idle, the node returns a *Clear-to-Broadcast* (CTB) control packet, containing its identifier (ID), to the source. Notice that all and only the nodes in the furthest non-empty sector will (simultaneously) transmit a CTB packet.

v) If the source receives a single CTB packet, then it forwards the message to the node that has originated the CTB, which becomes the next relay (see Fig. 4.1a).

vi) If there are multiple nodes in the furthest area, the CTB packets will collide at the source node so that a collision resolution phase has to start (see Fig. 4.1b). It consists on the same mechanism iterated only among the colliding nodes, over a finer space scale.



**Figure 4.1** Contention Phase of the Urban Multi-hop Broadcast Protocol in case of a) the CTB frame is not successfully received at the first transmission, b) otherwise.



- vii) After a given number of iterations without a winner, the surviving nodes enter a random contention phase at the end of which a single node is elected as relay.

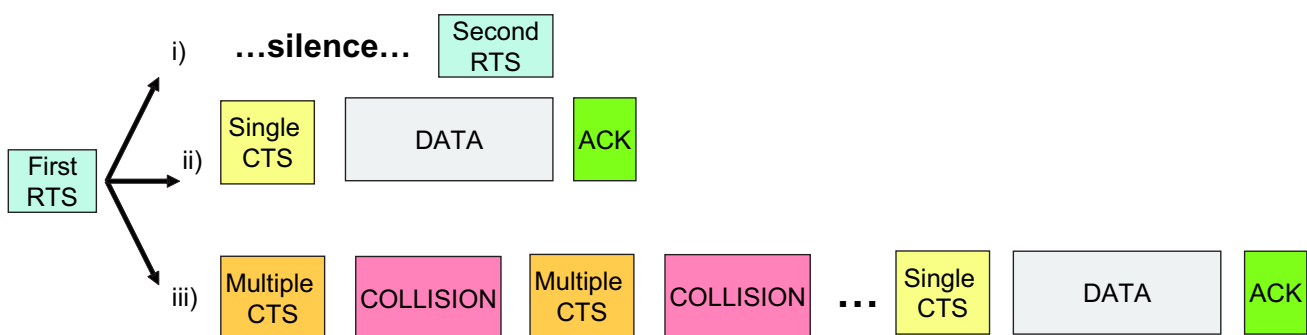
It is worth noticing that, according to the contention–resolution scheme, the potential relay nodes wait the longest time before retransmitting. This mechanism may lead to long latency, especially for high node densities thus it is not particularly suitable to deliver alert messages which need very low propagation delays.

## 4.2 Geographic Random Forwarding (GeRaF)

*Geographic Random Forwarding* (GeRaF) is a position–based routing protocol which presents several analogies with UMBP. Although GeRaF was developed for routing in wireless sensor networks [33, 34], it can be easily adapted to the scenario here considered. Similarly to UMBP, GeRaF attempts to maximize the progress of the message along the propagation line. To this end, the coverage area is equally divided in adjacent sectors, or advancement regions. Note that, the authors of [33] originally consider as advancements all the positions in the transmitter coverage area which are closer to the destination. In our case, we are interested in disseminate data to several users which are located in the same direction (the upcoming vehicles). Thus, we can identify as the advancement region the half of the coverage area in the direction of the message propagation.

The core of the GeRaF scheme is, as in the previous cases, the contention–resolution procedure for the election of the message relay. It involves the exchange of different control messages which can be summarized as follows (see Fig. 4.2).

- i) The source node polls the sectors in succession, starting from the farthest one. To this end, it transmits *Request-to-Send* (RTS) messages containing its geographical coordinates and the region the message is intended for. The first and the last RTS are intended for the nodes in the furthest



**Figure 4.2** Contention Phase of the GeRaF.

and nearest sectors, respectively. After any RTS transmission, the current source node waits a fixed time for some packets from its neighbors in the polled sectors. If none replies, it sends the following RTS.

- ii) Upon receiving the RTS message, all the nodes in the polled region reply by transmitting a *Clear-to-Send* (CTS) packet. If a single node returns the CTS packet, then it becomes the next relay and the source node starts to send the data.
- iii) If there are more nodes in the polled region, a collision occurs. In this case, the source issues a *COLLISION* message to start up a collision-resolution scheme at the eligible nodes. Nodes will reply to subsequent solicitations using a probabilistic bisection rule, that is, sending back control messages with a fixed probability of 0.5, until a node is finally elected next relay.

As UMB, also GeRaF attempts to maximize the per-hop message progress, to the expense of the forwarding delay. Indeed, the relay selection could last a long period due to different reasons. First, it can involve the exchange of a lot of control messages. This, using a IEEE 802.11 like protocol with random backoff can lead to a lot of wasting time. Second, the collision resolution phase may require a lot of attempts before a node wins the contention.

What we want to investigate is the trade off between maximizing the advancement progress and minimizing the time required to elect the next relay. We believe that a good set up of this trade off could lead to an algorithm with better performance in terms of delay. We go along this direction in the next Chapter.

# 5

## Smart Broadcast Protocol

---

*Speed is a great asset;  
but it's greater when it's combined with quickness  
- and there's a big difference.*  
(Ty Cobb)

In this Chapter we develop and analyze a data dissemination scheme for alert messages propagation in CARNETs. We have shown that, existing scheme try to maximize, at each hop, the progress towards the message propagation direction. This is done by reducing as much as possible the number of hops to relay the message to all the interested destinations. Nevertheless, the existing solutions design complex procedures to elect the next relay nodes which could lead to a long transmission delay. Our aim is to exploit the trade off between advancement and delay in order to develop a more performing scheme in terms of delay. This is due to the fact that we are interested in application where the delay constraints are usually very strong.

We first describe our reference scenario, then we give the detail of our protocol and, finally, we conclude with the results.

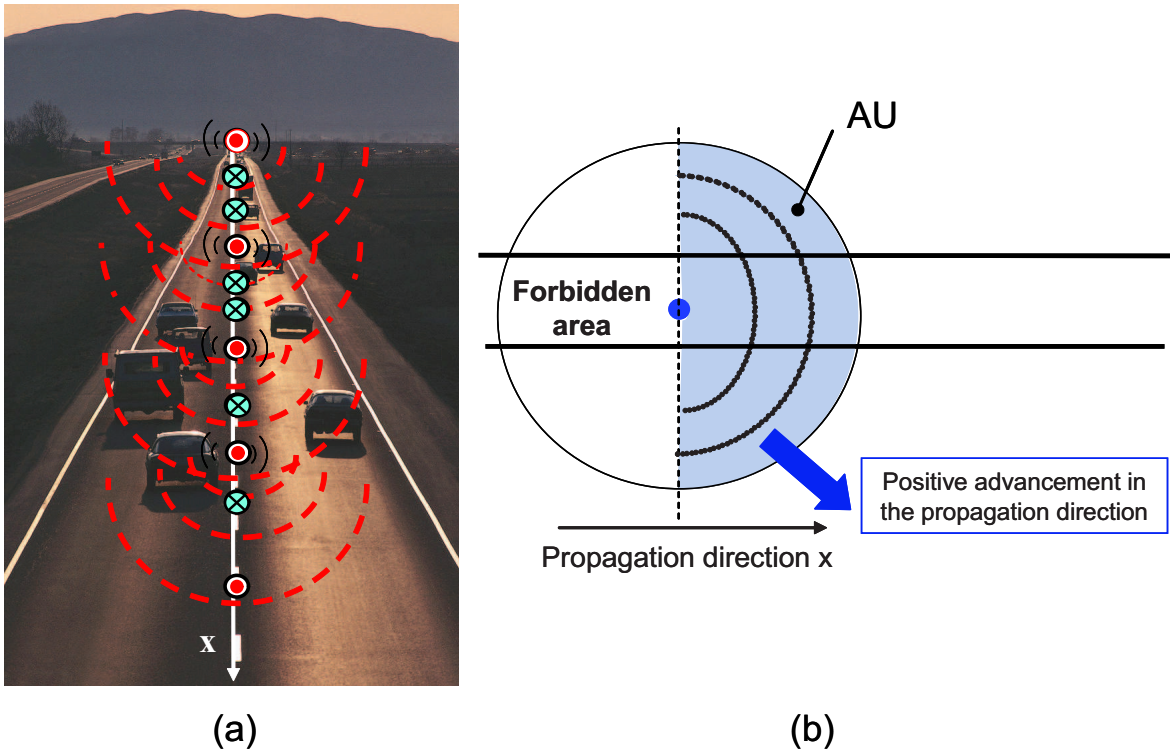
### 5.1 Reference Scenario

As we are interested in propagating alert messages in a CARNET, the more suitable application scenario is represented by a long highway where alert messages regarding accidents and dangerous situations have to be fast disseminated to the upcoming vehicles.

For these reasons, as reference network topology we focus on an ad-hoc network that develops along a strip-shaped area as reported in Fig. 5.1a.

We suppose that each vehicle is equipped with a GPS-like device so that each node knows its geographical position (with some approximation).

A node, named *source*, generates a broadcast message that has to be propagated along the strip in a specific direction. Each broadcast message contains a header field that includes the spatial coordinates of the transmitting node and the message-propagation direction which can be different according to the type of the alert message. To better catch the behavior of data dissemination scheme, we consider



**Figure 5.1** Reference Scenario.

no background traffic, so that only broadcast messages are propagated over the network.

We assume a unitary–circle reception model, so that a transmission is correctly received by all the nodes within a distance  $R$  from the source. However, we take this assumption only for the sake of simplicity, but the protocol works properly also when it is not satisfied. The road section is much smaller than the transmission range  $R$  of a node, so each transmission approximately covers a rectangular portion of the road, with size  $R$ . We set such a portion as the reference area unit (AU), as shown in Fig. 5.1b.

As our scheme will be based on similar rationale as the UMBP and GeRaf, we divided the portion of the coverage area in the direction of the message propagation in  $N_S$  sectors. Each of them has almost a rectangular shape and covers an area  $A = 1/N_S$  (assuming  $AU = 1$ ). Nodes are distributed on the strip according to a (bi–dimensional) Poisson process of intensity  $\lambda$  [nodes per unit of area], so that the number of nodes within the generic sector  $S_j$  will be a Poisson random variable of parameter  $\lambda_j = \lambda A$ .

Finally, we assume that nodes do not move significantly during the time taken by the contention procedure to be completed. Since the one–hop delivery time is of the order of milliseconds (as we will see in the next Sections) and the speed gap between vehicles proceeding in the same direction rarely exceeds 200 km/h, the variation of the distance in a hop time is negligible.

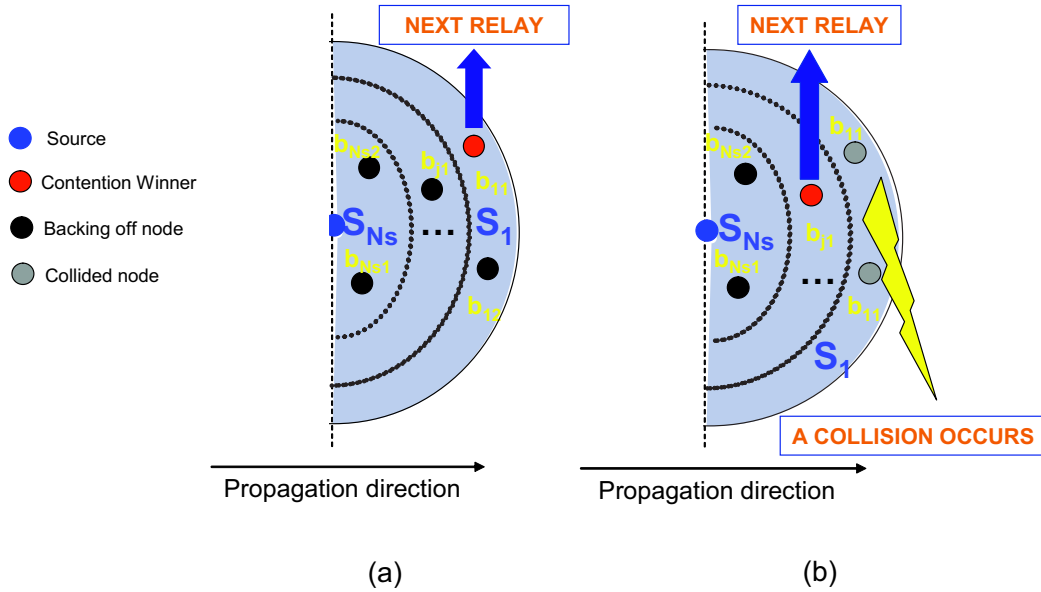


Figure 5.2 Main phases of the Smart Broadcast Protocol.

## 5.2 The Smart Broadcast Protocol Description

The *Smart Broadcast Protocol* (SB) has been designed to adhere as much as possible to the IEEE 802.11 specifications, so that its implementation in existing WiFi devices would require only marginal modifications of the existing firmware. We choose, as simulation environment, the IEEE 802.11b standard, but SB protocol can as well be implemented in an IEEE 802.11g/e/p PHY layer.

Similarly to UMBP and GeRaF, SB still leverages on the assumption that the coverage area can be partitioned in adjacent sectors and that nodes are capable of estimating their own position and, therefore, the sector they belong to. Hence, a contention–resolution procedure is performed to elect the relay node. Conversely to the other schemes, though, the SB does not necessarily select the relay in the region that provides largest progress, should this cost excessively in terms of time. The minimization of the *time* to perform a hop is, indeed, the main target of the Smart Broadcast (SB) protocol.

The details of the SB forwarding procedure are given in the following. To better understand the mechanisms, Fig. 5.2 reports the main phases of the relay–election procedure.

- i) The source node transmits a *Request-to-Broadcast* (RTB) control message. The RTB is a MAC–broadcast packet that contains the geographical position of the current source node and other control information, such as the sector width, the message propagation direction and the contention window size  $cw$ .
- ii) Upon receiving a RTB, nodes determine the sector they belong to by comparing their coordinates

with those of the source. Only nodes that follow the RTB source along the message propagation direction can participate to the relay election. Hence, such nodes pick a random backoff time in the contention window associated to the sector they belong to.

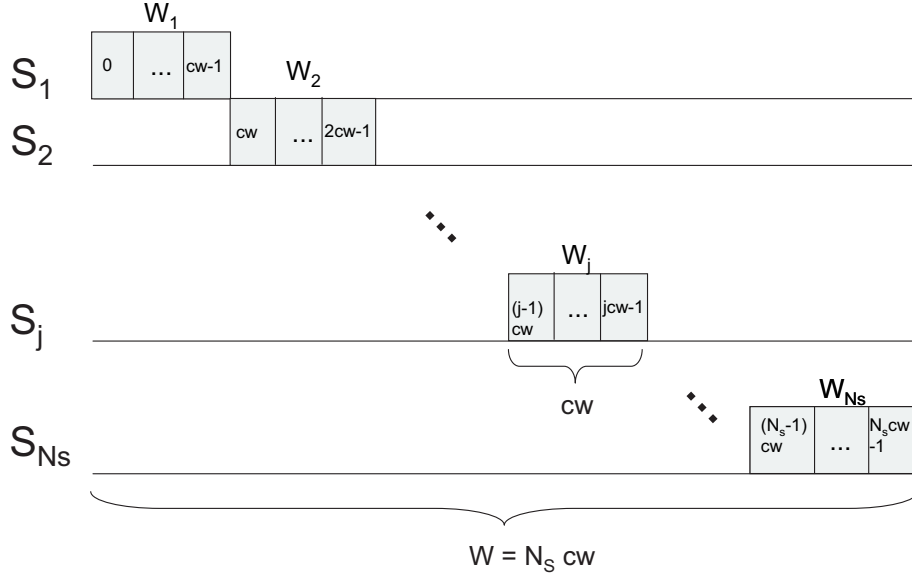
Let us name the  $N_S$  sectors from  $S_1$  to  $S_{N_S}$ , starting from the sector at the edge of the coverage range and moving towards the source node. Each sector  $S_j$  is associated to a contention window  $W_j$  of size  $cw$  (refer to Fig. 5.3), defined as follows:

$$W_j = \{(j-1)cw, (j-1)cw + 1, \dots, jcw - 1\}, \forall 1 \leq j \leq N_S;$$

Therefore, nodes in the outermost sector  $S_1$  will pick a random backoff value in the set  $W_1 = \{0, 1, \dots, cw - 1\}$ , nodes in  $S_2$  will select their backoff in the set  $W_2 = \{cw, cw + 1, \dots, 2cw - 1\}$  and so on. Notice that such contention windows provide a non-overlapping coverage of the set  $\mathcal{W} = \{0, 1, \dots, cw \cdot N_S - 1\}$ . Moreover, they guarantee that nodes in the further regions always transmit before the others.

- iii) According to the CSMA/CA policy of IEEE 802.11, the backoff counters are decremented by 1 at each idle slot, while countdown is frozen when the medium is busy. The countdown process is, hence, resumed after the channel has been idle for a *Distributed Inter Frame Spacing* (DIFS).
- iv) Whenever a node countdowns to zero, it sends a *Clear-to-Broadcast* (CTB) packet with its ID and coordinates. Nodes that receive a valid CTB packet, determine whether the CTB source lies farther along the message propagation direction and, in this case, they exit the contention phase. We assume that CTB packets do not require acknowledgement. This means that, in case of collision, the transmitting nodes do not longer participate to the contention (unless a new RTB packet is received in a later time). Nodes that overhear a collided transmission, on the contrary, remain in the contention phase and resume the backoff countdown as soon as the channel remains clear for a *Distributed Inter Frame Spacing* (DIFS).
- v) The contention phase is concluded when the source receives a valid CTB packet. Then, the source node broadcasts the message to nodes in its coverage area specifying within the MAC header the next relay, which is the node that has originated the successful CTB. The transmission occurs after a *Short Inter Frame Spacing* (SIFS), in order to gain priority over the still contending nodes.

Notice that, according to this strategy, the broadcast propagation can exhaust itself if no successful transmissions occur in any of the  $N_S$  sectors. To increase the robustness of the protocol, we assume that, after sending the RTB, the source node sets its own backoff counter to  $\max\{W_{N_S}\} = cw(N_S - 1)$ . If such a backoff is cleared before a valid CTB is received, the procedure is repeated anew after an extra time delay  $\Delta$ . This cunning makes the algorithm robust in case of node mobility and channel errors.



**Figure 5.3** Organization of the backoff windows.

## 5.3 Theoretical Analysis

In this Section we derive the analytical expressions of the average one-hop latency and message progress. Moreover, we will derive average one-hop progress and the message propagation speed.

The following proofs are referred to the network topology represented in Fig. 5.4.

### 5.3.1 One-hop Latency

First of all we consider the One-hop latency.

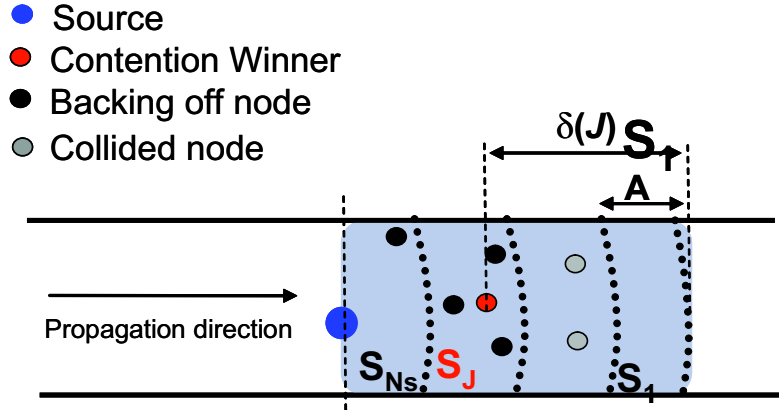
The *one-hop latency*,  $\tau$ , is defined as the mean time required before the broadcast message is successfully forwarded to the next relay node.

Note that this includes the time spent both to elect the relay node and to transmit the message.

Upon receiving a RTB packet, the nodes enter the contention phase and pick a random backoff value in their contention windows,  $W_j$ ,  $J = 1, 2, \dots, N_S$ . As noticed, the contention windows form a partition of the set  $\mathcal{W} = \{0, 1, \dots, N_S cw - 1\}$ . Let us denote by  $q_h$  the number of nodes that select the same backoff value  $h$ , with  $h \in \mathcal{W}$ . Since contending nodes are mutually in range, the countdown process occurs synchronously. Therefore, at the  $h$ -th countdown step, one of the following events occurs.

**I:**  $q_h = 0$ : No nodes transmit and the channel remains *Idle* for an entire slot.

**C:**  $q_h > 1$ : Multiple nodes transmit simultaneously, thus incurring into a *Collision*.



**Figure 5.4** SB: reference scenario for theoretical proofs.

**B** :  $q_h = 1$ : A single node transmits the CTB packet, thus winning the contention and becoming the next relay. After a SIFS, the node will receive the *broadcast message* to be relayed and the procedure will be concluded.

Under the assumptions reported in Section 5.1 (Poisson nodes distribution and independent back-off selection),  $\{q_h\}_{h \in W}$  are independent and identically distributed random variables, with a Poisson distribution of parameter  $\tilde{\lambda} = \lambda A / cw = \lambda / (cw N_S)$ . Therefore, the events  $I$ ,  $C$  and  $B$  occur with probabilities

$$\begin{aligned} P_I &= e^{-\tilde{\lambda}}; \\ P_C &= 1 - e^{-\tilde{\lambda}} (\tilde{\lambda} + 1); \\ P_B &= \tilde{\lambda} e^{-\tilde{\lambda}}; \end{aligned} \quad (5.1)$$

respectively. The number of unsuccessful events before the completion of the procedure is, hence, a geometrically distributed random variable, with average value  $n_U$  given by:

$$n_U = \frac{1 - P_B}{P_B}. \quad (5.2)$$

Now, the event  $I$  takes a time  $T_I$ , equal to a single time-slot. A collision event  $C$  takes a time  $T_C$ , given by the transmission time of a CTB packets, followed by a DIFS. Therefore, the average duration  $T_U$  of an unsuccessful countdown step is given by

$$T_U = T_I \frac{P_I}{1 - P_B} + T_C \frac{P_C}{1 - P_B}. \quad (5.3)$$

Finally, the event  $B$ , which concludes the contention phase, takes a time  $T_B$  that accounts for the CTB reception time, the SIFS and the message transmission time. The average re-broadcast time  $\tau$  can,



hence, be expressed as:

$$\tau = T_0 + T_B + n_U T_U + T_\Delta ; \quad (5.4)$$

where  $T_0$  is the *contention starting time*, equal to the RTB transmission time plus a DIFS. The term  $T_\Delta$  accounts for the extra time spent to restart the procedure, whenever no nodes in the coverage area win the contention. Under the simplifying assumption that successive iterations of the contention procedure are statistically independent, we easily get

$$T_\Delta = \left\lfloor \frac{n_U}{cwN_S} \right\rfloor (T_0 + \Delta) ;$$

where  $\lfloor x \rfloor$  denotes the integer part of  $x$ . Notice that, in typical operating condition, the contention procedure is successfully completed within a maximum contention window, so that  $T_\Delta$  can be generally neglected.

Replacing Eq. (5.2) into Eq. (5.4) (and neglecting  $T_\Delta$ ) we finally get

$$\tau \simeq T_0 + T_B + T_I \frac{P_I + KP_C}{P_B} ; \quad (5.5)$$

where the factor  $K$  is defined as  $K = T_C/T_I$ .

### 5.3.2 One-hop Message Progress

Let us now focus on the one-hop message progress.

The *one-hop message progress*,  $\delta$ , is defined as the additional distance covered by the message in a re-broadcast phase, on average. The message progress is equal to the average distance between the next relay and the source node.

Let us recall that sectors are numbered from 1 to  $N_S$ , starting from the furthest to the source node. Furthermore, let us assume that the next relay belongs to the sector  $J$ . Under this condition, the average message progress is given by

$$\delta(J) = (N_S - J)A + A/2 ; \quad (5.6)$$

where we recall that  $A = 1/N_S$  is the size of each sector. Notice that, the sector  $J$  contributes to the message progress only for half of its spatial extension. The reason is that nodes are randomly distributed within the sector, so that the relay node, on average, will be positioned in the middle of the sector.

Now, it remains to determine the statistic of  $J$ . From the message-progress perspective, each repetition of the contention phase represents a renewal epoch. Hence, we can focus on the contention phase where the relay is elected. The probability that the next relay node belongs to the sector  $J = r$  is equal

to the probability that the successful event  $B$  occurs at the backoff slot  $s \in W_r$ , given that  $B$  occurs within the  $cwN_S$  steps. In formula, we have

$$P_J(r) = P[s \in W_r | s \in \mathcal{W}], \quad r = 1, 2, \dots, N_S. \quad (5.7)$$

Denoting by  $P_s(h)$  the conditioned probability that  $s = h$ , given that  $s \in \mathcal{W}$ , we have

$$P_s(h) = \begin{cases} \frac{(1 - P_B)^h P_B}{1 - (1 - P_B)^{cwN_S}} & h = 0, 1, \dots, cwN_S - 1; \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

Putting Eq. (5.8) in Eq. (5.7), we easily get

$$P_J(r) = \sum_{h=w_r}^{w_r+cw-1} P_s(h) = \frac{(1 - P_B)^{w_r} (1 - (1 - P_B)^{cw})}{1 - (1 - P_B)^{cwN_S}}. \quad (5.9)$$

Hence, from Eq. (5.9) we get the average value of  $J$ :

$$m_J = \sum_{r=1}^{N_S} r P_J(r) = \frac{1}{1 - (1 - P_B)^{cw}} - \frac{N_S (1 - P_B)^{cwN_S}}{1 - (1 - P_B)^{cwN_S}}. \quad (5.10)$$

Finally, taking the expectation of both sides of Eq. (5.6), we get the final expression of the average per-hop progress:

$$\delta = (N_S - m_J)A + A/2. \quad (5.11)$$

### 5.3.3 Message Propagation Speed

We focus now on the last performance metric.

The *message propagation speed*,  $v$ , is defined as the number of area units covered by the message in a second.

In general, the process that describes the propagation of the message along a direction is correlated [9]. For the sake of simplicity, though, we neglect such a correlation and defines the average message propagation speed,  $v$ , as the ratio between the one-hop message progress  $\delta$ , measured in Area Units [AU], and the average one-hop latency  $\tau$ , measured in seconds:

$$v = \frac{\delta}{\tau}, \quad [AU/s]. \quad (5.12)$$

## 5.4 Optimal Parameters Setting

The analytical results give us a powerful tool to maximize the performance of SB. Starting from the analytical expression of  $\tau$  we can determine the setting of the protocol parameters that minimizes the message propagation latency.

The one-hop latency  $\tau$  is a function of the parameter  $\tilde{\lambda} = \lambda/(cwN_S)$ , through the probabilities  $P_I$ ,  $P_B$  and  $P_C$ . The factor  $\lambda$  is given by the nodes density in the coverage area and, hence, it depends on the considered scenario. Therefore, the two protocol parameters that can be set are  $N_S$  and  $cw$ . In order to speed up the propagation of the message, we set  $N_S$  to the largest value possible, which is determined by the precision of the node-position estimation and by the maximal coverage area.<sup>1</sup> Hence, the only remaining parameter to optimize is the contention window size  $cw$ .

In the following, we derive the value of  $cw$  that minimizes the average re-broadcast latency  $\tau$  given in Eq. (5.5). This is equivalent to minimize the cost function  $C(\tilde{\lambda}) = T_I(P_I + KP_C)/P_B$ . Replacing  $P_C$  with  $1 - P_I - P_B$  we have

$$C(\tilde{\lambda}) = -T_I K + T_I \frac{K - (K - 1)P_I}{P_B}. \quad (5.13)$$

Setting to zero the derivative of  $C(\tilde{\lambda})$  in  $\tilde{\lambda}$  we get, after some algebra, the following transcendent equation

$$\tilde{\lambda} = 1 - \frac{K - 1}{K} e^{-\tilde{\lambda}}. \quad (5.14)$$

Eq. (5.14) admits a single solution  $\tilde{\lambda}_{opt}$  in the interval  $(1/K, 1)$ , which can be easily found with standard numerical methods. The optimal  $cw$  value is, hence, obtained as

$$cw_{opt} = \text{round} \left( \frac{\lambda}{N_S \tilde{\lambda}_{opt}} \right); \quad (5.15)$$

where  $\text{round}(x)$  denotes the rounding function.<sup>2</sup>

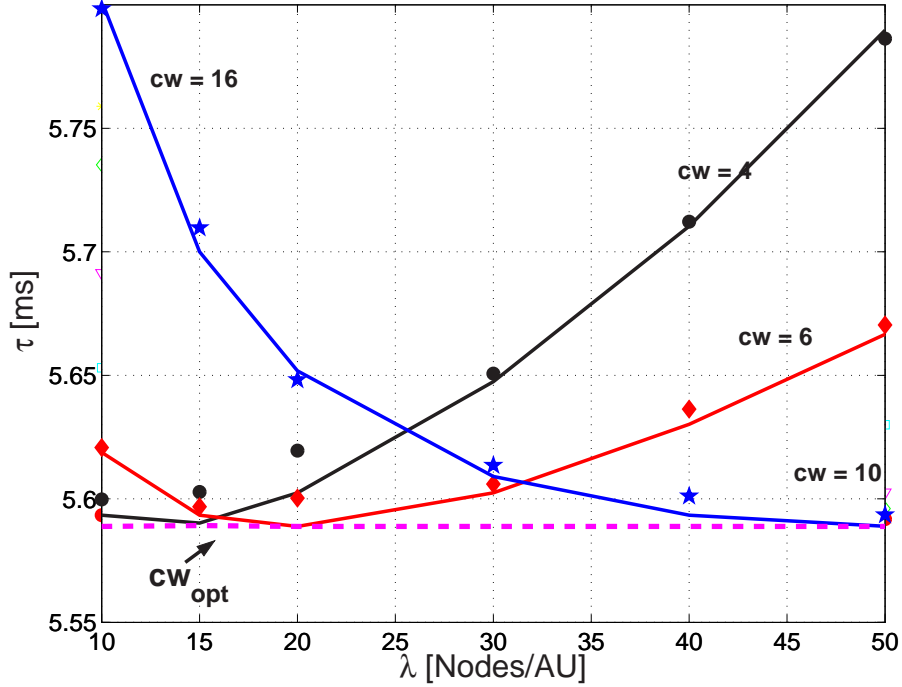
## 5.5 Validation of the Theoretical Analysis

In this Section, we compare the mathematical results obtained in the previous Sections with simulation outcomes, in order to validate our theoretical analysis.

Fig. 5.5 shows the one-hop latency versus  $\lambda$ , for different values of  $cw$ . Lines refer to the theoretical results given by Eq. (5.5), while marks refer to the simulation outcomes. Such values are given by the ratio between the time spent to propagate the message over a given distance and the number of hops performed. In this way, we consider the correlation in the message propagation process that, on the contrary, is not included in the theoretical model. Nevertheless, the good matching of analytical and simulation results confirms the validity of the model.

<sup>1</sup>The tradeoff in the choice of  $N_S$  is between the probability that a sector is empty and the speed of the broadcast propagation. In the rest of the Chapter we assume  $N_S = 10$ .

<sup>2</sup>To avoid pathological cases, it is wiser to set  $cw_{opt}$  as the maximum between Eq. (5.15) and 2.

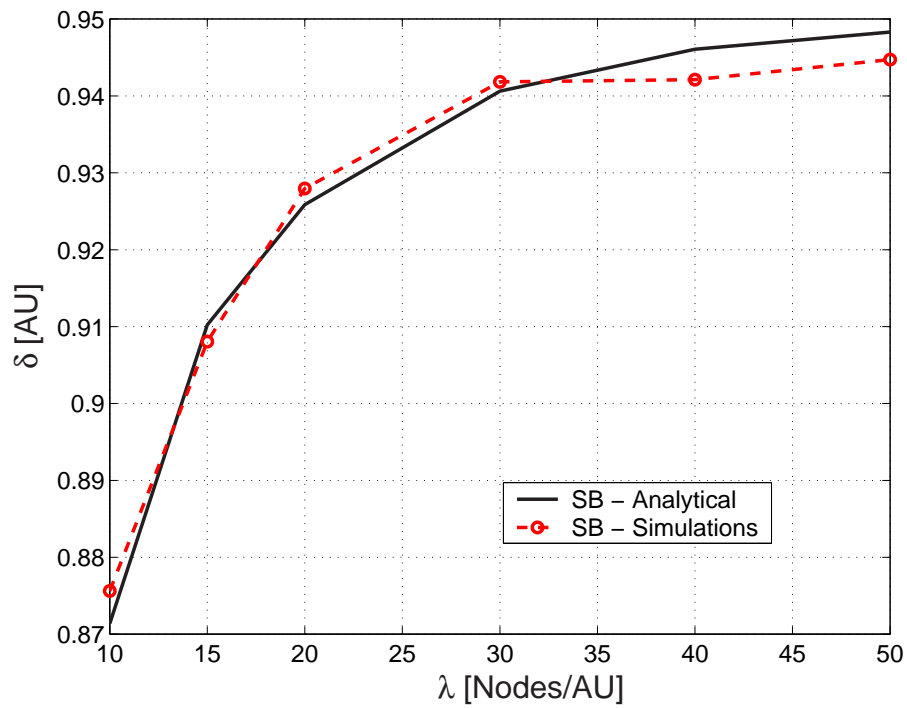


**Figure 5.5** One-hop latency  $\tau$ .

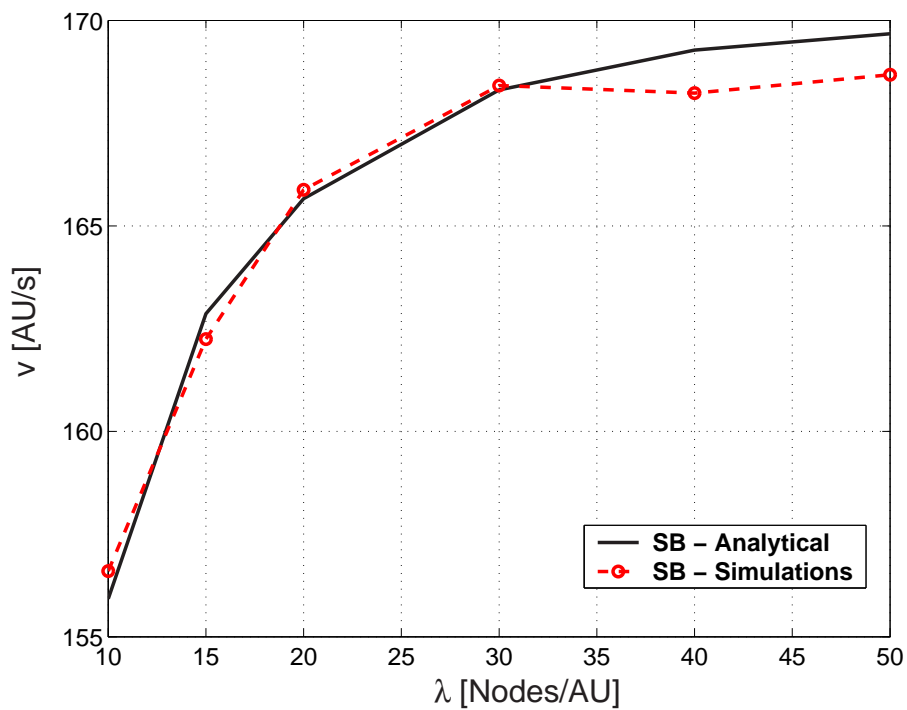
Fig. 5.5 also proves the validity of the optimization proposed in Section 5.4. The dashed bold curve that interpolates the minimum values of the other curves in the figure, indeed, has been obtained by Eq. (5.5), taking the  $cw_{opt}$  for each  $\lambda$ . We can see that, by using the optimal contention window value, we always get the lowest delay over all the possible  $cw$  values. This curve also reveals another important result: the per-hop latency obtained by using  $cw_{opt}$  is approximately constant at varying of the node density  $\lambda$ .

Fig. 5.6 shows the average one-hop progress  $\delta$  versus the node density  $\lambda$ . Curves have been obtained by assuming  $cw_{opt}$  for each  $\lambda$ . The continuous curve refers to the theoretical results, given by Eq. (5.11), while the dashed line interpolates the simulation outcomes and reports the experimental  $\delta$  given by the ratio between a reference distance (measured in area units) and the mean number of hops needed to reach this reference distances. The figure reveals that Eq. (5.11) captures rather closely the actual performance of the protocol. Moreover, we can observe that the higher the node density, the closer the per-hop progress to the maximum possible.

Finally, Fig. 5.7 represents the propagation speed,  $v$ , versus the nodes density  $\lambda$ . Once again, curves have been obtained by considering the optimal  $cw$  setting for each  $\lambda$ . The theoretical curve (continuous line) is given by Eq. (5.12). The simulation values, instead, are obtained as the ratio of the distance covered by the broadcast message in a time  $T$  over  $T$ . Once more, the matching of the two curves is



**Figure 5.6** One-hop message progress  $\delta$  with optimal parameters setting.



**Figure 5.7** Message propagation speed  $v$  with optimal parameters setting

rather good, thus confirming the validity of the theoretical model.

## 5.6 Performance Metrics, Simulations and Results

In the previous Section we had a preliminary view of the SB performance. In particular we studied its behavior at the varying of  $\lambda$  to prove the validity of the theoretical analysis. In this Section, instead, we focus on more practical aspects by comparing the performance of SB with the other existing schemes introduced in Chapter 2 and 4.

We focus particularly on the following performance metrics, which are the ones already defined in the previous Sections, plus three additional ones more meaningful from a practical point of view.

*R*: *Reliability*, defined as the ratio between the number of nodes reached by the broadcast message and the total number of nodes in the observed area ( $R = N_R/N_T$ ).

*U*: *Redundancy Factor* ( $U = N_{Tx}/N_T$ ), defined as the ratio between the number of retransmitting nodes ( $N_{Tx}$ ) and the total number of nodes ( $N_T$ ).

*T*: *Average Broadcast Time Delay*, defined as the time needed to reach all the nodes in observed area.

*v*: *One-hop message propagation speed*, defined as the average distance covered in one hop divided by the average time required to complete one transmission.

$\delta$ : *One-hop message progress*, defined as the average distance covered by each hop.

These indexes are selected to point out the proposed protocol improvement regarding the broadcast storm problem.

All the considered algorithms are implemented in the same simulator and run over the same scenario. To this aim we developed a network simulator, by using the OPNET commercial platform where the different solutions have been implemented upon an IEEE 802.11b MAC and physical layer.

Under these assumptions, we first compare SB with the flooding and the MCDS<sup>3</sup>-based protocols which represent the worst and best case, respectively.

Fig. 5.8, 5.9, 5.10 show the obtained results in terms of reliability (*R*), redundancy factor (*U*) and average broadcast time delay (*T*), respectively. We can see that *Smart Broadcast* performs much better than flooding algorithm and rather closely to the MCDS-based scheme.

Fig. 5.8 shows that the SB protocol achieves satisfactory performance in terms of reliability. In particular, it performs very closely to the MCDS-based scheme which guarantees always a full reliable system. In addition we observe that the SB reliability is almost constant as the node density increases.

<sup>3</sup>Minimum Connected Dominating Set

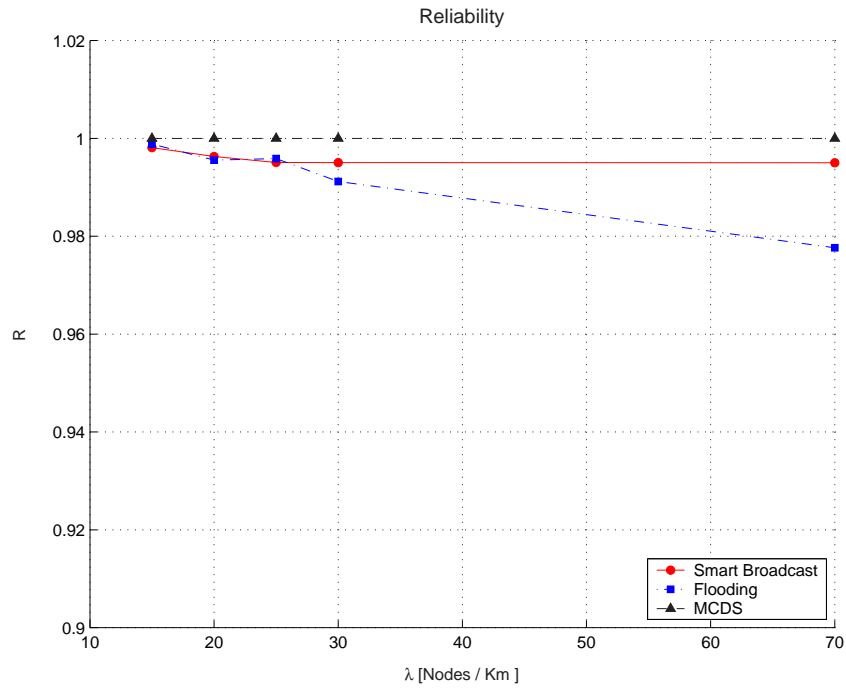


Figure 5.8 Reliability, R: comparison between Smart Broadcast, Flooding and MCDS-based algorithm

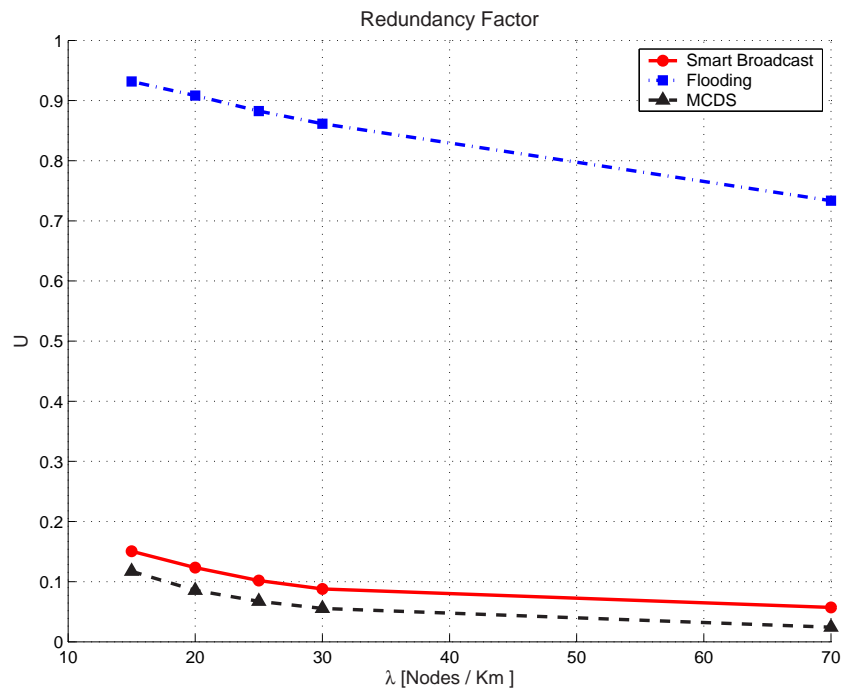
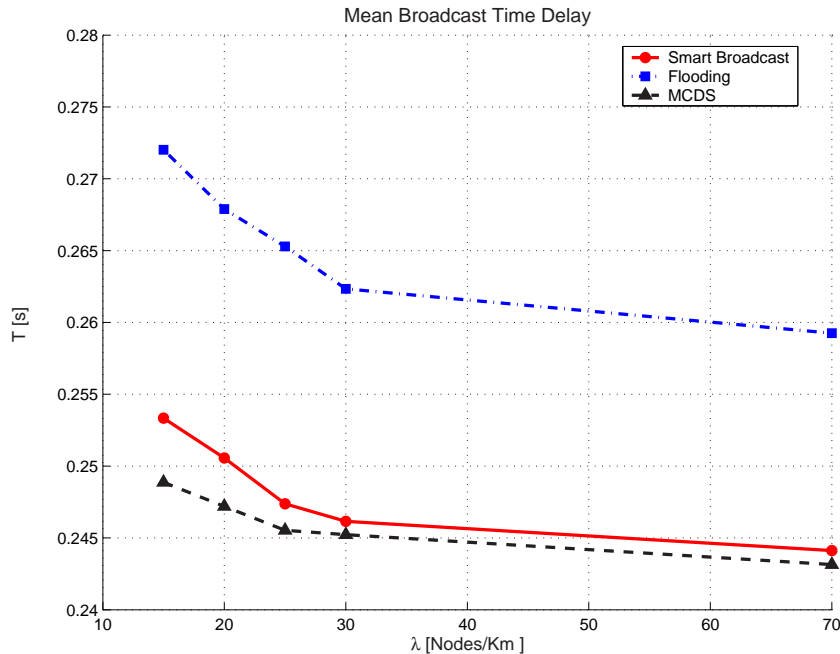


Figure 5.9 Redundancy Factor, U: comparison between Smart Broadcast, Flooding and MCDS-based algorithm



**Figure 5.10** Average broadcast time delay: comparison between Smart Broadcast, Flooding and MCDS-based algorithm

Note that the cases of high node densities are particularly challenging as they generate a lot of traffic, congestion and contention during the relay election phase. Thus, we expected decreasing performance of Smart Broadcast. Instead, the optimal setting of the parameters has beneficial effects and it guarantees the reliability of the system. On the contrary, the reliability level achieved by the simple flooding scheme suffers when the node density increases.

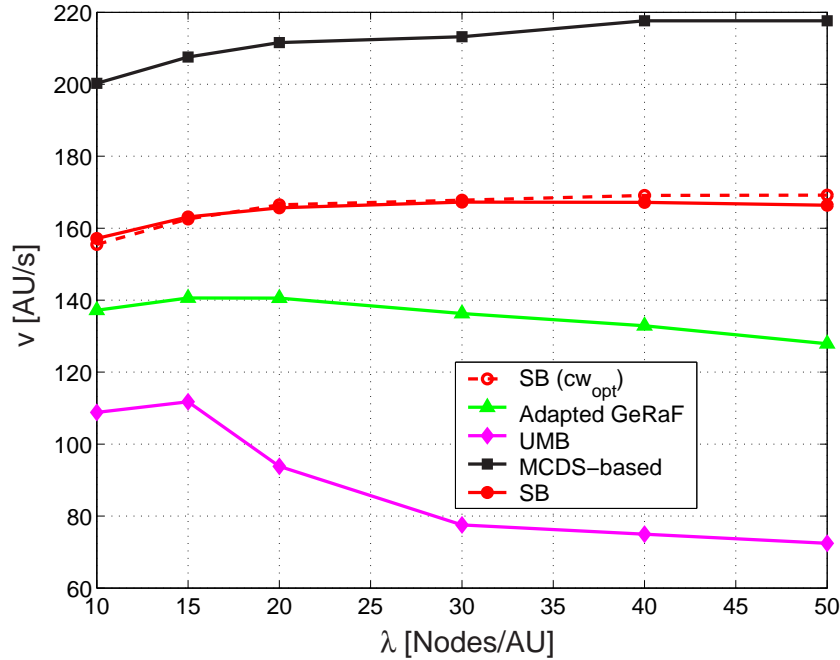
The redundancy factor curve reveals that SB can be useful to decrease the number of retransmitting nodes thus reducing the channel occupation and redundancy, as highlighted in Fig. 5.9. Note that the MCDS-based protocol introduces the lowest redundancy.

The broadcast time delay, represented in Fig. 5.10 gives us a measure of the promptness of the proposed protocol. In the broadcast applications for road-safety, the increasing of message propagation speed over the network is a primary aim of an efficient broadcast protocol design. Using the proposed protocol we show that is possible to reach all nodes in a wide area in a very short time (about 250 ms).

These preliminary simulations results show that SB is a promising scheme to time-efficiently disseminate data as its performance are very close to the performance achieved by the MCDS-based protocol. The next step of our analysis is comparing the SB behavior with more interesting solutions such as the UMBP and the GeRaF.

The protocol parameters of UMBP and GeRaF algorithms have been set as suggested in [32]





**Figure 5.11** Message propagation speed: protocols comparison

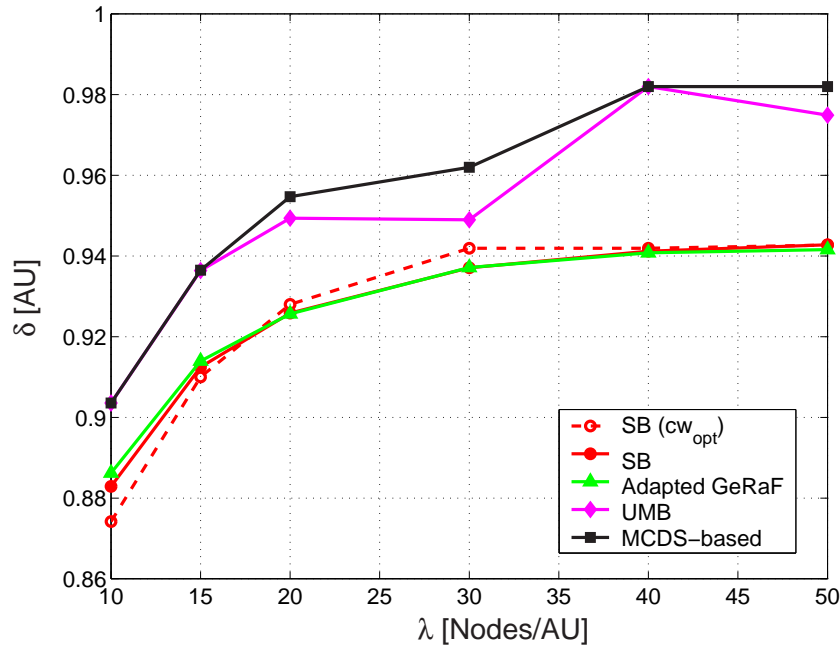
and [33], respectively. In particular, the number of sector  $N_S$  has been fixed to 10, as suggested in [32]. For the sake of fairness, we assume that all the schemes have equal transmission rate and coverage range and that the control packets are of the same size.

Fig. 5.11 shows the average propagation speed,  $v$ , for each scheme. In order to evaluate the dependency of the SB performance on the setting of the contention window parameter, we considered two set of results. The first, represented with a continuous line, has been obtained by fixing  $cw = 6$ , while the other, plotted with a dashed line, has been obtained by adopting the  $cw_{opt}$  for each  $\lambda$ .

From the Figure we can observe that the propagation speed achieved by SB is almost constant at varying of the node density. On the contrary, the propagation speed obtained by UMBP and GeRaF decreases with the increasing of the node density. The reason is that, in UMBP and GeRaF, higher nodes densities determine a greater number of collisions during the contention phase and, consequently, an increasing of the wasted time.

So far as SB is concerned, we can also observe that setting  $cw = 6$  leads to a marginal loss of performance with respect to the optimal case, thus proving that the SB scheme is robust to the variation of the scenario, provided that  $cw$  is set to optimal.

Fig. 5.12 shows the average one-hop progress,  $\delta$ , versus the node density  $\lambda$ , obtained by the different schemes forwarding a message over a reference distance. As we can see, the SB may lead to a slightly lower advancement than the other schemes. This is due to the fact that SB balances both the message



**Figure 5.12** One-hop message progress  $\delta$ : protocols comparison

progress and the latency, so that it might prefer a slightly longer but faster path over a shorter but slower one.

This last observation is particularly important as it states that to achieve the best time delay performance we do not need to maximize the message advancements. Indeed, the reduction of the wasting of time during the relay election phase has a stronger impact in the delay performance. The SB, compared to the UMBP and GeRaF, guarantees a faster procedure thus resulting in a winning strategy.

## 5.7 Robustness Analysis of Smart Broadcast Protocol

The optimization phase of SB, described in Section 5.4, determines the optimal value of the contention window size on the basis of the node density,  $\lambda$ . The main assumption, there, is that nodes are placed according to a Poisson process where the average node density is  $\lambda$ . However, as  $\lambda$  is constant, we are assuming that along the whole network nodes are distributed according to the same process. This fact could not be representative of the most common situations. We know that, in mobile scenarios, the node density could change sensitively from one area to another and also the statistical distribution of nodes can vary according to some mobility patterns. Thus, we need to investigate the performance of SB also when the node distribution is not a Poisson process and when the average node density can vary from one area to another.

In this Section, then, we study the robustness of SB to the variation of the node density  $\lambda$  (both in its average value and distribution). In particular, we introduce a formula to compute the optimal contention window size for general cases.

### 5.7.1 Validity Interval

To better understand our reasoning, we report here Eq. (5.1) and Eq. (5.13) found in Section 5.3 as function of two variables:

$$\begin{aligned}
 P_I(\lambda, cw) &= e^{-\frac{\lambda}{cw}}, \\
 P_B(\lambda, cw) &= \frac{\lambda}{cw} e^{-\frac{\lambda}{cw}}, \\
 P_C(\lambda, cw) &= 1 - e^{-\frac{\lambda}{cw}} \left( 1 + \frac{\lambda}{cw} \right), \\
 C(\lambda, cw) &= \frac{K - (K - 1)P_I}{P_B}.
 \end{aligned} \tag{5.16}$$

Note that assuming for  $\lambda$  a fixed value we obtain the same equations and cost function reported in Section 5.3. Fig. 5.5, as noticed before, reports the one-hop latency as function of  $\lambda$  for different  $cw$  and for  $cw_{opt}$ .

On the contrary, we want now to investigate how to optimize the value of  $cw$  for different distributions and for a wide range of average values of  $\lambda$ .

As a first step, we calculate the interval of  $\lambda$  that leads to the same value of  $cw_{opt}$ .

Let  $\lambda_0$  be a particular value of  $\lambda$  which leads to a specific  $cw_{opt_0}$ . Then, defining  $\mu = \frac{\lambda_0}{cw_{opt_0}}$  we can rewrite the last equation of Eq. (5.16) as:

$$\mu = 1 - \frac{K - 1}{K} e^{-\mu}, \tag{5.17}$$

which is an implicit function of  $\mu$  (dependant by  $K$ ).

On this basis we can write, in general,

$$\lambda_0 = \mu(K) cw_{opt_0}. \tag{5.18}$$

This relation is linear in  $\mu$  if, for instance,  $K$  is constant as it happens in our scenario.

If we assume  $cw_{opt_0} \in \mathbb{N}$ , using the following relation

$$\left[ \frac{\lambda_0}{\mu} \right] = \widetilde{cw}_{opt_0} \tag{5.19}$$

we obtain:

$$(\widetilde{cw}_{opt_0} - 1)\mu < \lambda < \widetilde{cw}_{opt_0}\mu. \quad (5.20)$$

Note that  $\widetilde{cw}_{opt_0}$  is only an approximation of the value  $cw_{opt_0}$  because we assume it is an integer. However, it represents a good value to estimate the validity interval of  $\lambda$ .

If we use the parameter of the scenario defined in Section 5.1, we obtain an interval with a range of at maximum 6.50[Nodes per area unit]. It is a good value but, in general, in wide and mobile networks it could be too restrictive. In addition it is referred to topologies where nodes are placed according to a Poisson distribution.

Thus, to increase the robustness of our scheme, we introduce a new optimization strategy which is able to guarantee satisfactory performance for a wider range of node density and for different distributions.

## 5.7.2 Optimal Contention Windows Size on Average

In this Section we calculate a value for  $cw$  which is optimal for a wide range of  $\lambda$ . We observe that, given a  $cw_{opt_0}$ , if we consider  $\tilde{\lambda} \neq \lambda_0$ , the value taken by the cost function measured for  $\tilde{\lambda}$  is higher than the value measured for  $\lambda_0$ . As an example, we report in Fig. 5.13 the cost function versus  $\mu$  for  $K = 15.2$  which is the value related to our scenario.

For this reason, we try to calculate a value for  $cw$  which gives the same value for the cost function even when  $\lambda$  varies. This means that, varying  $\lambda$ , the derivative of the cost function has to be equal to zero.

Let  $\lambda$  be a generic random variable. The last equation of Eq. (5.16) becomes a random process with  $cw \in \mathbb{R}$  which can be written as follows:

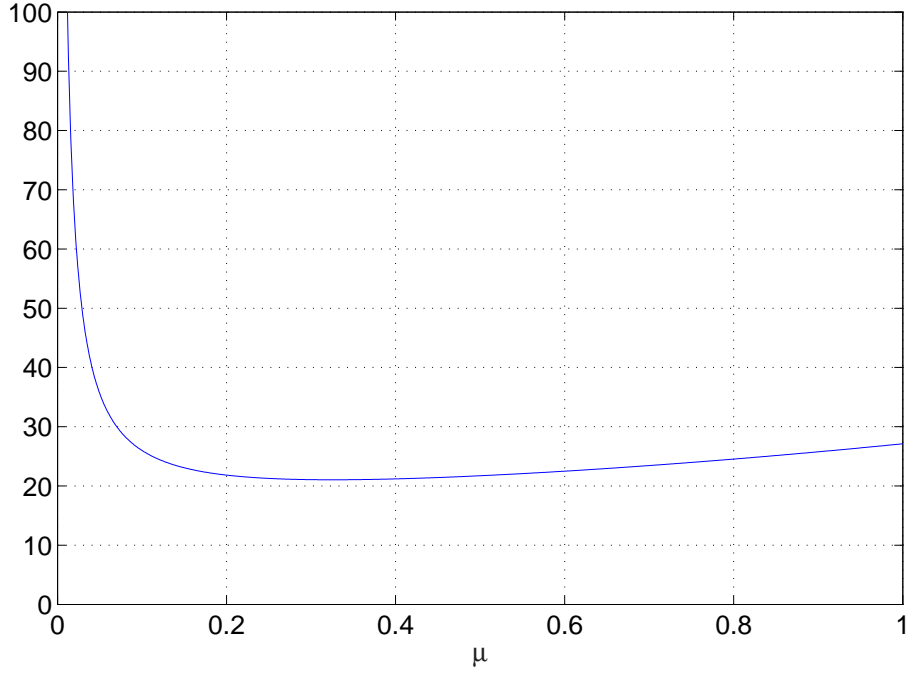
$$C(\lambda, cw) = \frac{cw}{\lambda} [K e^{-\frac{\lambda}{cw}} - (K - 1)]. \quad (5.21)$$

This random process is derivable if and only if all its realization are derivable. This is true if  $\lambda \in \mathbb{R}$ . Then we compute the derivative of  $C(\lambda, cw)$  and we force it to be zero on average.

$$E \left[ \frac{dC(\lambda, cw)}{dcw} \right] = 0. \quad (5.22)$$

From Eq. (5.22), after some algebra, we obtain the expression to calculate the value of  $cw$  when  $\lambda$  is a random variable with probability distribution  $f_\lambda$ . It is written as:

$$\frac{1}{cw} \frac{\int_0^\infty e^{\frac{a}{cw}} f_\lambda(a) da}{\int_0^\infty \frac{e^{\frac{a}{cw}}}{a} f_\lambda(a) da} = 1 - \frac{K - 1}{K} \frac{\int_0^\infty \frac{1}{a} f_\lambda(a) da}{\int_0^\infty \frac{e^{\frac{a}{cw}}}{a} f_\lambda(a) da} \quad (5.23)$$



**Figure 5.13**  $C(\mu)$  for  $K = 15.2$ .

where  $\int_0^\infty \frac{e^{-\frac{a}{cw}}}{a} f_\lambda(a) da > 0$ .

Note that, for  $f_\lambda = \delta(a - \lambda_0)$  we obtain the same expression of Eq. (5.15). In other words, Eq. (5.23) gives, for  $\lambda = \lambda_0$ , the same value of  $cw_{opt}$  found in Section 5.3. We refer to the optimal values of  $cw$  given by Eq. (5.23) as  $cw_{f_\lambda}$ .

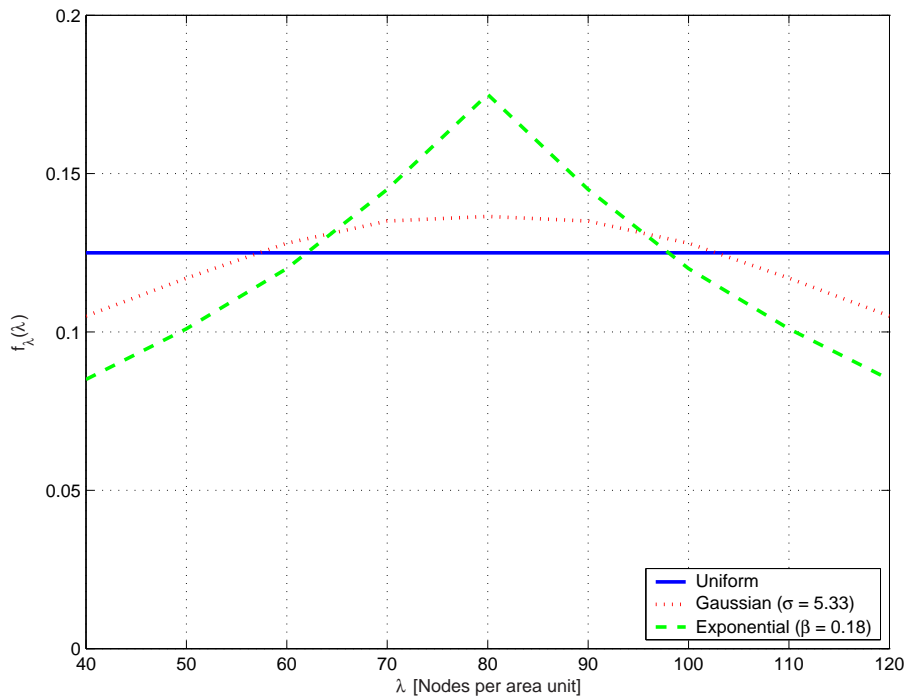
### 5.7.3 Performance Evaluation

We evaluate the performance of our approach by considering different node density distributions: uniform, gaussian and exponential. Note that we are focusing on the proper truncate versions of these distributions around different average values because we need to represent node densities which can not assume negative values. As an example we refer to Fig 5.14.

In Fig. 5.15 we report the  $cw_{f_\lambda}$  for different distributions of  $\lambda$ . On the x-axis we report the average value of  $\lambda$  whereas the standard deviation is 20 nodes per AU for all the distributions. We compare  $cw_{f_\lambda}$  with the  $cw_{opt}$  computed for a constant  $\lambda$  in Section 5.3. We observe that, in general,  $cw_{opt}$  is higher than  $cw_{f_\lambda}$  and for some distributions the difference is more evident. As an example, when the probability distribution is uniform or gaussian, the  $cw_{f_\lambda}$  can be lower than  $cw_{opt}$  of two slots. This means that, when we consider  $\lambda$  as a random variable, the lower values of  $\lambda$  have a higher weight in

the computation of the optimal value of  $cw$ .

In Fig. 5.16 we compare the performance of SB when  $\lambda$  is constant and equal to  $\lambda_0$  and when it is a random variable with average equal to  $\lambda_0$ . More in detail, we disseminate broadcast messages over a multi-hop network where, in the first case, the node density is constant whereas, in the other case, it changes from one hop to another. In addition, in the first case we use the optimal values of  $cw$  ( $cw_{opt}$ ) while in the second case we adopt  $cw_{f_\lambda}$ . As we can observe,  $cw_{f_\lambda}$  guarantees almost optimal SB performance also when the node density changes.



**Figure 5.14** Examples of considered probability distribution of  $\lambda$

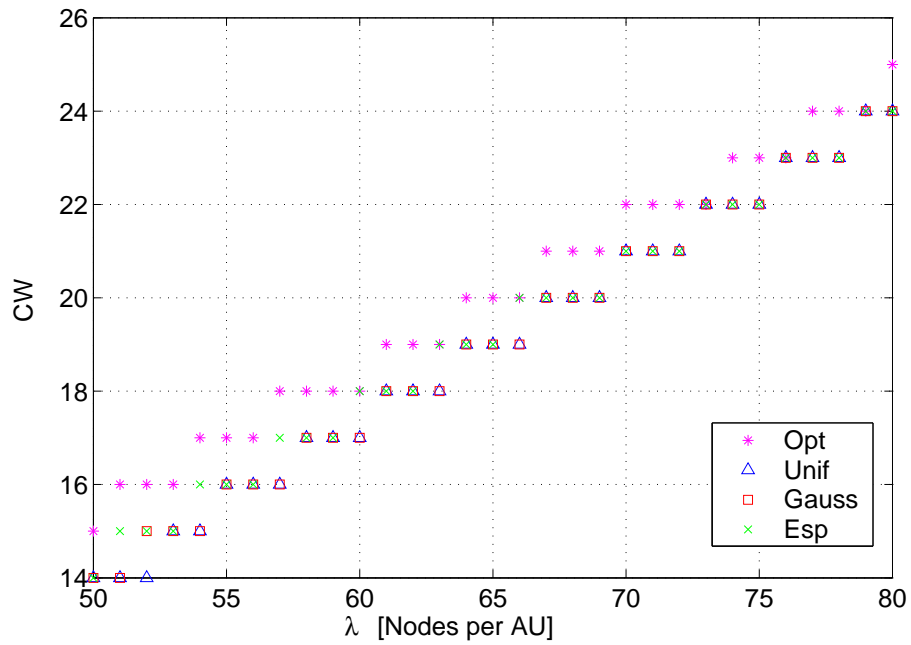


Figure 5.15 Optimal values of  $cw$  when the standard deviation of  $\lambda$  is 20[Nodes per AU]

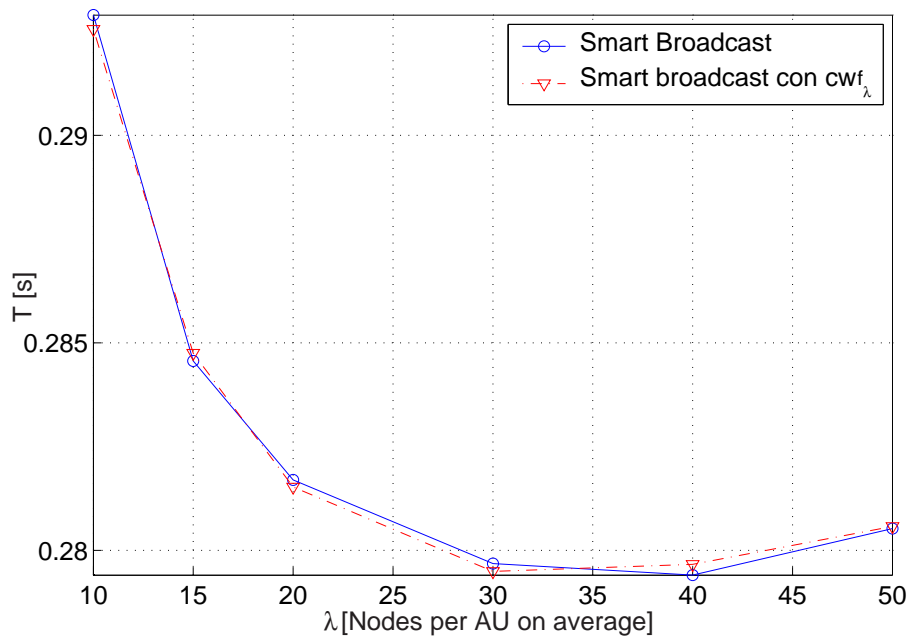


Figure 5.16 SB average broadcast delay for constant and variable node density

## 5.8 Discussion and Conclusions

In this Part, we have proposed and analyzed a novel position-aware protocol, named Smart Broadcast, for fast and reliable message propagation in CARNETs. It makes use of position information provided by a GPS-like system to speed up the message propagation along the network, by allowing furthest nodes to attempt transmission first.

The protocol performance has been theoretically evaluated in order to determine the optimal parameters setting at the varying of the scenario. Then, the algorithm has been tested through extensive simulation campaign, proving it yields high reliability, high message propagation speed and reduced redundancy, thus approaching the performance bounds of the MCDS-based solution.

Moreover, the comparison of the protocol with other well-known position based schemes, has revealed good performance in different operating conditions.

However, the solution has also shown a not trivial dependency on the setting of some parameters, such as the contention window value. On the basis of this observation, we also proposed an approach to make the protocol more robust at the varying of the network conditions.

Finally, we observe that the proposed approach represents a promising strategy for efficient data dissemination when the time constraints are particularly tight as it guarantees good performance and, at the same time, it can be applied with minimal effort in realistic network configurations.



## **Part II**

# **Efficient Data Dissemination via Network Coding**



# 6

## Data Dissemination via Network Coding

---

*A goal without a plan is just a wish.*

(Antoine de Saint-Exupery)

In Part I we focused on the definition of data dissemination schemes which are efficient in terms of delay. Such protocols are suitable in those situations where time constraints are particularly tight. In different scenarios where delay-tolerant applications are implemented, different approaches can be applied in order to reduce as much as possible the number of transmissions required to deliver some amount of data, thus saving energy resources. This goal could be interesting in WSNs or in pervasive systems where the reduction of the traffic plays an important role in guaranteeing better performance. To achieve this aim we can adopt recent network paradigms which introduce in the network protocols some kind of intelligence to further increase the efficiency in data transmissions.

In this Part we focus on efficient data dissemination via network coding in wireless pervasive networks. Data dissemination is one of the most important services in such networks: it can be used to disseminate control data such as routing information, device status, performance metrics and so on. But data broadcasting could also be useful when a lot of users want to share some kind of data in a distributed way: downloading the same files, sharing knowledge about the network services and so on. These concepts apply to the so called networks of things where users can interact with the networks to find objects, gather information or drive actuators. In all these scenarios, similar information, usually distributed over the whole network, has to be exchanged by the users. Some of these application scenarios are described and discussed in [2]. The heterogeneity of the devices together with the wireless characteristics of the networks, make the definition of an efficient data dissemination scheme a challenging issue. The main aim here is to guarantee a limited usage of the network resources in order to save energy or increase the throughput. This translates in defining a data dissemination scheme which is able to deliver a lot of data at the minimum cost in terms of number of transmissions. We know that the transmission phase is particularly expensive for wireless devices and, in addition, high traffic leads to unsatisfactory performance due to the contention mechanisms used in wireless communications. Thus, reducing the number of transmissions to deliver data to multiple users is a very important goal in such networks.

A lot of studies, in literature, follow this direction by proposing different solutions. Recently, a

technique known as in-network data aggregation has been developed for wireless sensor networks [35]. It consists of some data processing along the network in order to combine packets and reduce the amount of information to be spread. However, since the data processing performed over the packets is generally lossy, in-network data aggregation is useful when nodes are interested on some average measures of the required information. For instance, it is particularly useful in wireless sensor networks for environmental monitoring to gather some information about the average temperature in a specific area and so on. This kind of approach, however, can not be applied for lossless data dissemination services.

However, the approach used by in-network data aggregation suggests us that most of the researches in energy efficient data dissemination goes in a specific direction: they aim to introduce some intelligence at the network layer in order to carry more information by transmitting the same number of packets, even coded in some way. An interesting issue is achieving this goal without loss of information: thus, one promising approach is *network coding* as it is one kind of lossless in-network data aggregation suitable to disseminate data to multiple destinations. In the following Chapters we study how to apply network coding paradigm in data dissemination in order to make the data delivery more efficient in terms of energy and throughput. We study the impact of network coding on the existing networking protocols, we develop novel data dissemination schemes based on network coding and finally we try to couple network coding with other important paradigms such as MIMO (*Multiple Input Multiple Output*) techniques.

The rest of the Part is organized as follows. Chapter 7 gives some background about network coding techniques. In Chapter 8, we study the impact of different MAC layers on several existing network coding data dissemination schemes. In Chapter 9, we define and analyze a proactive network coding algorithm which outperforms the existing strategies. In Chapter 10 we propose an alternative approach to jointly exploit network coding and MIMO techniques in an integrated system.

# 7

## Network Coding Background

---

*All truths are easy to understand once they are discovered; the point is to discover them.*

(Galileo Galilei)

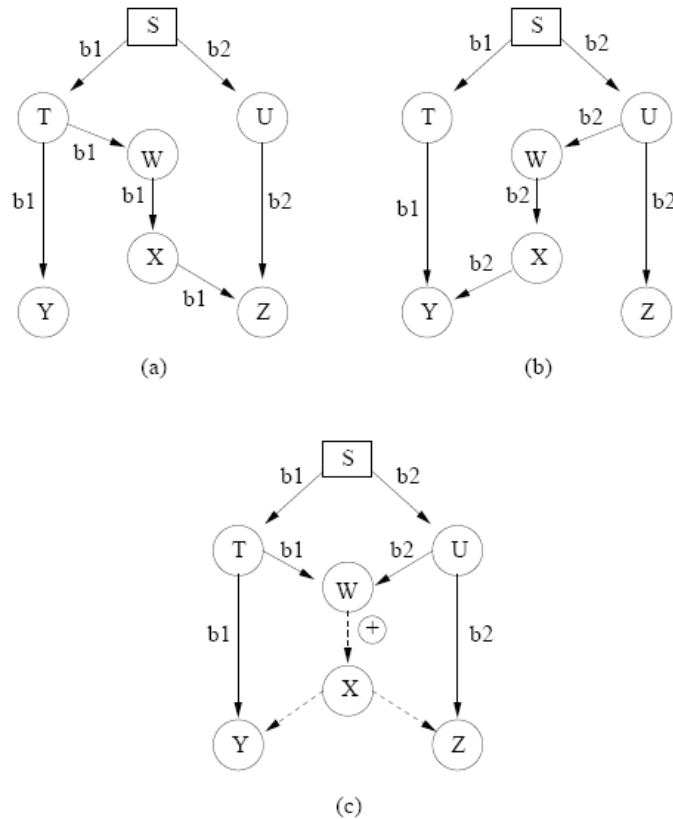
The concept of network coding is firstly introduced in [36] by Ahlswede et al. and it can be formulated in various ways at different levels of generality.

A simple definition is reported next:

Network coding is 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 throughput of the network.

In other words, it is a packet dissemination strategy which can be used to improve the throughput, thus guaranteeing high performance. In contrast to the store and forward paradigm, network coding implements a more complex *store, code, and forward* approach where each node stores the incoming packets in its own buffer, and, at transmission time, sends a *combination* of the stored data. To successfully decode, say,  $K$  packets, a node must collect  $K$  independent combinations. It can provide the highest gains in multicast or broadcast networks. More specifically, network coding can typically achieve higher transmission rates than separate unicast transmissions when information sources transmit to multiple destinations or to all nodes in the network.

The basic mechanism of network coding is introduced in [36] through a simple example which we report for reader's ease: Consider the *acyclic* network depicted in Fig. 7.1, with 6 nodes and a single source  $S$ . Time is subdivided in slots in a TDMA fashion and wireless links can carry a single flow at a time, i.e., either flow  $b_1$  or  $b_2$  can be transmitted during a time slot over a specific link. The problem to be solved is to send two bit flows  $b_1$  and  $b_2$  to both destinations  $Y$  and  $Z$  by exploiting the network at its maximum capacity. In other words, the flows are to be multicast to both destinations in the most efficient way. As a first attempt to solve the problem, one might devise a simple transmission schedule which consists of alternating two distinct transmission modes between even and odd slots as follows. In even slots we may let links  $TW$ ,  $WX$ ,  $XY$ ,  $TY$ ,  $XZ$  carry flow  $b_1$ , and links  $SU$  and  $UZ$  carry flow  $b_2$ . Hence, during even slots node  $Y$  receives flow  $b_1$  only, whereas node  $Z$  receives both flows. This



**Figure 7.1** Network coding: a first simple example.

transmission schedule is depicted in Fig. 7.1(a). During odd slots, instead, the situation may be reverted according to the transmission schedule in Fig. 7.1(b). In such a case, node  $Y$  receives both flows and node  $Z$  receives flow  $b_2$  only. Overall, this strategy leads to an average throughput of 1.5 flows per slot. A different way to attack the problem is provided by network coding and is shown in Fig. 7.1(c). This time, node  $W$  derives from flows  $b_1$  and  $b_2$  the exclusive-OR  $b_1 \oplus b_2$ . The link  $WX$  replicates  $b_1 \oplus b_2$  which is finally passed to both destinations  $Y$  and  $Z$ . Destination nodes can now decode both flows by re-applying the XOR operation. This strategy obtains a throughput of 2 flows per slot, which is the maximum achievable in the above problem. We observe that the exclusive-OR is a simple form of coding. If the same objective is to be achieved by store and forward techniques (simple replication of the incoming data flow), then at least two subsequent slots are needed to deliver  $b_1$  and  $b_2$  to both destinations. With coding, the two flows can instead be delivered to both  $Y$  and  $Z$  in a single slot. It is therefore apparent that coding, besides offering advantages in terms of throughput, may also decrease the latency. Note, however, that a drawback of the above example is that the coding/decoding scheme has to be agreed upon beforehand. That is, node  $W$  must know in advance that it has to process the

received flows by means of some coding strategy. While this is acceptable to present the advantages of coding by means of the above example, it has profound implications on the actual applicability of such a technique in distributed and therefore uncoordinated networks. These implications together with possible solutions will be explored later on in this Chapter.

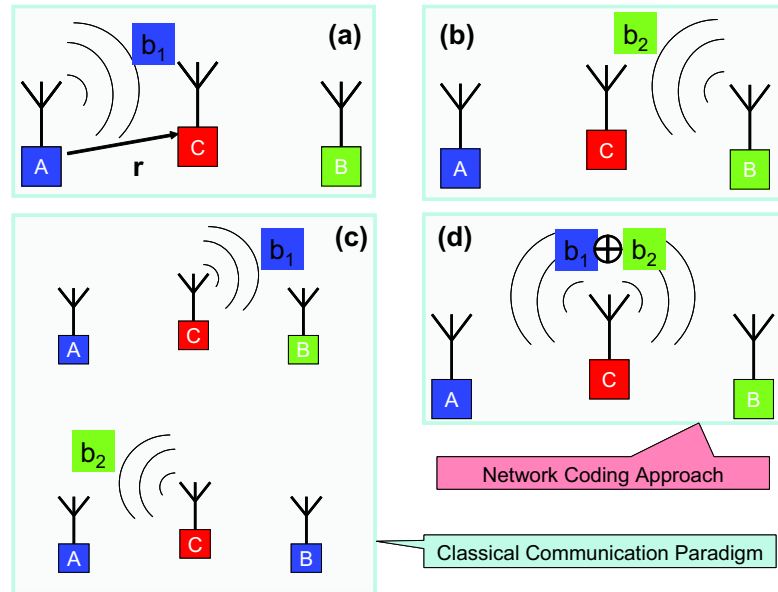
Hence, we can say that

Network coding may offer increased bit rates and decreased latencies with respect to separate transmissions.

The above concepts introduce a new way to define the capacity of a link. In more detail, we can think of the *transport network capacity* as the *physical capacity* of the link in terms of its bit-rate, which is different from the *information network capacity* which is instead the capacity of the link to transport innovative information. Referring to our above example, channel  $WX$  has a transport capacity of a single flow per time slot. However, if we transmit a coded information over such a link, then we obtain the higher transmission rate of 2 data flows per time slot. In other words, the information capacity with coding is higher.

To better explain this idea, we propose the additional example reported in Fig. 7.2 which is closer to the wireless ad hoc communications. It represents a two hops network where each of Node  $A$  and Node  $B$  have to exchange a data packet,  $b_1$  and  $b_2$ , respectively. Both the routing paths, namely the one from Node  $A$  to Node  $B$  and the other way around, go through Node  $C$  which has to forward two packets. The first transmission phase, i.e., both Node  $A$  and  $B$  send their own packets to Node  $C$ , is equivalent for the classical communication paradigm and for network coding (see Fig. 7.2a-b). Then, according to the classical communication paradigm, we need other two transmissions to deliver both  $b_1$  and  $b_2$  to the corresponding destination (see Fig. 7.2c). With network coding, instead, we can use only one additional transmission by allowing Node  $C$  to transmit the XOR-ed version of packet  $b_1$  and  $b_2$  to both Node  $A$  and  $B$  as reported in Fig. 7.2d. At the reception of this packet, Node  $A$  ( $B$ ) can decode packet  $b_2$  ( $b_1$ ) by subtracting its own packet from the received one. This example shows how network coding is particularly effective whenever there are overlapping data flows, because it can exploit both the broadcast nature of the channel and the coding process to simultaneously deliver different packets to multiple users.

After having briefly discussed the possible advantages offered by network coding, we now need to make things practical. In fact, it is actually infeasible to have pre-defined coding rules for every node in the network, as this would require full network knowledge. More than this, in distributed ad hoc scenarios, we need to cope with many constraints. For instance, nodes are not synchronized and, as a consequence, coding operation should not depend on time synchronization. Also, as messages travel through the network, they are exposed to delays (mainly due to processing at intermediate nodes and



**Figure 7.2** Network coding in multi-hop wireless communications.

channel contention mechanisms). Finally, too complex coding schemes are actually to be avoided as they might be infeasible for resource limited wireless devices. In the following, we discuss possible solutions to these issues.

## 7.1 Linear Network Coding

First of all, network codes that involve only linear mappings are of particular interest as they can be executed at low computational cost. When we refer to *linear network coding* [37] we intend that output flow of a given node is obtained as linear combination of its input flows. The coefficients of the combination are, by definition, selected from a finite Galois field. Moreover, the information traversing a non source node has the following property:

The content of any information flowing out of a set of non source nodes can be derived from the accumulated information that has flown into the set of nodes.

The principle of linear network coding can be simply described. Assume to have an acyclic graph  $(V, E)$  having unit capacity edges, a sender  $s \in V$  and a set of receivers,  $T \subseteq V$ . The broadcast capacity  $h$  is the minimum number of edges in any cut between the sender and a receiver [36]. Each edge  $e \in E$  emanating from a node  $v$  ( $v = in(e)$ ) carries a symbol  $y(e)$  that is a linear combination of the symbols  $y(e')$  on the edges  $e'$  entering in  $v$  ( $v = out(e')$ ). Formally:



$$y(e) = \sum_{e':out(e')=v} m_e(e')y(e'). \quad (7.1)$$

The *local encoding vector*  $\overrightarrow{m}(e) = [m_e(e')]_{e':out(e')=v}$  represents the encoding function at a node  $v$  along the edge  $e$ . In general, if  $v$  is the source node  $s$ , the artificial edges  $(e'_1, \dots, e'_h)$ , entering in  $s$  and carrying the  $h$  source symbols ( $y(e') = x_i, i = 1, \dots, h$ ) are introduced. This assumption is important because, using that, by working an induction argument, we can say that the symbols emanating from any node on any edge are a linear combination of the source symbols:

$$y(e) = \sum_{i=1}^h g_i(e)x_i, \quad (7.2)$$

where the vector  $\overrightarrow{g}(e) = [g_1(e), \dots, g_h(e)]$  is the *global encoding vector*. This vector can be recursively determined as:

$$g(e) = \sum_{e':out(e')=v} m_e(e')g(e'). \quad (7.3)$$

Thus, each received coded symbol  $y(e)$  can be considered by a node as the linear combination of the original symbols  $x_i$  where the coefficients belong to the global encoding vector and are elements of a Galois Field ( $GF$ ).

Reference [37] shows that linear codes are sufficient, in a multicast scenario, to considerably improve the throughput of the system. Further work, by Koetter and Médard, focus on finding the coefficients of linear encoding and decoding functions for a given network, by studying some of their properties as well as the time needed for their calculations. Most of the current work on distributed and asynchronous wireless ad hoc networks [38–40] exploit linear coding as, while retaining most of the benefits, are computationally efficient and hence do not affect the energy consumption much.

We now have to understand how to design, in practice, the codes and how to do that in a distributed fashion (thinking, for instance, of the case of a wireless ad hoc network). To this end, two different approaches are possible:

- Deterministic algorithms to build codes.
- Random techniques.

The first approach implies a centralized supervision: synchronization of the transmissions, definition and assignment of the encoding and decoding vectors and so on. The second approach, instead, allows to implement network coding in a completely distributed manner without the need for any kind

of collaboration/synchronization among nodes. Thus the latter could be a better choice for our wireless scenario and it is the focus of our research.

## 7.2 Random Network Coding

In [38], the authors propose a distributed scheme for practical network coding that obviates the need for centralized knowledge about the network topology, encoding/decoding functions and so on.

It shall be observed that most of the theoretical work on network coding assumes that symbols flow synchronously throughout the network and edges have integer or unit capacities. In real networks, however, information travels asynchronously in packets which are subjected to random delays and losses. Moreover, edges in general have unknown capacities. In addition, theoretical studies mainly deal with acyclic networks, but in real networks cycles abound (most of the edges are bidirectional). Finally, theoretical models of network coding do not consider the problems arising in having heterogeneous devices (in particular the difference among storage and processing capabilities, are not considered). All these reasons justify the need for developing a completely distributed scheme for network coding.

Referring to the network coding model, we can make some considerations about the decoding capabilities of the system. More in detail, let us consider a node  $t$  receiving along its  $h$  incoming edges  $e_1, \dots, e_h$  the symbols  $y(e_1), \dots, y(e_h)$ , where:

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}. \quad (7.4)$$

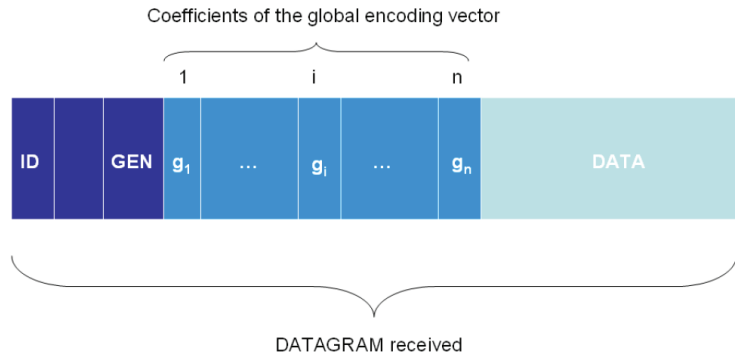
On the basis of this structure, it is possible to prove that

Node  $t$  can recover the source symbols  $x_1, \dots, x_h$  as long as the matrix  $G_t$ , formed by the global encoding vectors, has *rank*  $h$ .

The strength of random coding is that if the local encoding vectors are generated randomly, and the symbols lie in a finite field of sufficient size, the above property is true with an high probability, i.e., matrix  $G_t$  has full *rank* with high probability. For this reason, random network coding can be a powerful and desirable solution. The fundamental idea, which is presented in [38], consists of including within each packet flowing on the edge  $e$  the  $h$ -dimensional global encoding vector  $\vec{g}(e)$ . In Fig. 7.3 an example of packet format suitable to implement network coding is presented.

The cost of this solution is the overhead of transmitting  $h$  extra symbols in each packet which is, however, reasonable. The great advantages of such an approach can be summarize as follows:

- It is completely *distributed*.



**Figure 7.3** Packet format in network coding communications.

- It does *not* require any *topology knowledge*.
- It is *robust* to the packet loss or link failure.

The previously described scheme needs some improvements to work properly in a network where packets transmissions are not synchronous. In [38], this problem is solved introducing a *buffer model* and the concept of *generation*. In particular, all packets related to the same set of  $h$  source vectors are said to be in the same generation and  $h$  is referred to as the *generation size*. Furthermore, all packets in the same generation are tagged with the same *generation number*. Hence, packets arriving at a given node on any of its incoming edges are put into a single buffer and sorted by generation number. Only those packets belonging to the same generation can be coded together.

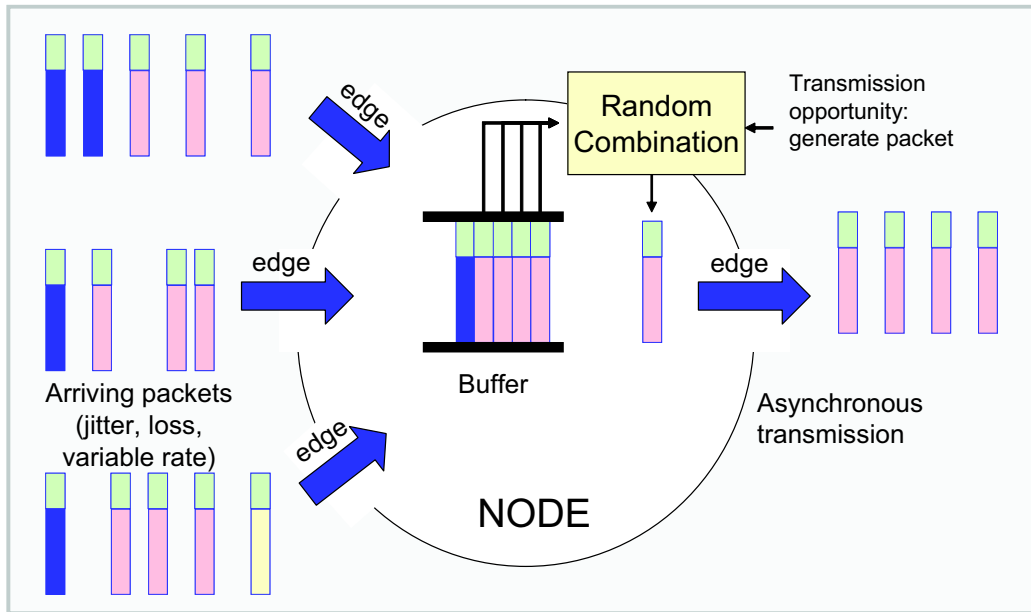
Moreover, packets can be classified into *innovative* or *non-innovative*, depending on whether they increase the rank of the global encoding vectors. We finally observe that one of the disadvantages of this scheme consists of the delay introduced in the decoding phase.

Summarizing, the complete network coding process is represented in Fig. 7.4 where a node receives multiple packets on different edges, stores them into a buffer and when a transmission opportunity occurs, it sends a random combination of packets belonging to the same generation.

In the rest of this thesis, when we refer to network coding (NC) we always mean the linear random approach introduced in this Section.

### 7.2.1 Data Dissemination via Network Coding

One of the most known case where random network coding can play a fundamental role to improve the system performance is in the broadcast communication scenario. References [39–41] present a lot of work in this context by comparing the performance, in terms of resilience to mobility, channel



**Figure 7.4** An example of the network coding process.

errors, throughput and delay, of network coding and traditional broadcasting schemes. Also, canonical network structures (circular network and rectangular grid) are considered from a theoretical point of view by quantifying the possible improvements. To this purpose, the ratio between the total number of transmissions required to broadcast one information unit to all nodes in the network using network coding against flooding is defined and used as a performance indicator to compare network coding with standard solutions. Some theorems show that for circular networks and rectangular grids the above ratio is  $1/2$  and  $3/4$ , respectively. In addition, in [39–41] the authors prove that schemes achieving this ratio exist. Algorithms for general networks are also proposed, even though they all assume the presence of a central supervisor.

### 7.3 More Complex Approaches

We conclude this Chapter by mentioning some further studies on the network coding technique. In [42], authors introduce the problem of network coding in settings where there is a cost associated with network operations. Costs may be related, for instance, to the energy consumed to transmit a packet over a link. The focus of this study is on the problem of minimum cost routing for multicast connections. Using network coding, this problem can be posed as a linear optimization problem that admits a distributed solution [42]. Another work [43] considers the problem of joint network coding and forward error correction. However, this issue is at its early stages and many issues remain to be

investigated. Further, reference [44] addresses the problem of content distribution of large files in large unstructured overlay networks using network coding. Authors compare network coding with other schemes that transmit unencoded information and, also, schemes in which only the source is allowed to generate and transmit encoded packets.

In general, a lot of work still has to be done, in particular towards defining *efficient* and *practical* schemes for network coding. In fact, while quite a few theoretical results are available, they often disregard practical and important aspects such as link failures, packet losses, delays, asynchronous transmissions. These issues are investigated in the following Chapters.



# 8

## MAC Layers and Packet Combination Strategies on Network Coding Data Dissemination

---

*Good judgment comes from experience, and  
experience comes from bad judgment.*  
(Barry LePatner)

We observed in Chapter 7 that most of the work done so far has focused on the theoretical aspects of network coding [45–48]. In fact, researchers only recently started to look at practical solutions to reap the full benefits of network coding techniques in actual network settings [49]. Initially, practical schemes were proposed for wired networks, where coding strategies were applied to peer-to-peer applications [44, 50]. In [38], one of the first examples of a simple practical solution for network coding, the authors focused on how the coding matrix as well as the information related to the random combination of packets can be shared by different nodes. This is a crucial aspect for network coding algorithms to work in actual networks. Further work can be found in [1, 40, 51]. A recent paper [1] focuses on unicast transmissions exploiting the network coding paradigm. In [1], it is experimentally shown that large gains, in terms of maximum throughput, are possible even in the case of unicast transmissions. The scheme presented in [51] jointly considers packet combinations with ARQ strategies for wireless sensor networks. In [40], the authors analyze and present some heuristics to combine the packets and prove the superiority of network coding with respect to flooding schemes in multi-hop wireless networks.

We note that, even if some valuable work has already been pursued, many practical aspects, such as the interaction between MAC schedules and network coding techniques, still need to be properly addressed. In our view, these practical aspects may limit the benefits achievable with network coding. Motivated by these needs, in this Chapter we present results on MAC schedules and packet combination rules. In particular, we show their impact on system performance and we propose a first improvement to better cope with collisions and suboptimal schedules. We consider a CSMA-like system affected by collisions, interference, and a random scheduling of the packets. In such a scenario, we test the behavior of network coding over simple wireless network configurations in order to capture the effects of each protocol component. Results are obtained using the ns2 network simulator, appropriately extended to include network coding functions.

## 8.1 Problem Description

In the rest of the Chapter we evaluate the impact of a realistic MAC and physical layer on random network coding in wireless ad hoc networks. In general, such networks are severely constrained by interference and channel impairments, especially in the case of broadcast communication. Consider, for instance, that each node is interested in retrieving information from all other nodes in the network. In this case, the use of traditional access mechanisms such as CSMA-like protocols would incur high contention on the wireless channel which, in turn, translates into a high number of collisions and of dropped packets. Network coding is a promising technique to increase system performance by reducing the number of transmissions and exploiting the random combination of data to increase transmission efficiency.

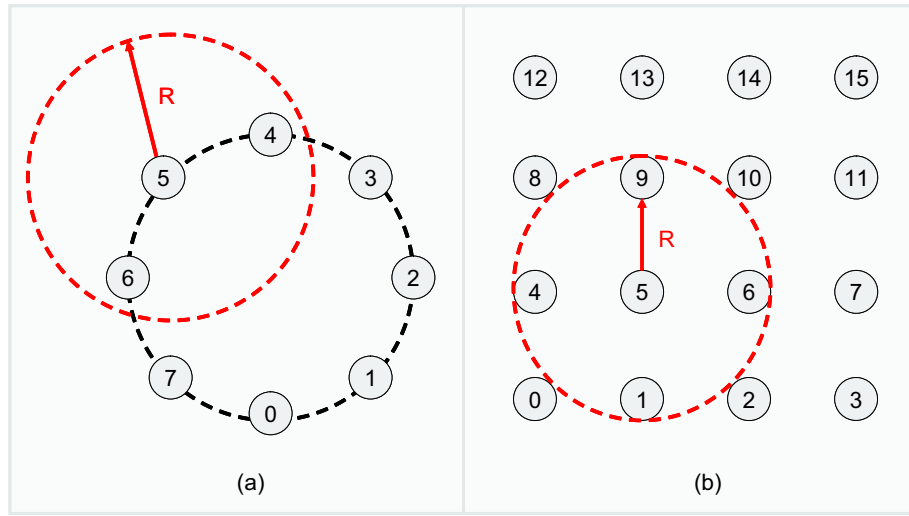
Our contribution differs from previous work in the following aspects. First, we systematically analyze the impact of the MAC protocol in use on the network coding performance. We note that previous studies [1, 49] address the problem of implementing network coding over actual MAC protocols. However, they lack a thorough analysis of their impact on network coding. Second, we focus on the broadcast communications paradigm rather than applying network coding to the case of unicast flows as we are interested in data dissemination services. Indeed, applying network coding to unicast flows can lead to important throughput gain as showed by [1] but we believe that it is in broadcast situations that network coding can solve most of the problems highlighted up to now. Finally, we look at network coding strategies which do not need any knowledge about the status of neighboring nodes, thereby requiring very little overhead.

In our opinion, two main factors are to be taken into account when using network coding as part of practical solutions for wireless ad hoc networks, namely:

- *Collisions*: collisions are a source of packet losses. It is important to understand their impact on the performance of network coding.
- *Packet Scheduling*: using random access will not create perfect transmission schedules. In fact, the number of neighbors, their traffic pattern, and their movement are not known a priori. Moreover, obtaining such information in order to build optimal transmission schedules is not convenient due to the large overhead involved. Understanding the impact of packet scheduling is another crucial point in the design of practical solutions.

In the following, we consider very simple network configurations to precisely understand and highlight the above issues. We start with the reference scenarios shown in Fig. 8.1. In the first network configuration, nodes are placed on a circular topology and each node has exactly two neighbors. In the second configuration, nodes have four neighbors and are placed on a grid. Finally, we also consider random





**Figure 8.1** Reference topologies for the performance evaluation.

networks, where nodes are randomly positioned within the simulation area. For the traffic pattern, each node inserts into the network a single original packet and is interested in collecting all the other inserted packets. Original packets are generated according to either a *random* or *deterministic* traffic pattern. In the former case, each node inserts its original packet independently by 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 original packets sequentially in each node. Subsequent insertions are separated by fixed time intervals of  $\Delta_2 = 1$  s. That is, the first original packet is inserted in Node 0 at time 0 s, the second one in node 1 at time 1 s, and so on. The reason for this generation strategy is twofold. First, if the interval  $\Delta_2$  is sufficiently large, the collision probability is sufficiently small. Second, if the transmission schedule is  $\{\text{Node } 0, \text{Node } 1, \dots, \text{Node } n - 1\}$  (see Fig. 8.1(a) where  $n = 8$ ) we obtain the scheme theoretically derived in [40], which was shown to achieve the maximum throughput in circular networks.

In the following, we describe the schemes and the algorithms we analyzed. In addition, we propose a solution which improves the network coding performance in wireless ad hoc networks.

## 8.2 MAC Protocols

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

### 8.2.1 IEEE 802.11b

We consider IEEE 802.11b as the baseline medium access protocol. Note that the network coding strategies we examine are based on broadcast transmissions. Hence, we adopt the basic access provided by IEEE 802.11b which, in the broadcast case, does not use any acknowledgment mechanism. As a consequence, in case of a collision, no retransmission occurs and the packet is lost, resulting in high inefficiency and low packet delivery ratio.

### 8.2.2 IEEE 802.11b with Pseudo Broadcast [1]

This scheme is an improvement of the basic IEEE 802.11b, where an acknowledgment mechanism is implemented. According to the idea proposed in [1], a given node first broadcasts a packet to its neighbors, by randomly picking one of them and including its address in the packet header. Only the node whose address matches the one contained in the header sends an acknowledgment to the sending device. This is done according to the basic IEEE 802.11b unicast communication (no RTS/CTS). All other neighbors overhear/decode the transmission but do not respond to the sending node. The packet is retransmitted in case there is no acknowledgment. Note that, using this mechanism, only collisions at the addressed receiver can be detected, while collisions occurring at any of the remaining neighbors are ignored. Also, this strategy does not solve the hidden node problem.

### 8.2.3 IEEE 802.11 with Pseudo Broadcast and RTS/CTS Handshaking

To further improve the packet delivery ratio, we propose to consider the previous scheme with additional RTS/CTS handshake. These control messages are introduced to alleviate the hidden node problem. The CTS is only transmitted by the node addressed in the packet header. The delay introduced by this technique is expected to be higher, due to the additional control packets. Moreover, as for the previous schemes, we can not detect collisions at all overhearing nodes.

### 8.2.4 Ideal MAC

With the term ideal MAC we refer to a very simple mechanism where transmitted packets are only affected by the transmission delay,  $\Delta_{tx}$ . That is, we can assume to have an omniscient entity which regulates the transmissions in order to avoid interference and collisions. This means that, as a node sends a packet, all its neighbors successfully receive the message after a (fixed) transmission delay.  $\Delta_{tx}$  is computed using the same rate and packet size of the above MAC protocols. This scheme, which is not feasible in practice, is analyzed to obtain an upper bound on the achievable performance. Such an upper bound is used as a benchmark for the other solutions.

## 8.3 Network Coding Strategies

The core of the network coding strategies we use are the same as the ones proposed in [38]. These are based on random linear coding where the coefficients of the combination are included in each transmitted packet. In addition, we implement three different techniques for combining packets. The first two are inspired by the work in [40], while the last one is a new proposal.

In [40], it was shown that network coding allows to reduce the number of transmissions, with respect to pure store and forward, for a certain targeted packet delivery ratio. The achievable reduction in the number of messages generally depends on the number of neighbors. For instance, if there are 2 neighboring nodes, network coding over circular networks halves the number of transmissions needed to achieve a packet delivery ratio equal to one. A node does not need to transmit a new packet at each reception of an innovative message. This is the basic idea of the network coding algorithms proposed in [40]. In the following, we detail the packet combination strategies considered in this Chapter. All the presented schemes are characterized by a design parameter, named *forwarding factor*,  $\rho$ , which is defined as the ratio between the number of packets transmitted and the number of innovative packets received, per node. It determines the average number of packets that each node can transmit.

### 8.3.1 Probabilistic Network Coding

This approach exploits random linear coding. Each node sends a random linear combination of the packets in its buffer, as discussed in Chapter 7. Only the reception of innovative packets carries additional information. Hence, non-innovative packets can be discarded. With probabilistic network coding, when a node receives an innovative packet, it makes a decision as to whether a new random combination should be transmitted or not. Specifically, upon the reception of an innovative message, a new combination is transmitted with probability  $p$  by assigning to the *forwarding factor*  $\rho$  the value of  $p$ . For  $\rho = 0.5$ , a node on average sends a new message for every two innovative packets received. As per our discussion above,  $\rho = 0.5$  would theoretically (ideal scheduling, no collisions) assure a packet delivery ratio equal to 1 when the number of neighbors is 2.

### 8.3.2 Semi-deterministic Network Coding

This strategy is quite similar to the previous scheme. In this case, for a given forwarding factor  $\rho$ , 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 for every two received innovative packets. The forwarding factor, in this case, is not related to a probability, but it is rather used as a threshold on the number of incoming messages.

### 8.3.3 Timed Network Coding

The two previous schemes have two major drawbacks. The first drawback is that they are particularly sensitive to packet losses due to, e.g., collisions, as shown in the Section of the results. In fact, if one of the transmitted packets is lost, the propagation of the information through the network could be interrupted. To better illustrate this, let  $\rho < 1$  be the forwarding factor in use. In this case, for a given targeted packet delivery ratio, we can reduce the number of new combinations transmitted. The effect of such an operation is to increase the transmission efficiency at the expense of a higher sensitivity to packet losses. 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. As the timer expires, the node decides to send out a new random combination with probability  $p = \rho$ . The timer,  $\tau$ , is a uniform random variable in  $[0, \tau_{max}]$ . The main advantages of this timing approach are twofold. First, it facilitates packet mixing, thus reducing the likelihood of transmitting non-innovative packets. Without the timer, indeed, some of the nodes that receive an innovative packet might decide to simultaneously send out a new packet combination. This can lead to a non innovative transmission, especially when the buffers are almost empty. With the introduction of a waiting interval before coding, 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 reducing 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 long enough to allow the collection of more than one packet, which translates to selecting  $\tau_{max} \approx 10 - 30$  ms. In the rest of the Chapter we consider  $\tau_{max} = 20$  ms.

## 8.4 Simulation Results

In this Section, we report the most relevant results obtained via ns2 simulations. All presented schemes are evaluated over the simple topologies introduced in Section 8.1, taking into account the *random* and the *deterministic* traffic patterns. We tested the algorithms varying the *forwarding factor*  $\rho$  from 0.1 to 1 and the number of nodes in the network,  $n$ . For the circular topologies, we considered  $n \in \{4, 8, 12, 16\}$  and for grid configurations  $n \in \{9, 16, 64\}$ . Regarding the MAC layer, we considered a transmission rate of 1 Mbps. Each packet has an extra overhead which is accounted for to transmit the

coefficients of the random combinations [38]. As discussed in [38], such an overhead is tolerable for practical cases. In the following, we define the performance metrics we look at in our investigation:

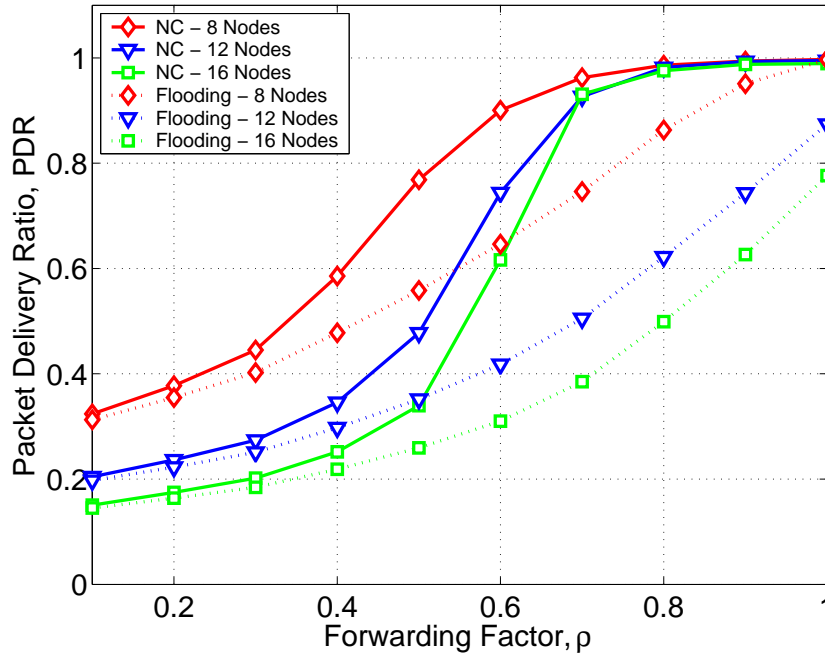
- *Packet Delivery Ratio, PDR*: is defined as the ratio between the number of successfully received (and decoded) packets and the number of packets in which a node is interested, averaged over all nodes.
- *Packet Delivery Delay, D*: is the average time between the first transmission of a packet and its reception and successful decoding at the destination nodes.
- *Protocol Overhead*: is the ratio between the number of transmitted packets ( $Pkt_{tx}$ ) at the MAC layer and the number of successfully decoded packets ( $Pkt_{dcd}$ ). This value depends both on the adopted MAC protocol and on the efficiency of the network coding strategy. For example, we expect that IEEE 802.11b pseudo broadcast with RTS/CTS, will show higher protocol overhead compared to other MAC schemes. On the other hand, timed network coding should decrease the protocol overhead by suppressing unnecessary transmissions. Note that this metric gives us a measure of the energy consumption as well.
- *Collision Ratio*: is the number of collided packets at the receiver ( $Pkt_{col}$ ) over the total number of received packets ( $Pkt_{rcv}$ ). Observe that in a broadcast wireless environment, tracking the number of collisions could be a problem. In fact, the same packet may collide only for a subset of the receiving nodes. For this reason, we evaluate the number of collisions at each receiver.

In the simulations, we compare the network coding strategies introduced in Section 8.3 against each other and against probabilistic flooding. This is done to point out the possible benefits of the network coding paradigm with respect to standard store-and-forward. The probabilistic flooding considered in this case uses a forwarding factor  $\rho$ , which is simply the probability of forwarding a new incoming packet.

We organize our performance analysis in two parts. In Section 8.4.1, we evaluate the impact of different MAC protocols on network coding whereas, in Section 8.4.2, we focus on different packet combination strategies.

### 8.4.1 The Impact of MAC Protocols

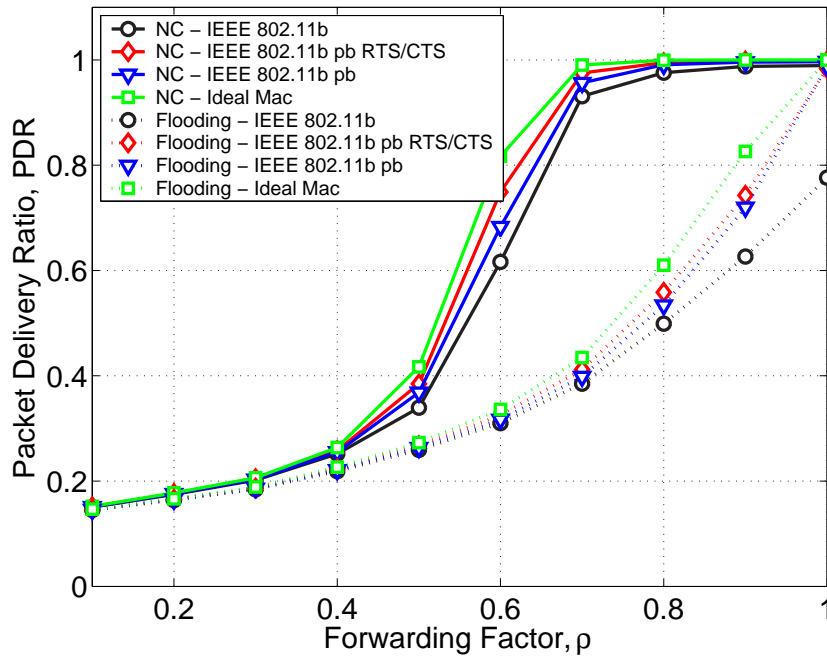
We start the performance analysis with Fig. 8.2 and Fig. 8.3, where we compare probabilistic network coding (solid lines) against probabilistic flooding (dotted lines) in a circular network topology. Fig. 8.2 shows the packet delivery ratio vs.  $\rho$  for different numbers of nodes,  $n$ . For all forwarding factors  $\rho$ , network coding outperforms probabilistic flooding. As an example, for  $n = 12$  and  $\rho = 0.6$ ,



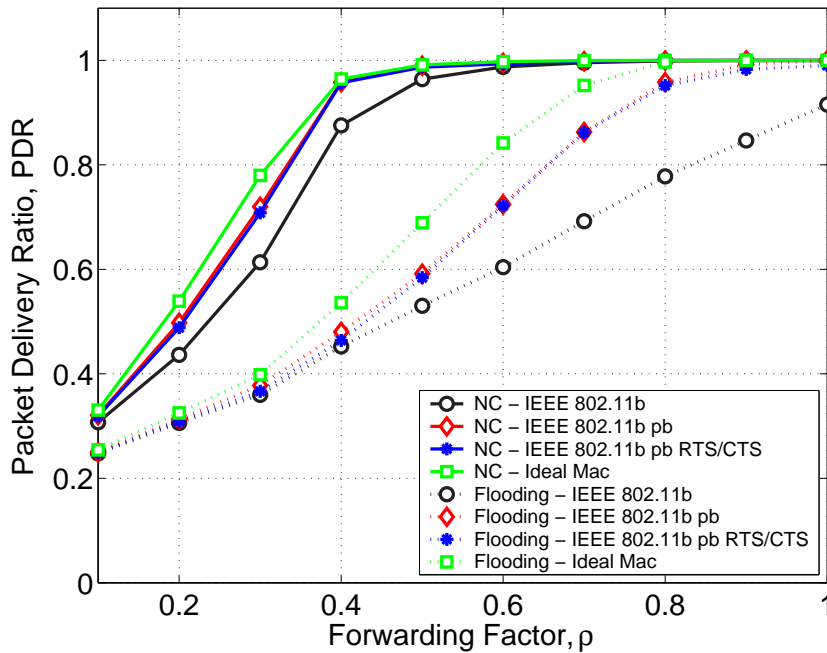
**Figure 8.2** Performance of *Probabilistic network coding* and *Probabilistic Flooding* in circular networks – Several network sizes and IEEE 802.11b

network coding achieves  $PDR \approx 0.75$ , whereas  $PDR \approx 0.42$  for the flooding scheme. This gain increases with increasing  $n$  as well as with increasing  $\rho$ . In addition, network coding with  $\rho = 1$  always results in a  $PDR$  very close to one. This is not true for flooding, which is considerably impacted by packet losses. As observed above, for this topology a  $PDR$  equal to one is theoretically achievable by setting  $\rho = 0.5$  [40]. However, we observe from Fig. 8.2 that this performance level is never reached in practice and that the actual  $PDR$  depends on the network size  $n$ . These effects are 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. In Fig. 8.3 we focus on circular topology with  $n = 16$  (worst case in Fig. 8.2) and we look at the impact of the MAC layer on both probabilistic network coding and flooding. Once again, we observe the superiority of network coding. Moreover, we can evaluate the importance of the MAC scheme in use. For  $\rho = 0.6$ , IEEE 802.11b achieves  $PDR \approx 0.6$ , whereas an ideal MAC achieves  $PDR \approx 0.8$ . This corresponds to a 25% lose of packet delivery ratio for the real MAC with respect to the ideal case. On the other hand, for this value of the forwarding factor a perfect schedule leads to full packet delivery ratio. 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.

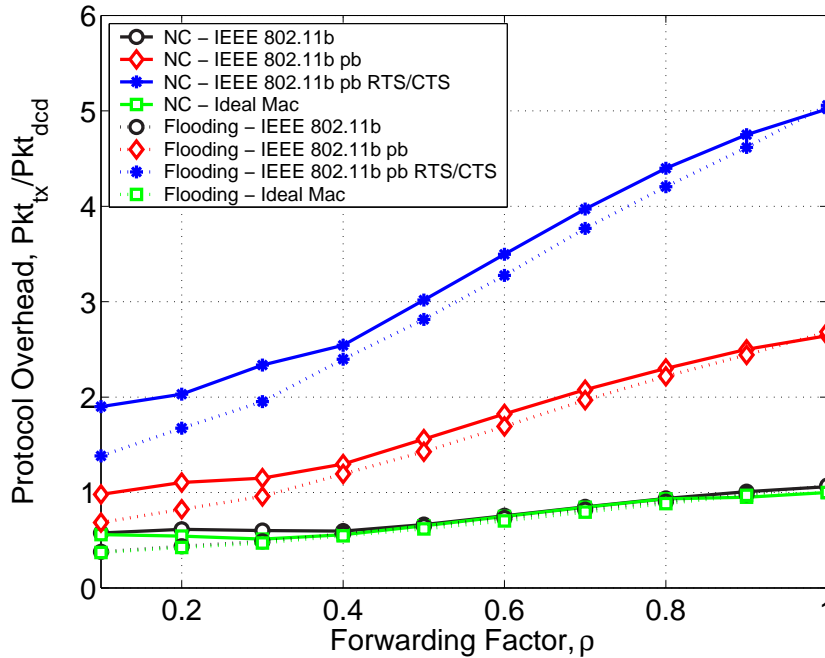
Even though circular networks are a simple reference scenario, useful to easily capture network



**Figure 8.3** Performance of *Probabilistic network coding* and *Probabilistic Flooding* in circular networks – Network size  $n = 16$  and different MAC protocols



**Figure 8.4** Packet Delivery Ratio: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with  $n = 16$ .

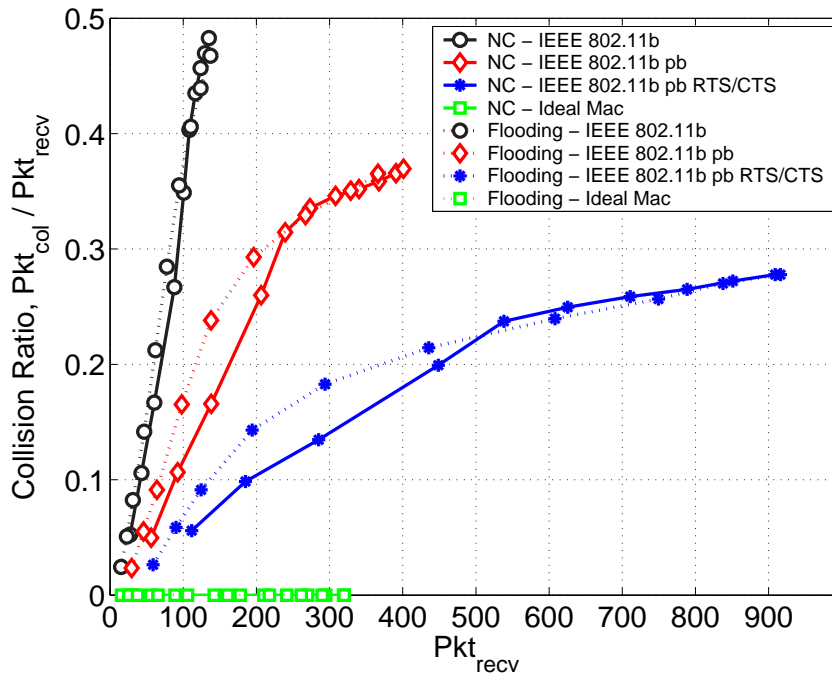


**Figure 8.5** Protocol Overhead: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with  $n = 16$ .

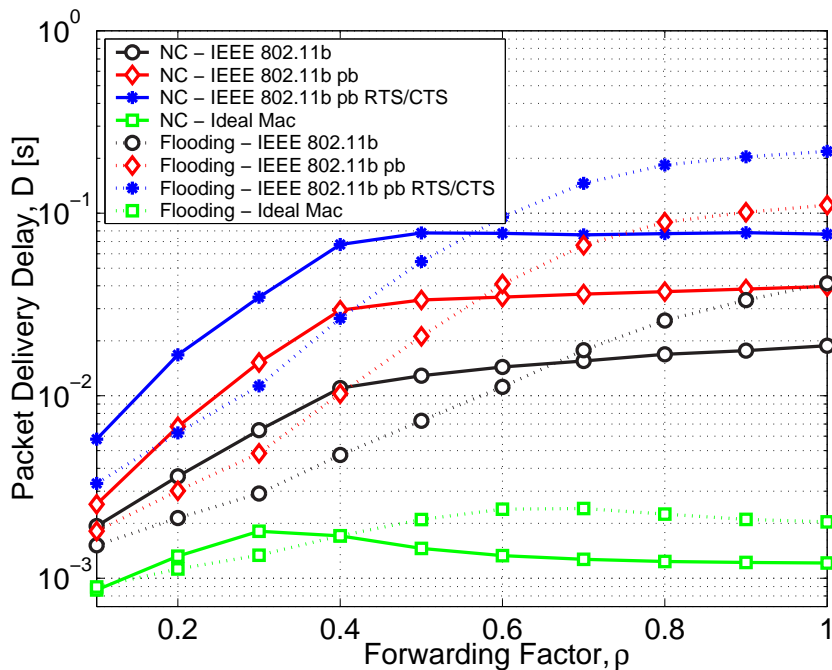
coding behavior, we focus now on a more realistic setting where nodes are placed over a grid (see Fig. 8.1b). We consider here only grid networks with  $n = 16$  in order to directly compare them with the circular case. However, in our simulations, we noticed the same behavior also for different network sizes. In Fig. 8.4, we show the impact of different MAC protocols on the packet delivery ratio of probabilistic network coding and flooding. As expected, the achieved performance is better than in the circular case due to the higher number of neighbors (4 instead of 2), which favors packet mixing and dissemination. Also in this scenario, the presence of realistic MAC layers reduces significantly the packet delivery ratio for a given value of  $\rho$ . In addition, Fig. 8.5 shows the protocol overhead vs.  $\rho$  for each MAC protocol. It is noted that the schemes implementing collision avoidance policies (i.e., IEEE 802.11b with pseudo broadcast and IEEE 802.11 with pseudo broadcast and RTS/CTS handshaking) improve the packet delivery ratio 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, RTS and CTS packets). In addition, we note that when we compare probabilistic network coding and flooding performance against  $\rho$ , we have a fair comparison as, given a specific  $\rho$  and a fixed MAC protocol, both network coding and flooding lead to very close protocol overhead.

Fig. 8.6 shows the effectiveness of pseudo broadcast and pseudo broadcast with RTS/CTS in decreasing the number of collisions: for a given value of the  $Pkt_{col}/Pkt_{recv}$  ratio the number of received

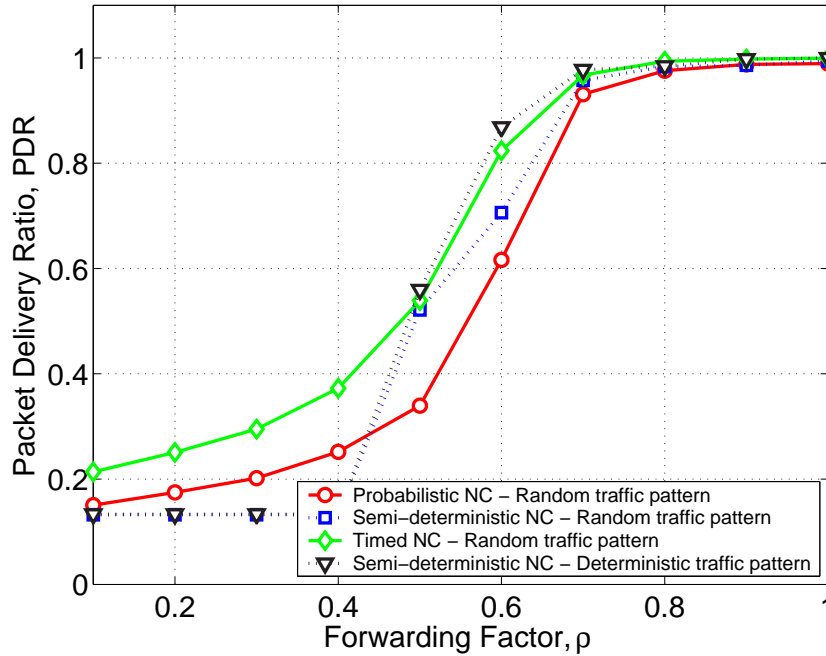




**Figure 8.6** Collision Ratio: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with  $n = 16$ .



**Figure 8.7** Packet Delivery Delay: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with  $n = 16$ .



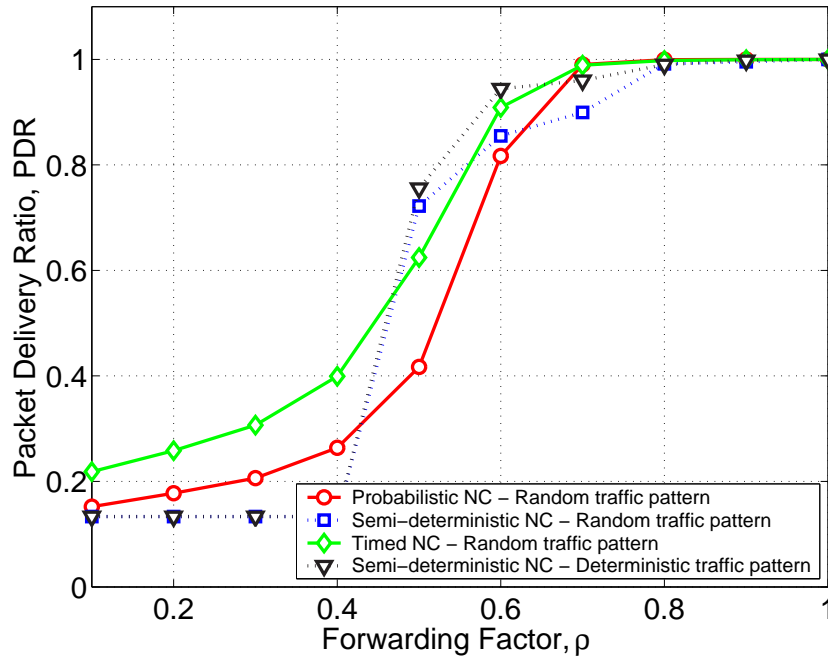
**Figure 8.8** Performance comparison of different combination strategies in circular networks with  $n = 16$  and MAC IEEE 802.11b.

packets ( $Pkt_{recv}$ ) in pseudo broadcast with RTS/CTS is the highest. This is due to both the higher  $PDR$  of the scheme and, mostly, the additional retransmissions caused by the acknowledgments. On the downside, using additional techniques to recover from packet loss leads to longer delays, as can be seen from Fig. 8.7. The average delay increase is about one order of magnitude in the worst case (pseudo broadcast with RTS/CTS). We also note that network coding always outperforms the flooding scheme when using the same MAC and that its delay stabilizes for increasing  $\rho$ . The reason for the stabilization and even decrease in delay is that for increasing  $\rho$  (beyond a given  $\rho^*$ ),  $PDR$  remains close to one but the number of innovative packets flowing in the network continues to increase. This has the effect of allowing earlier decoding.

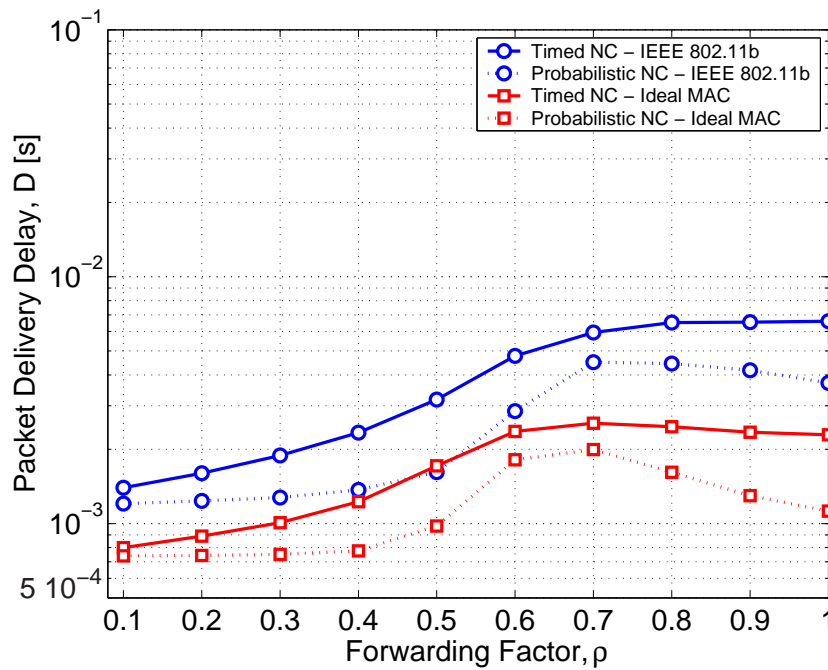
To summarize, we observe that the presence of actual MAC protocols reduces network coding performance in terms of packet delivery ratio. This performance reduction depends on the network size, contrary to what happens for the ideal MAC case. In addition, collision avoidance policies give little improvement in terms of packet delivery ratio, while leading to poor overhead and delay performance.

#### 8.4.2 Different Packet Combination Strategies

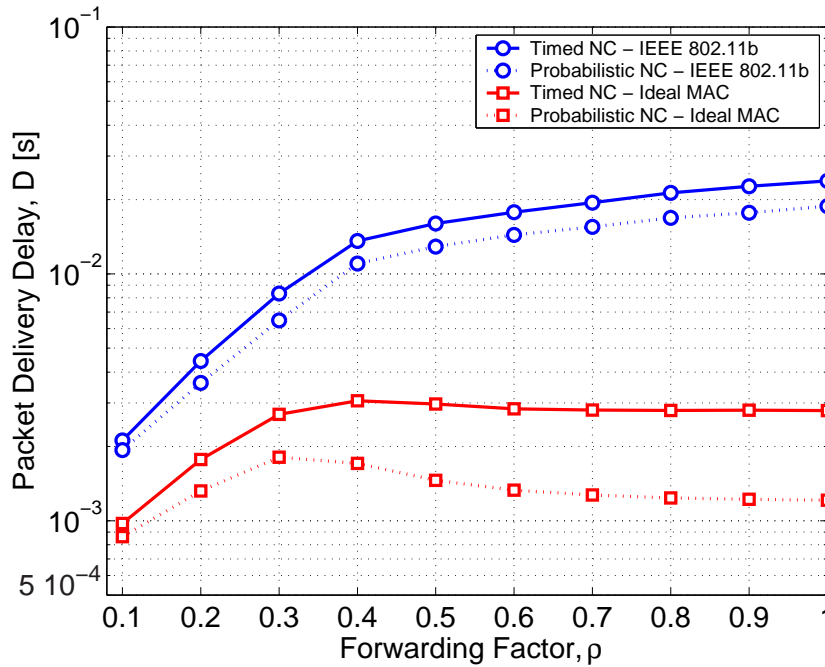
We now evaluate the impact of the network coding schemes described in Section 8.3. Fig. 8.8 and Fig. 8.9 show the packet delivery ratio performance for a circular network with  $n = 16$  by varying the



**Figure 8.9** Performance comparison of different combination strategies in circular networks with  $n = 16$  and ideal MAC.



**Figure 8.10** Circular Networks: Performance comparison of Packet Delivery Delay for *Probabilistic* and *Timed Network Coding* using IEEE 802.11b and Ideal MAC for  $n = 16$ .



**Figure 8.11** Grid Networks: Performance comparison of Packet Delivery Delay for *Probabilistic* and *Timed Network Coding* using IEEE 802.11b and Ideal MAC for  $n = 16$ .

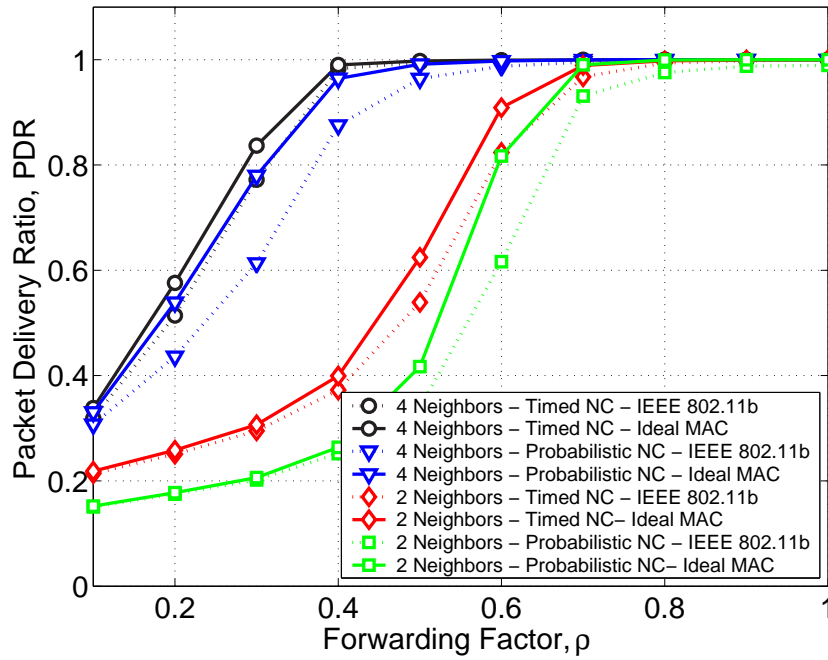
packet combination strategy (for a fixed MAC protocol). In Fig. 8.8 we use an IEEE 802.11b MAC with both the semi-deterministic and the probabilistic combination methods. The semi-deterministic schemes (dotted lines) show a sudden phase change, where  $PDR$  remains constant up to  $\rho^* = 0.4$  and then suddenly increases for higher forwarding factors. 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  $\rho^*$  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 in probabilistic and timed network coding, whose sending rules are based on probabilities rather than on hard thresholds. Notably, 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  $PDR \approx 0.35$ , whereas timed network coding leads to  $PDR \approx 0.55$ , which corresponds to an improvement of about 36%. The same considerations hold for Fig. 8.9 with the only difference that in this case we adopt an ideal MAC. The performance is thus rescaled accordingly. As can be seen from a direct comparison of the two figures,

the impact of MAC on packet delivery ratio performance is smaller than that of the packet combination strategy in use. We also note that, in Fig. 8.9, for  $\rho = 0.5$   $PDR$  is higher than 0.8 but is still strictly lower than one (theoretical bound), even if we use an ideal MAC and a deterministic scheduling. This is due to the fact that our deterministic scheduling approach is only an approximation of the ideal scheme in [40].

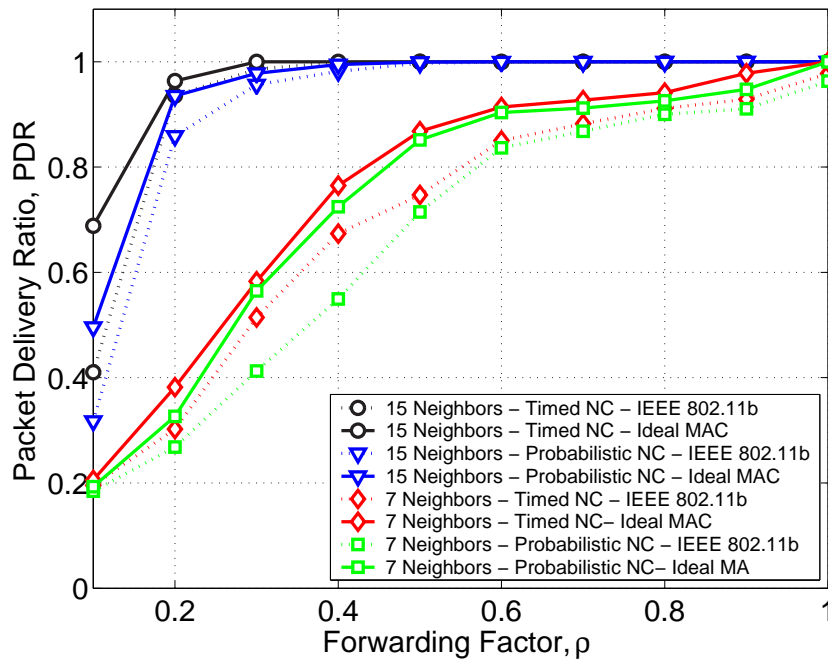
Timed network coding is further evaluated in Fig. 8.10, where we plot the performance in terms of delay. We observe that the timed strategy introduces an additional delay. Also, there are some expected differences between ideal and actual MAC. For IEEE 802.11b, the delay increase is reasonably small (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 in terms of packet delivery ratio than pseudo broadcast, by leading to similar extra-delays. For this reason, the timed scheme may make sense when the goal is to maximize the packet delivery ratio (throughput) by accepting some delay degradation.

The delay in the grid network (four neighbors per node) scenario is plotted in Fig. 8.11: the impact of the adopted MAC is more pronounced than for circular networks. This means that the importance of MAC is higher when the number of neighbors increases (circular  $\rightarrow$  grid scenario) as said before. On the other hand, we note that the gap between timed and probabilistic schemes is smaller than in Fig. 8.10.

The performance analysis is continued in Fig. 8.12, where probabilistic network coding and timed network coding are compared for the cases of two and four neighbors. The timed combination strategy outperforms probabilistic network coding with plain IEEE 802.11b by about 30% for  $\rho \approx 0.3$ . Also, we observe that the gap between ideal and actual MAC is tighter when the number of neighbors is four (grid networks). This, together with the result in Fig. 8.11, suggests that the timed strategy becomes more effective with an increasing number of neighbors. In Fig. 8.13, we report  $PDR$  for ideal/actual MACs and probabilistic/timed strategies in random networks with node densities of 7 and 15 neighbors per node (selected as representative of different density scenarios). Only connected topologies were considered to obtain this plot. Similarly to what observed earlier,  $PDR$  increases with increasing node density (7  $\rightarrow$  15). We stress that, in an ideal grid scenario,  $PDR \rightarrow 1$  as  $\rho$  approaches the inverse of the number of neighbors. If this were true for random networks, in Fig. 8.13 for, e.g., 15 neighbors we should get a  $PDR = 1$  when  $\rho \approx 0.06$ . However, this is not verified for two reasons: first, random networks are not uniform in the sense that some nodes have more neighbors than others; second, a probabilistic forwarding policy cannot get to the expected performance, which is instead achievable using an ad hoc deterministic scheduling (see [40]). For the 7-neighbor case, with ideal MAC we can still get  $PDR = 1$  by properly tuning  $\rho$ . This is, however, not the case when an actual MAC is used, where even  $\rho = 1$  does not suffice to get  $PDR = 1$ . Finally, we can note that the gain achieved by the



**Figure 8.12** Random Networks: Performance comparison of Packet Delivery Ratio for *Probabilistic* and *Timed Network Coding* for ideal and actual MAC.



**Figure 8.13** Random Networks: Performance comparison of Packet Delivery Ratio for *Probabilistic* and *Timed Network Coding* for ideal and actual MAC.

timed strategy against the probabilistic scheme remains significant. In general, random networks are impacted by the very poor performance of the dissemination procedure in the proximity of the nodes with low degree (small number of neighbors). For these nodes the probabilistic approach does not work properly and the information flow (new innovative packets) is likely to be stopped.

Our results show that even slight modifications to the packet combination strategy may lead to considerable performance improvements. In light of this, directions for future research include strategies to exploit some knowledge about number of neighbors and coding state of nearby devices, in order to efficiently handle packet forwarding at nodes with low degree. This should help coping with the lack of regularity exhibited by random networks.

## 8.5 Analysis of Adaptive Packet Combination Strategies

In previous Sections we always consider the probabilistic network coding strategy as the reference protocol. According to it, all nodes fix the same forwarding factor. Our aim was to analyze the network coding behavior versus the forwarding factor. Nevertheless, it is intuitive that when a node has a lot of neighbors, it is sufficient for him to receive few independent packets from each neighbor to decode all the original data. On the contrary, if a node has a single neighbor, it needs to receive at least  $K$  packets from that neighbor to decode  $K$  original data. This means that, in populated areas, nodes need to use low forwarding factors as the data is forwarded anyway. In low-density areas, instead, nodes have to utilize high forwarding factors to guarantee that all nodes can receive enough packets to decode the original data. This is the basic idea of the *adaptive forwarding factor strategies* studied in this Section. The core of such schemes is that each node independently selects its forwarding factor on the basis of local topology knowledge gathered in its neighborhood. The aim is to further reduce the number of transmissions required to disseminate data by tuning the forwarding factor according to the specific network configurations. We introduce in Section 8.5.1 the adaptive strategies we take into account by giving some motivations. Then, we present the performance analysis of such schemes.

### 8.5.1 Adaptive Send-count Approaches

In this Section we consider adaptive random network coding schemes. The main aim is to reduce the amount of packets to be transmitted in order to achieve full reliability. The packet combination strategies considered here are based on the use of the *adaptive forwarding factor*. It means that, in a distributed way, each node decides its own forwarding factor on the basis of different metrics.

The basic idea of these metrics is that when a node transmits an innovative packet to its neighbors, only few of them have to send out a new combination in order to disseminate the new innovative

contribution. The open issues is how to decide which nodes have to participate to the forwarding phase. An idea could be simply based on a probabilistic approach, other schemes could be based on the distance from the current source of the packets. Node farther from the current source are most suitable to disseminate the information where it is still missing. For these reasons, we define the following approaches:

- *One-Hop-Neighbor-based Forwarding Factor*: this strategy is similar to the basic probabilistic network coding. The forwarding probability  $\rho(x)$  that node  $x$  sends out a new packet combination at the reception of an innovative packet varies from node to node. Specifically it depends on the number of  $x$ 's neighbors according to the following heuristic:

$$\rho(x) = \frac{k}{n(x)}, \quad (8.1)$$

where  $\rho(x)$  is the forwarding probability of node  $x$ ,  $n(x)$  is the number of neighbors of node  $x$  and  $k$  is a parameter chosen by the user. This heuristic immediately derives from the above considerations for regular networks. If we assume a uniform node density, each node sends  $1/n(x)$  packets, on average, but it can receive the whole information by collecting packets from all its neighbors. The problem is that, in general, the node density varies from one location to another so that to assure that each node can decode all the information wants we have to introduce some redundancy given by the  $k$  factor.

- *Two-Hop-Neighbor-based Forwarding Factor*: to better catch the node distribution, instead to consider the 1-hop neighbors of a node, we can focus on the 2-hop neighbors. In this case the forwarding probability  $\rho(x)$  is given by:

$$\rho(x) = \frac{k}{\min_{v \in n_2(x)} n(v)}, \quad (8.2)$$

which depends on the minimum number of two-hop neighbors of the node  $x$ . In this way we can speed up the information dissemination also in those areas where the node density is not uniform and especially where some nodes have very few neighbors.

The strategy based on the two hop neighbors can be specified by the use of different metrics. Other two examples of its implementation can be represented by the following expressions:

$$\rho(x) = \frac{k}{\min_{v \in n_2(x)} n_v(x)}, \quad (8.3)$$

$$\rho(x) = \frac{k}{n_{2m}(x)}, \quad (8.4)$$

where, considering each two-hop neighbor ( $v \in n_2(x)$ ) of  $x$ , we evaluate the number of neighbors of both  $v$  and  $x$  ( $n_v(x)$ ) and take the minimum value.  $n_{2m}(x)$  is the average number of 2-hop



neighbors. The basic idea of the last strategy is the same of the previous one but here we take the average number instead of the minimum to try to reduce the overhead. However, we expect similar performance.

- *Transmitter-based Forwarding Factor*: in this case the forwarding probability  $\rho(x)$  at node  $x$  is given by:

$$\rho(x) = \frac{k}{n_{src}}, \quad (8.5)$$

where  $k$  is defined by the user,  $src$  is the current transmitter node of the packet received by  $x$  and  $n_{src}$  is the number of  $src$ 's neighbors. In this way, the number of nodes which retransmit the information are on average equal to  $k$ .

- *Position-based Forwarding Factor*: the basic idea of the approach based on the node position is to use the information about the distance of a node from the current source to decide if forwarding a new packet combination or not. In particular, nodes farther away from the current source should have a higher probability to forward new packets. This could lead to two benefits: i) outermost nodes have a higher chance to send a packet which is innovative for a greater number of nodes; ii) simultaneous transmissions of outermost nodes could generate less interference. According to these observations, we develop two strategies based on both node positions and on the number of neighbors.

In the *Simple Mixed Forwarding Factor* we propose a heuristic where the forwarding probability  $\rho(x, r)$  at node  $x$  depends on the number of neighbors of the current source  $src$  and on the distance  $r$  of  $x$  from  $src$ . It is given by:

$$\rho(x, r) = \left( \frac{k}{n(src)} \right) \left( 1 + \frac{(r - R/2)}{R} \right), \quad (8.6)$$

where  $R$  is the transmission range, that is supposed fixed and equal for each node. According to Eq. 8.6, a node farther from the current source has a higher probability to transmit a new combination than the other nodes. As an example, let us to consider a coverage area of range  $R$ : nodes at distance  $r = R$  from the current source have a probability equal to  $\frac{k}{n(src)}$  multiplied by a factor of 1.5, nodes in the middle have a probability equal to  $\frac{k}{n(src)}$  whereas nodes very near to the current source have a probability equal to  $\frac{k}{n(src)}$  multiplied by a factor of 0.5. This works well in case of uniform distribution of nodes. In addition, we can note that, if there is an area where all nodes have a short distance from the current source, it is possible that no node decides to transmit a new combination and the information dissemination could be blocked.

The second strategy is named *Mixed Forwarding Factor*. Also in this case the forwarding probability  $\rho(x, r)$  at node  $x$  depends on both the number of neighbors of the current source  $src$  and

the distance  $r$  of  $x$  from  $src$ . It is given by:

$$\rho(x, r) = \frac{k3r}{2NR}. \quad (8.7)$$

This strategy differs from the *Simple Mixed Forwarding Factor* as, here, the factor  $\frac{k3r}{2NR}$  is normalized in such a way that the average number of nodes which send a new combination is equal to  $k$ . To better understand this fact we briefly report here a mathematical derivation of Eq. (8.7). We want  $\rho(x, r)$  to be proportional to the distance  $r$  of  $x$  from  $src$ , so let us suppose that  $\rho(x, r) = cr$  where  $c$  is a constant. In addition, for each coverage area we want that only  $k$  nodes forward a new packet combination. As a consequence, the following condition has to be hold:

$$\int_0^R \mu 2\pi r \rho(x, r) dr = k \quad (8.8)$$

where  $\mu$  is the node density (i.e.,  $\mu 2\pi R$  is the number of nodes in the  $src$  coverage area). At this point, with simple steps we can derive Eq. (8.7):

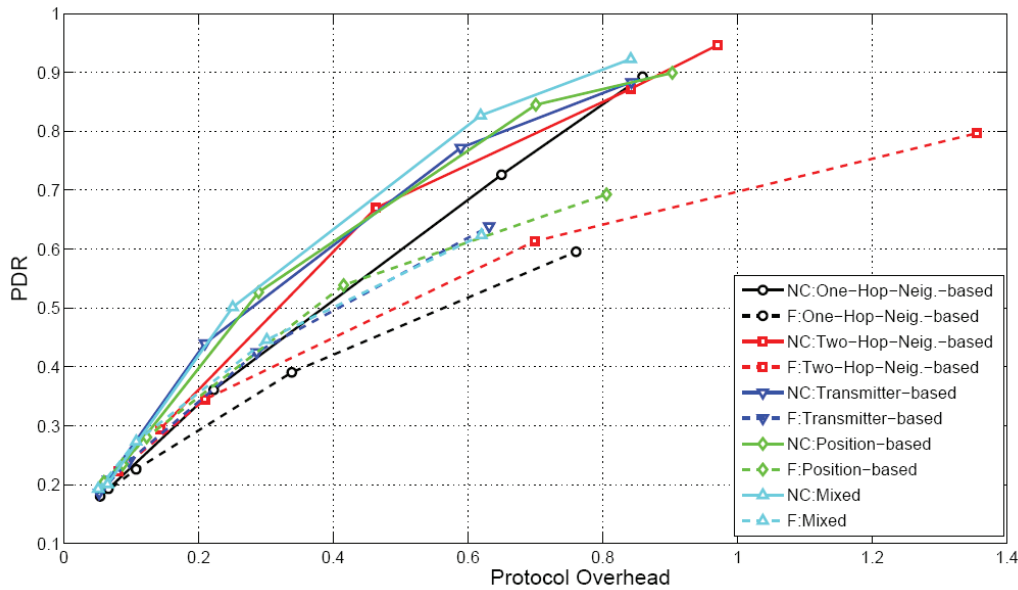
$$\begin{aligned} \int_0^R \mu 2\pi r c r dr &= k \\ \int_0^R c r^2 dr &= \frac{k}{\mu 2\pi} \\ \frac{R^3}{3} &= \frac{k}{c \mu 2\pi} \\ c &= \frac{3k}{\mu 2\pi R} \\ rho(x, r) &= \frac{3kr}{\mu 2\pi R} \end{aligned} \quad (8.9)$$

Note that this equation gives good performance only when the node density is high.

On the basis of the previous description we expect that *Mixed Forwarding Factor* performs better than the others as it takes into account both the node density to limit the redundancy and the node position to favor the dissemination speed.

## A Performance Evaluation

We evaluate, in this Section, all the strategies based on the adaptive forwarding factor approach in a realistic environment where IEEE 802.11b is used by taking into account different node densities. For sake of simplicity, we report here only the results related to the *packet delivery ratio* plotted versus the *protocol overhead* which is defined as the ratio between the total number of transmitted packets



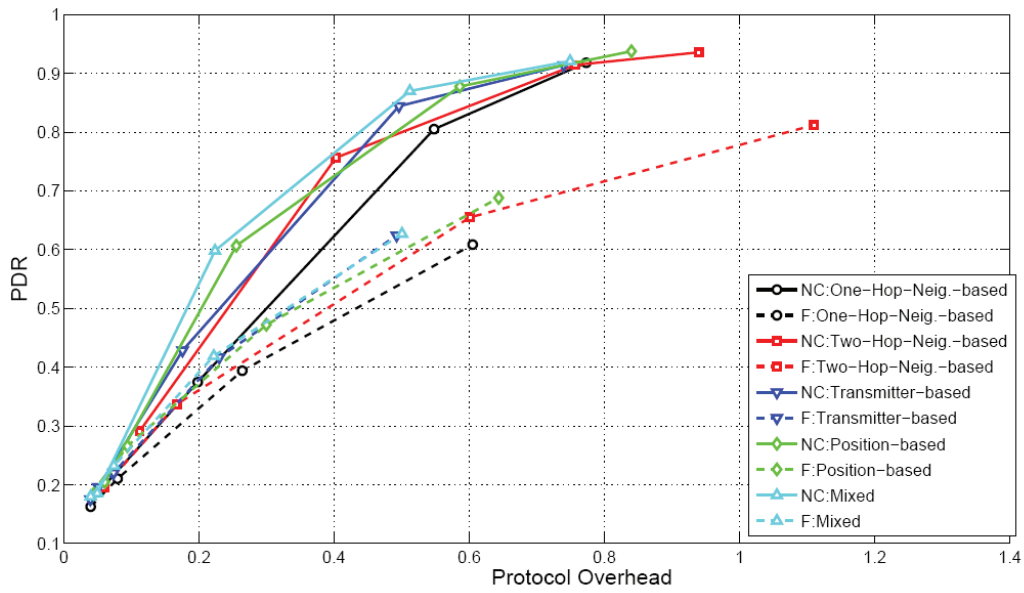
**Figure 8.14** 39 Nodes (7 neighbors on average): Comparison among different strategies based on adaptive forwarding factor using IEEE 802.11b

( $Pkt_{tx}$ ) over the total number of decoded packets ( $Pkt_{dcd}$ ) per node. Figs. 8.14, 8.15, 8.16 and 8.17 refer to different network sizes. We compare network coding performance with probabilistic flooding where the probability to forward a message is evaluated using the same equations as network coding. The first thing we can observe is that network coding strategies always outperform flooding schemes giving a gain of about 1.6. In addition, the benefit of network coding is higher when the node density increases as it can strongly reduce the redundancy. On the contrary, the differences among the adaptive forwarding factor strategies are limited and the performance is very close with each other. *Mixed Forwarding Factor* always performs better as we expected. The advantages are more evident in case of low node density as it guarantees major advancements in data dissemination. In addition, we can note that, due to the presence of a real MAC layer (IEEE 802.11b), no strategy is able to reach the full packet delivery ratio.

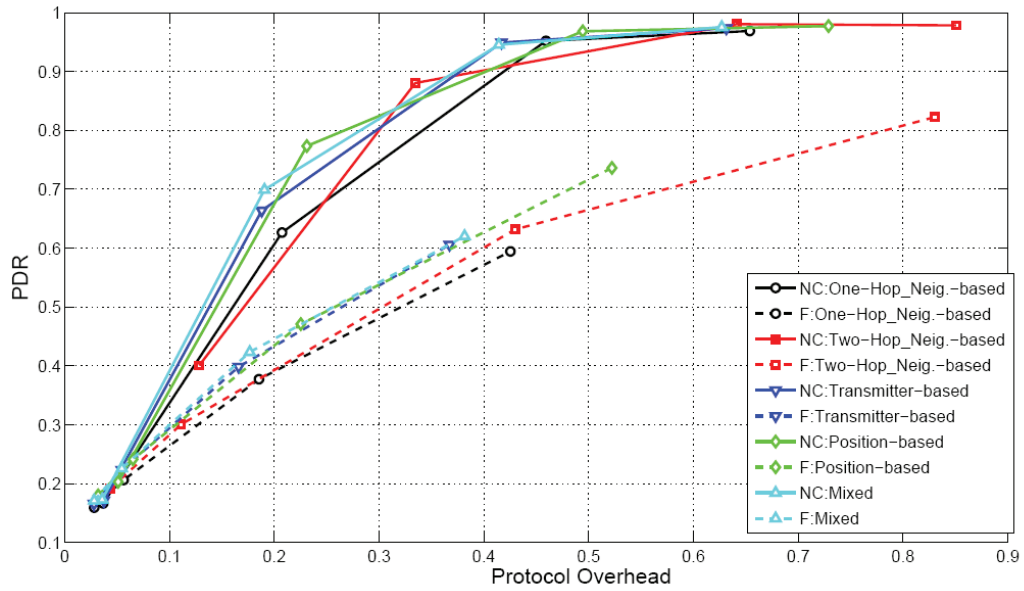
## B Observations

The analysis of different adaptive approaches leads to two main observations.

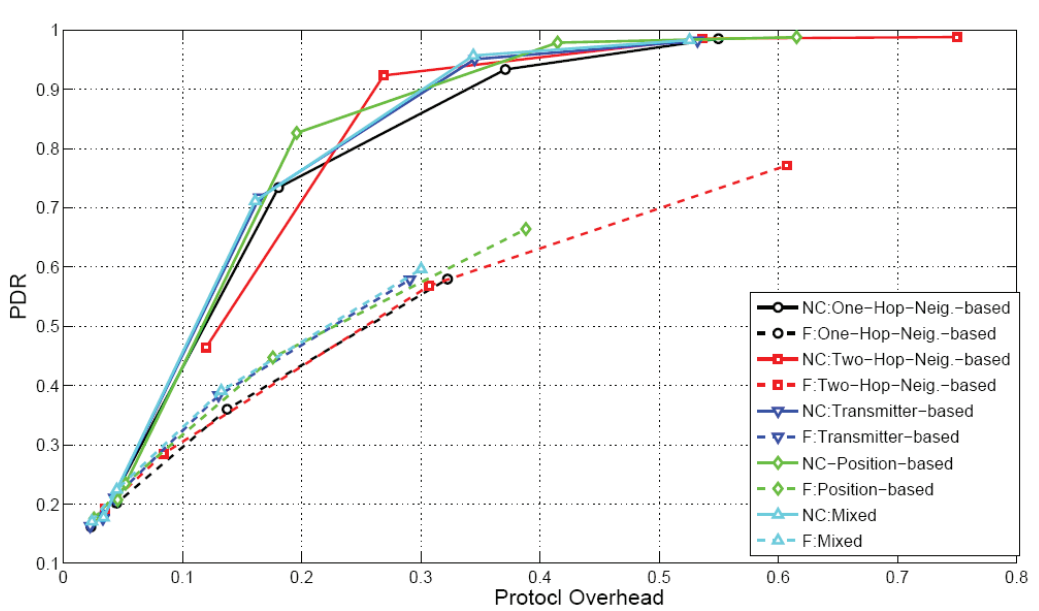
First, the different strategies show similar performance versus the protocol overhead. This means that, for instance, to achieve the same performance in terms of packet delivery ratio the strategy in use has to guarantee a specific amount of protocol overhead. The way how such a protocol overhead is provided has a limited importance. However, some strategies perform slightly better than the others as



**Figure 8.15** 50 Nodes (9 neighbors on average): Comparison among different strategies based on adaptive forwarding factor using IEEE 802.11b



**Figure 8.16** 67 Nodes (12 neighbors on average): Comparison among different strategies based on adaptive forwarding factor using IEEE 802.11b



**Figure 8.17** 83 Nodes (15 neighbors on average): Comparison among different strategies based on adaptive forwarding factor using IEEE 802.11b

they favor the data dissemination in the more critical areas by limiting the interference. It is the case, for instance, of the *Mixed Forwarding Factor*.

Second, we focus on the protocol overhead: we note that, our analysis with a fixed forwarding factor states that, to achieve a satisfactory packet delivery ratio (say  $PDR = 0.97$ ), nodes need  $\rho = 0.4$ . This value of the forwarding factor produces a protocol overhead which is the same used by the adaptive strategy to achieve similar reliability performance. This fact confirms what we previously said, i.e., using a probabilistic network coding approach we can guarantee satisfactory performance only with a relative high protocol overhead. Probabilistic network coding strongly outperforms probabilistic flooding but can not achieve the theoretical performance predicted by the analysis. The main reason is that, in realistic environments, we need to deal with the problem of interference and packet losses. To assure a certain level of packet delivery ratio, we need to introduce in the network some redundancy, i.e., transmit more packets than the strictly needed ones. In addition, probabilistic network coding does not provide a way to regulate the redundancy introduced as all nodes' behavior is only known probabilistically. This produces on average the same behavior for all the strategies.

To further reduce the protocol overhead we need to move toward different directions by exploiting other network coding approaches. However, before doing this, we need to better understand how low the protocol overhead in realistic environments could be. We look at this in the next Section.

## 8.6 Innovation-based Network Coding

The existing practical network coding schemes do not use complete topology information neither knowledge about the network status. Usually they exploit only local information about, for instance, the number of neighbors.

The first observation we can do is that a network coding scheme based on a complete knowledge of the network might perform better than a strategy with a partial system view as it can manage more efficiently the network resources. For this reason, we develop a scheme, *Innovation-based network coding* to use it as a comparison for other schemes performance.

In the following the *Innovation-based network coding* is introduced and the simulation results are reported. The main assumption we use here is the complete knowledge of the network (i.e., both topology and network status).

### 8.6.1 Description of the Scheme

We focus now on the packet combination strategy which we name *Innovation-based network coding*. It consists of a simple schedule of transmissions: nodes gain the opportunity to transmit on the basis of a priority which is determined at each step of the simulation supposing to have a *complete knowledge of the network state*.

There are many methods to define the priority  $P_{TX}(x, t)$ , of a generic node  $x$  at round  $t$ , to gain the opportunity to transmit. Here, we consider the two defined in the following:

$$P'_{TX}(x, t) = \sum_{i=1}^{n(x,t)} INN_{pkt}(i, t), \quad (8.10)$$

where  $n(x, t)$  is the number of neighbors of node  $x$  at round  $t$  and  $INN_{pkt}(i, t)$  is the number of innovative packets the node  $x$  could send to its  $i$ -th neighbor combining all packets currently (i.e., at round  $t$ ) stored in its buffer. Note that if the nodes are static  $n(x, t) = n(x)$ . We name this scheme  $S_1$ . The second priority definition is given by:

$$P''_{TX}(x, t) = n(x, t) - n_{NI}(x, t), \quad (8.11)$$

where where  $n(x, t)$  is the number of neighbors of node  $x$  at round  $t$  and  $n_{NI}(x, t)$  is the number of neighbors of node  $x$  at round  $t$  for which a packet from node  $x$  is non-innovative. We call this scheme  $S_2$ .

In each round  $t$ , the node with the highest priority transmits. Note that, in case of more nodes with the highest priority, only one node (randomly chosen) is elected as the current transmitter. It generates a new combination and this is inserted in its neighbors' buffers. At this point the new network status

is computed, the priorities are updated and a new round starts. The procedure continues until all nodes receive all packets.

### 8.6.2 Simulation Results

We present here the results obtained using the *Innovation-based network coding*, considering the metrics listed below. In the rest of the document we consider the same metrics to evaluate also the other schemes.

- *Packet Delivery Ratio* (PDR): it is the number of decoded packets over the total number of wanted packets per node.
- *Overhead* (OH): it is the number of transmitted packets over the number of decoded packets. In other words, it is the number of transmissions required to decode a packet per node, on average.
- *Protocol Redundancy* (RX): it is represented by the number of received packets over the number of decoded packets per node. Note that, if this number is equal to 1, we have no redundancy because each node, which receives  $K$  packets, decodes exactly  $K$  packets, i.e., it doesn't receive any non-innovative packets. On the contrary, when  $RX$  is greater than 1, it gives us a quantitative measure of non-innovative packets received by each node, on average.
- *Protocol Efficiency* ( $\eta$ ): it is the ratio between the number of decoded packets and the number of transmitted packets.

In each figure, we compare  $S_1$  (blue circled line) and  $S_2$  (red squared line). All the metrics are plotted vs. the number of nodes in the network (i.e., varying the node density) and they are averaged over  $N_{sim} = 10$  different simulations for each scenario.

Fig. 8.18 shows that the *Innovation-based network coding* can guarantee full reliability (exactly 100% for each node density) for any node density using both  $S_1$  and  $S_2$ . Indeed, due to the Eq. 8.10 and Eq. 8.11, the simulations stop when all nodes receive all packets.

The positive aspect is that this result is achieved using a very low overhead as we can see from Fig. 8.19. As an example, for networks with  $N = 67$  nodes, each node has to send only 8 packets, on average, to decode 67 packets. As the node density increases, this value decreases (e.g., when  $N = 83$  each node has to send 9 packets to decode all the 83 packets) due to the higher diversity in the packet combinations which guarantees a lower level of redundancy.

Fig. 8.20 reports the ratio between the number of received packets and the number of decoded packets. We know that to decode  $N$  packets each node has to receive  $N$  innovative packets. We can reread the results in Fig. 8.20 as an indication of the percentage of non-innovative packets received or,

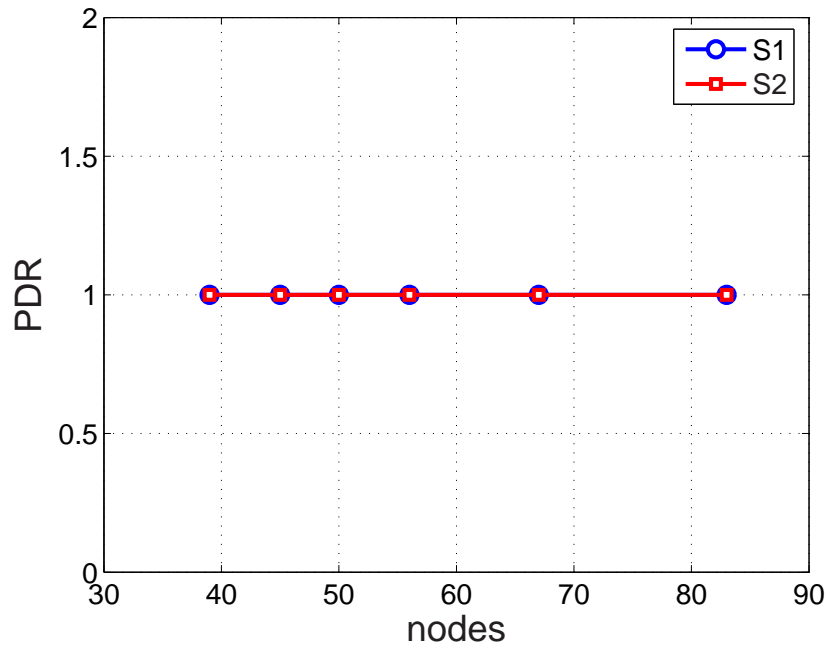


Figure 8.18 Packet Delivery Ratio

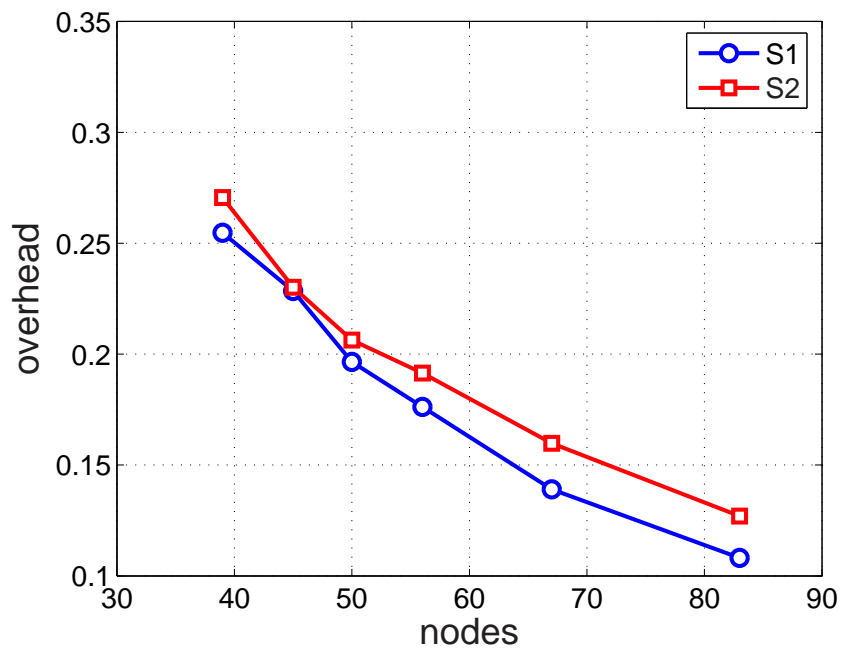


Figure 8.19 Protocol Overhead



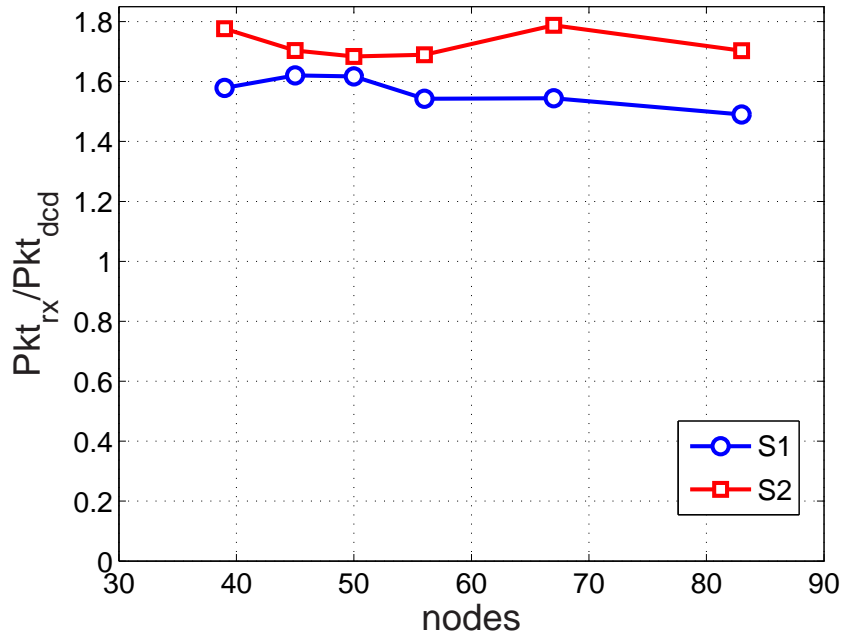


Figure 8.20 Protocol Redundancy

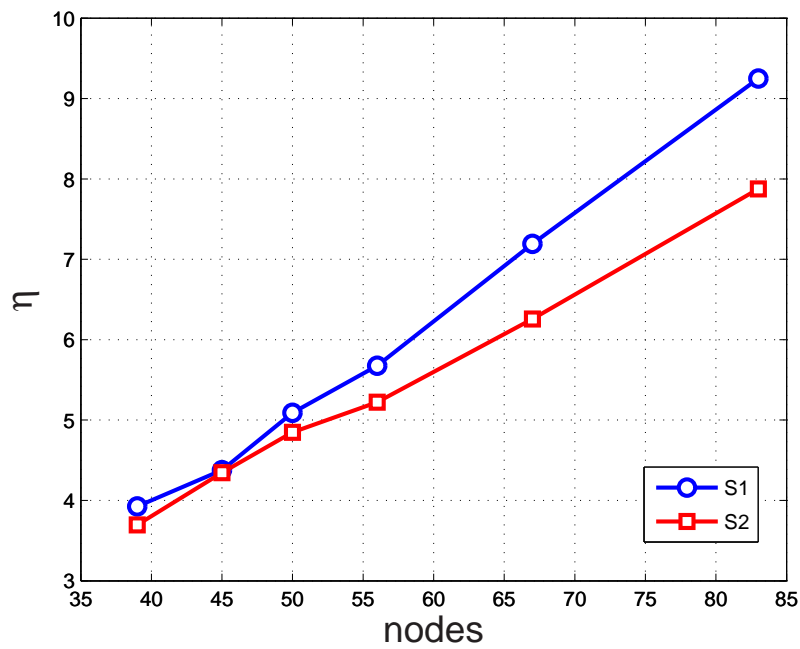


Figure 8.21 Protocol Efficiency

also, the percentage of redundancy introduced by each scheme. The shape of the lines, in this case, is not so smoothed due to the fact that this metric is sensitive to the network topology and we average over a limited number of different settings.

Finally, Fig. 8.21 presents the protocol efficiency which is very high for this scheme. It is another way to see the protocol overhead. Indeed, it could be simply obtained by  $1/\text{overhead}$ .

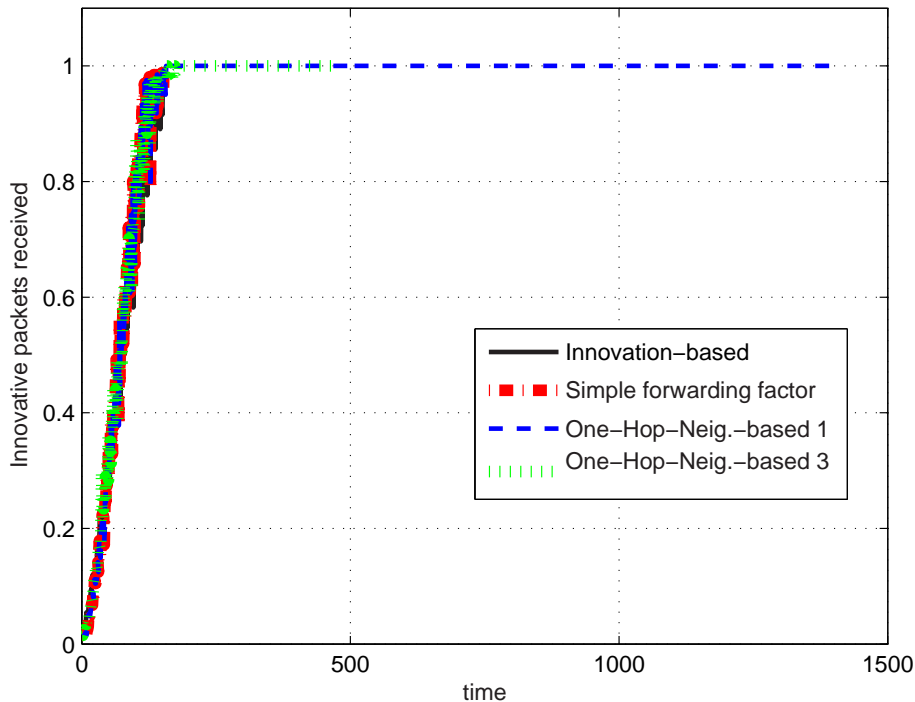
From the presented results we can say that the approach  $S_1$  always outperforms strategy  $S_2$ . In fact, it guarantees always a lower overhead and a lower redundancy thus leading to a higher protocol efficiency, to achieve the same reliability. This fact could be explained looking at the different priority management used by the two scheme. Using  $S_1$  we try to speed up mostly the information dissemination in those areas where the information is missing while, using  $S_2$  we distribute the information uniformly over the network thus incurring in a slower dissemination. As an example, we can consider a node,  $A$ , having two neighbors  $B$  and  $C$ .  $A$  has a packet to be transmitted which, we assume to be innovative for node  $B$  and non-innovative for node  $C$ . In addition, we suppose that the difference between the space dimensions of  $A$ 's matrix and  $B$ 's matrix is two. It means that  $A$  could send to  $B$  two innovative packets. If we compute  $A$ 's priority, we obtain 2 using Eq. 8.10 and 1 with Eq. 8.11. So,  $S_1$  favors the information flow also where the number of neighbors is low but they lack a lot of information.

We can consider these results as an upper bound for the studies on network coding in random networks developed over more complex simulators. When we implement more practical strategies we have to face the problem of collecting information from the neighborhood, organizing the packet scheduling and so on. We can also note that, in general, the performance increases as the node density increases. This is due to the fact that, in networks with higher density, we have more diversity when we combine packets. This produces fewer non-innovative packets.

### 8.6.3 Observations

The results presented in this Chapter has shown that probabilistic reactive network coding is generally a good solution for broadcasting data in. However, we have highlighted that this technique is likely to suffer from the presence of interference and collisions in actual radio environments. The main problem of reactive schemes is that new random combinations of packets are only generated and transmitted when innovative information is received. Innovative packets may however be lost in scenarios with packet collisions, thus interrupting data propagation. Furthermore, insertion of innovative information into an area often causes all nodes in the area to attempt their new transmissions simultaneously and this further increases the collision probability.

In addition, in reactive probabilistic network coding, nodes send out new combinations based on a



**Figure 8.22** Reception of innovative packets versus the time.

forwarding factor  $\rho$  which depends on the number of neighbors they have [40]. We observe that there are particular topologies where this strategy does not work. As an example, let's think of the case where a given node  $x$  has a large number of neighbors and one of them, say node  $y$ , has only  $x$  as its neighbor. Due to its high number of neighbors (small  $\rho$ ),  $x$  sends out a small number of packets and, in turn,  $y$  is unlikely to be able to decode all the wanted information (as it did not receive enough independent combinations from  $x$ ). We may increase  $\rho$  until the number of transmissions per node allows the recovery of a sufficient number of packets also in the above case. Unfortunately this solution has two drawbacks: increasing  $\rho$  leads to more severe channel congestion and increased overhead (number of packets sent per original packet). This is clearly not desirable as it largely neutralizes the performance gain due to network coding.

In Fig. 8.22 we report a graphical example of different strategies behavior over the time. The Figure represents the number of innovative packets stored in a node buffer on average. This quantity is plotted against time to show how the innovative information in the buffer increases. We compare four different scheme: *Innovation-based network coding*, *Simple forwarding factor* scheme, *One-Hop-Neighbor-based forwarding factor* scheme with  $k = 1, 3$ . *Simple forwarding factor* scheme is introduce here for the first time. It represent a simple approximation of the *Innovation-based network coding* where

nodes simply follow the *One-Hop-Neighbor-based forwarding factor* scheme but stop their transmission when all their neighbors have received all the required packets. It is implemented assuming that nodes have a complete knowledge of the neighborhood. It is not feasible in practice without any local message exchange but this strategy represent a first step towards the definition of our proposal that we will describe in the next Chapter.

First of all we note that, when we use the *One-Hop-Neighbor-based forwarding factor*, a simulation run is longer than in the two other cases. This is because all the transmissions stop when no further innovative packets are received. On the contrary, the condition to stop the transmission when *Innovation-based network coding* is used, is different and it is based on the amount of innovative information stored in the buffers. Second, we observe that, in the last period of the data dissemination through *One-Hop-Neighbor-based forwarding factor* approach, the number of innovative packets does not increase. This means that most of the last transmissions are non-innovative, i.e. unnecessary. Indeed, according to the reactive nature of these schemes, when a node receives an innovative packet, it decide with a probability higher than zero to send out a new combination even though all its neighbors have already decoded all the packets. These situations, on the contrary, are avoided by *Innovation-based network coding* and *Simple forwarding factor* schemes.

## 8.7 Conclusions

In this Chapter we analyzed the impact of realistic MAC layer, transmission schedule, and packet combination strategy on random network coding. We also introduced a simple timing policy which improves network coding performance. By evaluating the packet delivery ratio of network coding strategies in the presence of different CSMA protocols, we observed that the impact of the MAC layer is not as large as expected. That is, introducing mechanisms to alleviate the collision problem only leads to limited improvements in the packet delivery ratio performance and affects the latency. On the other hand, we observed that the packet combination strategy plays a fundamental role. In fact, our proposal, in spite of its simplicity, shows promising results. We note also that all the schemes and protocols studied in this Chapter do not achieve the theoretical performance of [40], and this motivates further research, especially concerning the combination strategy to use. As an example, by looking at Fig. 8.12 and focusing on the timed network coding case for  $\rho = 0.5$  and two neighbors, we note that the packet delivery ratio is  $PDR \approx 0.6$ . This means that there is still room for improvement in order to get closer to the performance of  $PDR = 1$ . In addition, in Section 8.6.3 we showed why the existing network coding schemes lack efficiency by the use of a lot of non-innovative transmissions.

On the basis of such results, we propose, in the next Chapter, a novel data dissemination scheme based on network coding which outperforms the existing protocols in practical environments.

# 9

## Proactive Network Coding

---

*Everything should be made as simple as possible,  
but not one bit simpler.*

(Albert Einstein)

All the network coding schemes introduced previously are based on a *reactive* approach. We also noted that they are not efficient in practical scenarios as they introduce a lot of redundant data transmission. The same performance could be achieved by a protocol which uses less transmissions. The design of such a scheme is the aim of this Chapter. Indeed, in the following we propose a network coding data dissemination scheme based on a *proactive* approach (referred to in the following as *Proactive Network Coding (ProNC)*) which achieves good performance also in actual CSMA/CA environments. In particular, we focus on scenarios where data has to be exchanged among all the users of a wireless ad hoc network. Our scheme is completely distributed and self-adaptable and requires very limited network knowledge, which can be easily acquired by overhearing the exchanged data. We show the superiority of our approach by comparing it against existing network coding strategies [40] and against an idealized scheme with a perfect priority scheduler. In the latter case, access priorities are calculated using full knowledge of the buffer contents of all nodes.

### 9.1 Proactive Network Coding (ProNC)

A challenging problem in wireless ad hoc networks consists of efficiently disseminating information network-wide. In this Chapter, we address this problem by devising broadcast schemes based on network coding. We consider a scenario where at every node an application inserts packets into the network, and all nodes want to collect all the inserted packets. We refer to the inserted packets as *original packets*. Probabilistic reactive network coding is generally a good solution for broadcasting data in these settings [40]; however, previous Chapter has highlighted that this technique is likely to suffer from the presence of interference and collisions in actual radio environments. The main problem of reactive schemes is that new random combinations of packets are only generated and transmitted when innovative (i.e., linearly independent) information is received. Innovative packets may however be lost in scenarios with packet collisions, thus interrupting data propagation. Furthermore, insertion of

innovative information into an area often causes all nodes in the area to attempt their new transmissions simultaneously and this further increases the collision probability.

In this Chapter we look at an alternative strategy. Instead of considering the reactive paradigm introduced above, we adopt a *proactive* approach where each node periodically sends out new random packet combinations. The main advantages of a proactive scheme are: 1) it does not require the reception of innovative information to continue data dissemination, so it is more robust to interference and collisions; and 2) its performance does not depend on the specific choice of the forwarding factor  $\rho$ . A proactive data dissemination scheme needs two important components to work:

1. A set of conditions to stop transmissions when all original packets have been delivered to all nodes, i.e., *Stopping Conditions (SC)*.
2. A strategy to set the frequency at which the new random packet combinations are sent out so as to avoid network congestion and to save energy consumption. In the rest of the Chapter we refer to this strategy as *Rate Adaptation* mechanism.

In Section 9.1.1, we first describe the basic rules of our proactive network coding (*ProNC*) data dissemination scheme. In Section 9.1.2, we define the *Stopping Conditions* and, in Section 9.1.3, we discuss the problem of finding a proper *Rate Adaptation* heuristic. Finally, in Section 9.1.4, we highlight some aspects related to the implementation of *ProNC*.

### 9.1.1 Basic Rules for ProNC

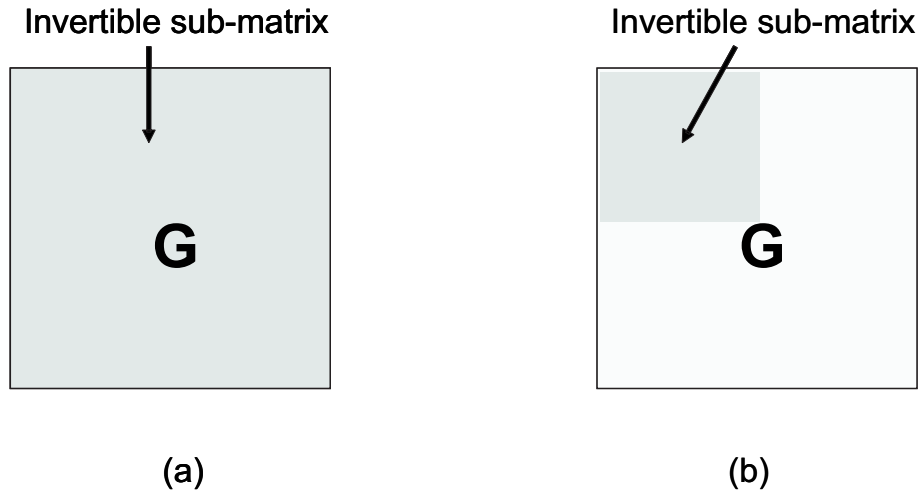
We assume that each node can be in one of two different states: *active* and *inactive*. The basic idea of the proactive approach is that an *active* node periodically sends out a new packet combination to its neighbors, while an *inactive* node does not. To switch from one state to the other, a node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

**Rule 1:** *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.

**Rule 2:** *A node becomes inactive when the Stopping Condition 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.

**Rule 3:** *A node becomes active again when the Stopping Condition 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 shall be used to assess whether the stopping condition still holds.



**Figure 9.1** a) *Strong* and b) *Weak* Stopping Conditions

### 9.1.2 Stopping Conditions

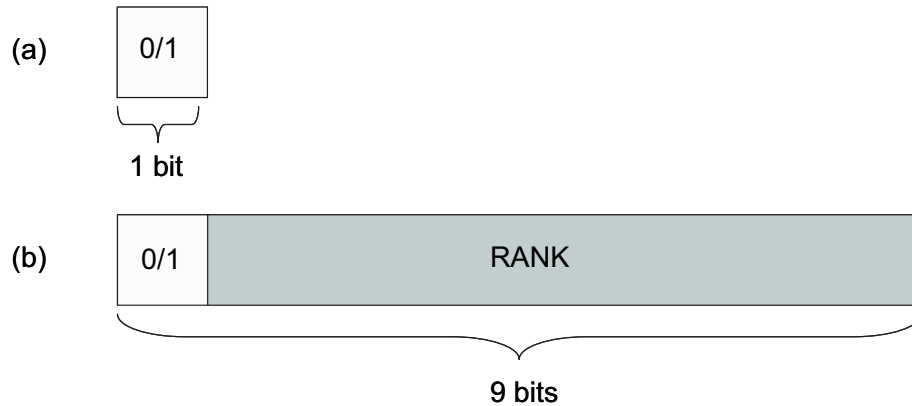
There are different ways to define the *Stopping Conditions* for proactive network coding. They depend, in general, on the amount of information that each node has to collect in order to decide whether to suspend its transmissions. Our main aim is to keep the overhead as low as possible. The motivations for this are twofold: our scheme should be distributed and self-adaptable so that nodes should not need full knowledge about the network topology/status; moreover, maintaining the overhead low contributes to avoiding network congestion for a given traffic condition.

We identify two simple cases in which a node has to suspend its transmission.

1. In the first case, all neighbors of a node  $x$  have decoded all the packets they require and thus no further transmissions by  $x$  are necessary (see Fig 9.1a).
2. The second is when the subspace spanned by the information vectors (i.e., packets) available at node  $x$  is contained in the subspace spanned by the information vectors at each of the node's neighbors. In this case,  $x$ 's packets will not be innovative for any of its neighbors and the node should suspend its transmission (see Fig 9.1b).

Based on these observations, we propose two different conditions which are referred to as *Strong* and *Weak* Stopping Conditions (*SSC* and *WSC*, respectively). They implicitly define two different proactive schemes.

According to the *SSC*, nodes send out beacons (*Strong Stopping Messages, SSM*) to their neighbors when they have decoded all the packets they are interested in (refer to Fig. 9.2a). Each node collects *SSMs* from its neighborhood in order to autonomously verify the *SSC* and thus suspend its own transmission.



**Figure 9.2** Beacon structure for the a) *Strong* and b) *Weak* Stopping Conditions

*Strong Stopping Conditions:* when a node receive a *SSM* from each of its known neighbors the *SSC* is verified and its transmissions are stopped.

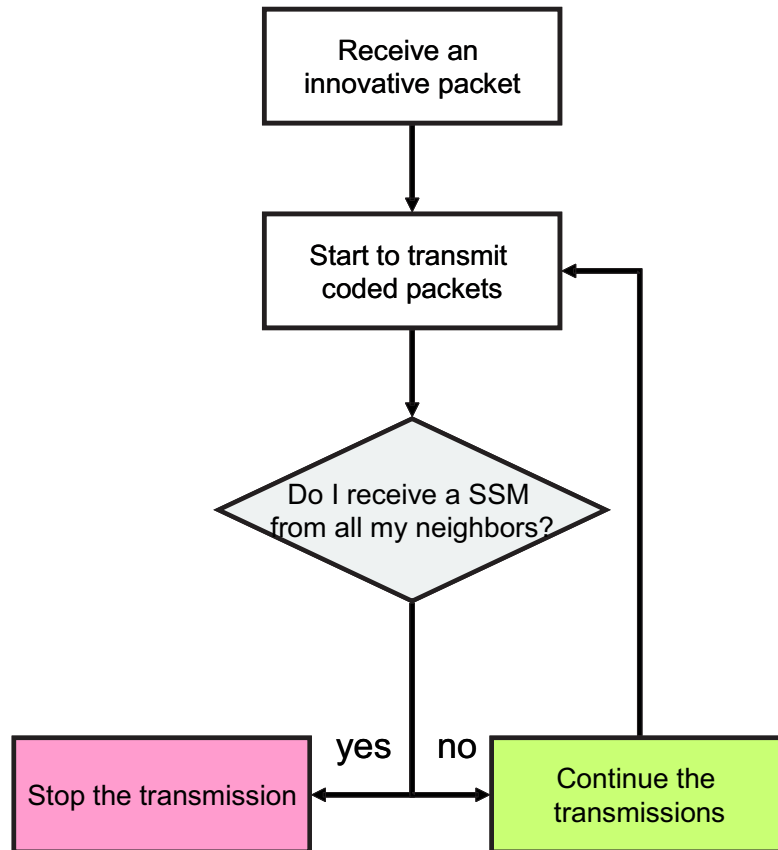
We refer to this scheme as *Strong ProNC* as it requires strong assumptions on the data traffic and we report in Fig. 9.3 the main steps of this protocol. Note that each node, in order to send out *SSMs*, needs to know in advance how many packets it wants to collect. This fact implies that each node has full knowledge about the amount (and type) of data flowing over the network. Note that the collection of this information, in practice, may be infeasible. As a practical example, imagine that a node is interested in collecting sensor readings from all sensors placed in a specific area. In order to send a *SSM*, the node must know in advance the number of transmitting sensors and hence the number of packets to collect.

The second strategy, instead, is based on the *WSC*. During data propagation, each node sends out beacons (*Weak Stopping Messages, WSM*) 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 (refer to Fig. 9.2b).

*Weak Stopping Conditions:* Each node suspends its transmissions when all its neighbors can decode all the packets in their buffers, i.e., all the decoding fields are equal to 1 and their decoding matrices all have the same rank, i.e., all the rank fields are equal

Note that, at a certain time a node can satisfy the *WSC* and thus it suspends its transmissions. Then, something in its neighborhood could change as a consequence of the data flowing. If the *WSC* are not still verified the node starts again to transmit.

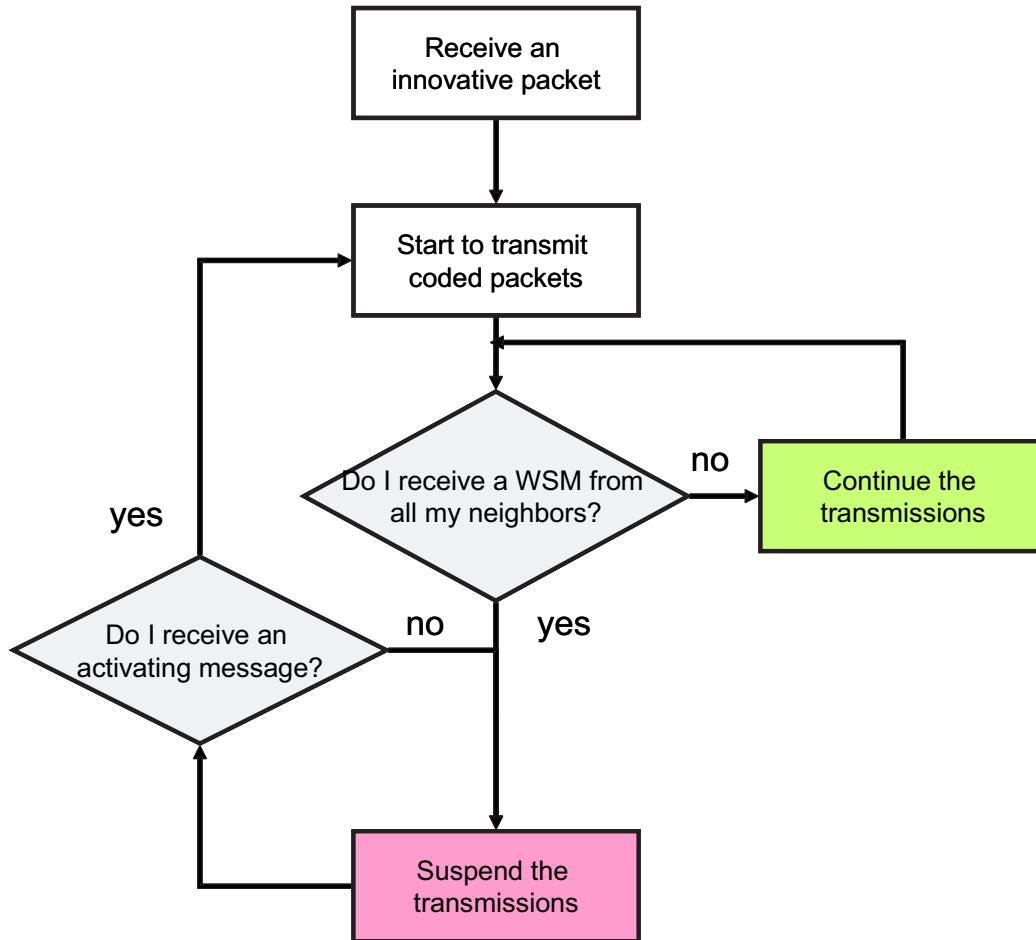




**Figure 9.3** Flow chart of the *Strong ProNC* protocol

We refer to this second strategy as *Weak ProNC* because it does not require any knowledge about the data traffic and has a limited overhead. Its main steps are summarized in Fig. 9.4. 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.

We compare the performance of both *Strong* and *Weak ProNC* in Section 9.2. In addition, we also consider, as a benchmark, a scheme based on the complete knowledge of all the buffers' contents and where an omniscient entity regulates the packet transmissions. Note that implementing such a scheme in a distributed network requires to have each node send out state vectors describing the content of its own buffer at the reception of an innovative packet (in a way similar to [1]). In case of all-to-all communications this leads to an unsustainable overhead in terms of transmissions and memory.



**Figure 9.4** Flow chart of the *Weak ProNC* protocol

### 9.1.3 Rate Adaptation Heuristic

When a node is active, it periodically sends out new random packet combinations generated from its own buffer. A crucial aspect of *ProNC* is therefore the adopted rate adaptation strategy, as previously defined in Section 9.1. A proper selection of the transmission rate used at each node is important to avoid congestion while achieving good decoding performance. This selection translates into choosing a proper time interval between two consecutive transmissions, (referred to as  $\tau$ ).

Our first aim is to avoid both synchronization among nodes and a high simultaneous usage of the channel. For this reason, we take  $\tau$  as a uniform random variable in  $[\tau_{avg} - \tau_{avg}/2, \tau_{avg} + \tau_{avg}/2]$ , where  $\tau_{avg}$  is the average value of  $\tau$ . We also define the quantity  $\mu_{avg} = 1/\tau_{avg}$  which represents the average packet transmission rate. In addition, we allow  $\tau$  to vary across consecutive transmissions and across nodes. This avoids synchronization and limits channel contention. The second problem to be considered is the selection of a good value for  $\mu_{avg}$ ; when all nodes use a high transmission rate,

channel collisions are the bottleneck and, in turn, we would expect unsatisfactory delay performance. However, at very low transmission rates the delivery delay will also be very long and information will slowly propagate through the network. For intermediate transmission rates, an optimal value of  $\mu_{avg}$  should exist. This optimal value should minimize the delay while giving acceptable protocol overhead performance (packet transmissions per recovered packet). We finally note that the optimal  $\mu_{avg}$  should depend on the node density (number of neighbors per node). In particular, for increasing densities  $\mu_{avg}$  should decrease so as to keep the number of collisions at an acceptable level.

In Section 9.2.2 we analyze, by means of simulations, that an optimal transmission rate in fact exists and we validate the statement above. Based on these results, in Section 9.2.3 we detail a density dependent rate adaptation heuristic and we compare its performance against reactive network coding.

#### 9.1.4 Implementation Notes

*ProNC* requires the estimation of the number of neighbors at each node. This can be simply achieved by monitoring the source addresses of incoming packets. Note that both the stopping conditions and the rate adaptation mechanism depend on the node density. In addition, *Stopping Messages* 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<sup>1</sup>. In both cases, the additional overhead is acceptable. On the downside, when a node becomes inactive it must send out at least one *Stopping Message* to communicate its change of status and this packet may be useless for coding purposes.

We stress 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 in Section 9.2.

## 9.2 Simulation Results

In this Section, we evaluate the performance of *ProNC* by means of ns2 simulations. First of all we briefly introduce network topology and traffic pattern and define the considered performance metrics. Then, we analyze the behavior of *Strong* and *Weak ProNC* for different transmission rates. Subsequently, we introduce and evaluate a rate adaptation heuristic. Finally, we compare *ProNC* against the reactive probabilistic network coding schemes proposed in [40, 52].

<sup>1</sup>A single byte often suffices in practice, i.e., when the number of packets to be coded together is lower than or equal to 256. Coding over more original packets would imply the inversion, at the receivers, of large matrices which is impractical and difficult to obtain as a realtime operation.

### 9.2.1 Reference Scenario and Performance Metrics

We consider random topologies as they better capture the main characteristics of actual network settings, especially in wireless ad hoc scenarios. Nodes are randomly placed within a fixed area in such a way that the topology is always connected but the paths among sources and destinations can be multi-hop. We consider several average node densities by varying the average number of neighbors,  $n_v \in \{7, 8, 9, 10, 12, 15\}$ . For medium access control, we adopt the basic IEEE802.11b broadcast mode, accounting for channel errors and collisions. In addition, we assume that at the beginning of the simulation each node has a single original packet to disseminate to all other nodes.

Next, we define the performance metrics that will be used, in Section 9.2.2, to study the behavior of *ProNC*:

**Packet Delivery Ratio, PDR:** is defined as the ratio between the number of successfully received (and decoded) packets, and the total number of packets a node is interested in. This metric is averaged over all nodes.

**Packet Delivery Delay, D:** is the time between the insertion of an original packet (i.e., the beginning of the simulation) and its successful decoding at a receiver, averaged over all nodes that receive it and over all the original packets.

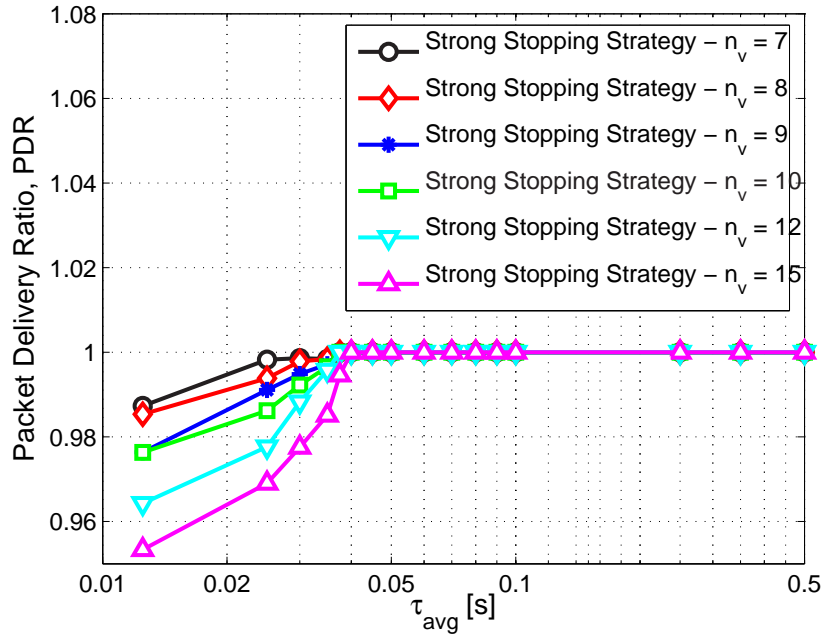
**Protocol Overhead:** is defined as the ratio between the total number of packets transmitted at the MAC layer (including also control packets) and the total number of packets successfully decoded, summed over all nodes<sup>2</sup>.

### 9.2.2 Evaluation of Strong and Weak Stopping Policies

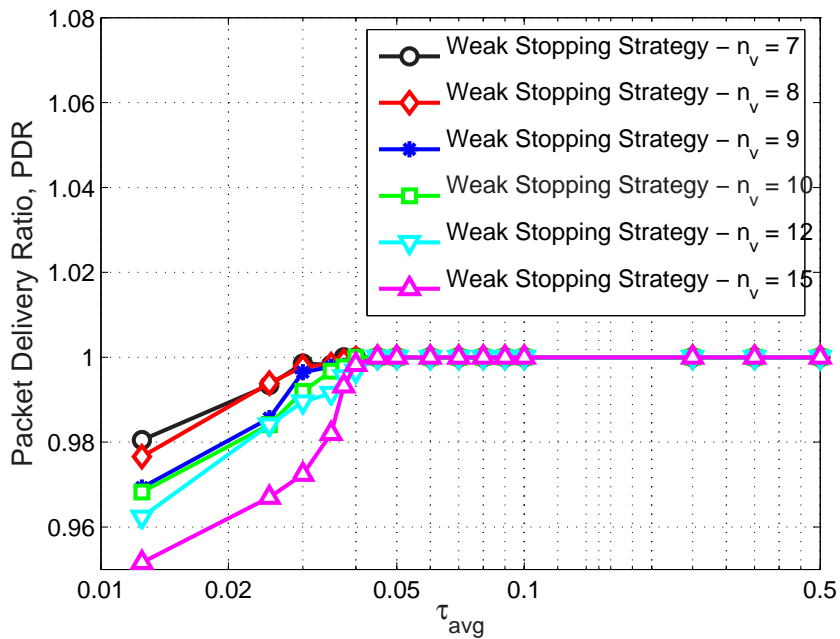
In this Section we study the behavior of *ProNC* under different network conditions when varying  $\tau_{avg} = 1/\mu_{avg}$ . The following results are plotted as a function of  $\tau_{avg}$  to emphasize the impact of inter-packet transmission times on network coding performance. In this Section we assume  $\tau_{avg}$  as a fixed parameter, equal for all nodes and independent of the node density.

In Figs. 9.5–9.10, we show the performance of *Strong ProNC* and *Weak ProNC*, respectively. The behavior of these strategies is very similar, especially for high values of  $\tau_{avg}$  (i.e, low transmission rates). In Fig. 9.5 and Fig. 9.6 we plot the packet delivery ratio. This metric is always equal to one except for the cases where  $\tau_{avg}$  is very small (high  $\mu_{avg}$ ). Under these operating conditions the protocol overhead performance is considerably impacted as well, as can be observed from Figs. 9.7 and 9.8. These behaviors are due to the high collision rates that we get when  $\tau_{avg}$  is excessively small. We further observe that, when the network is congested (small  $\tau_{avg}$ ), *Weak ProNC* performs slightly better than *Strong ProNC*. In fact, *Weak Stopping Conditions* allow nodes to momentarily stop their

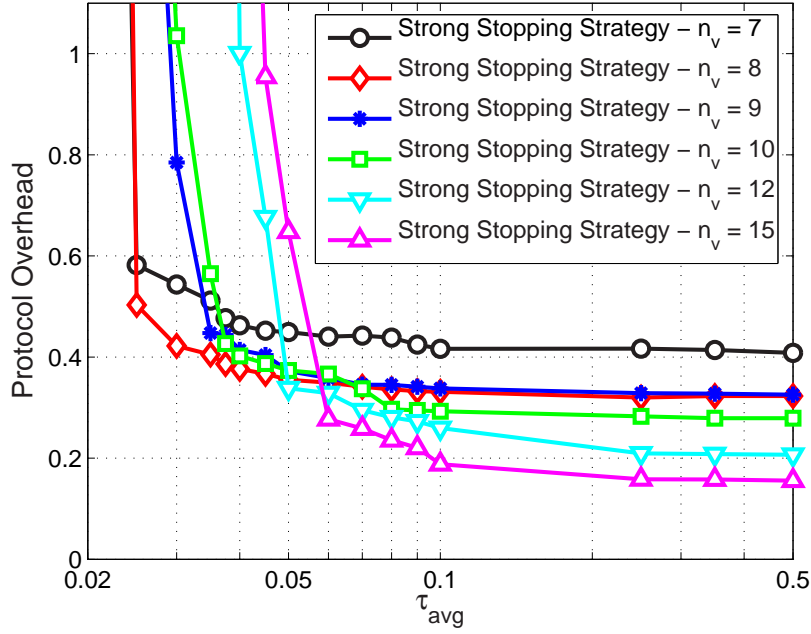
<sup>2</sup>Note that, due to the broadcast nature of the channel and the use of network coding, this ratio could be less than 1.



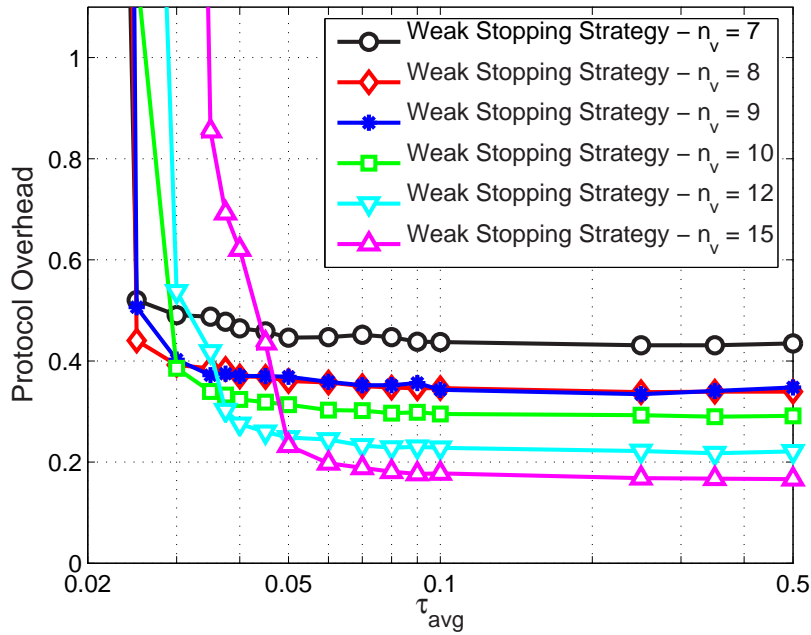
**Figure 9.5** Packet Delivery Ratio: Performance of *Strong ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .



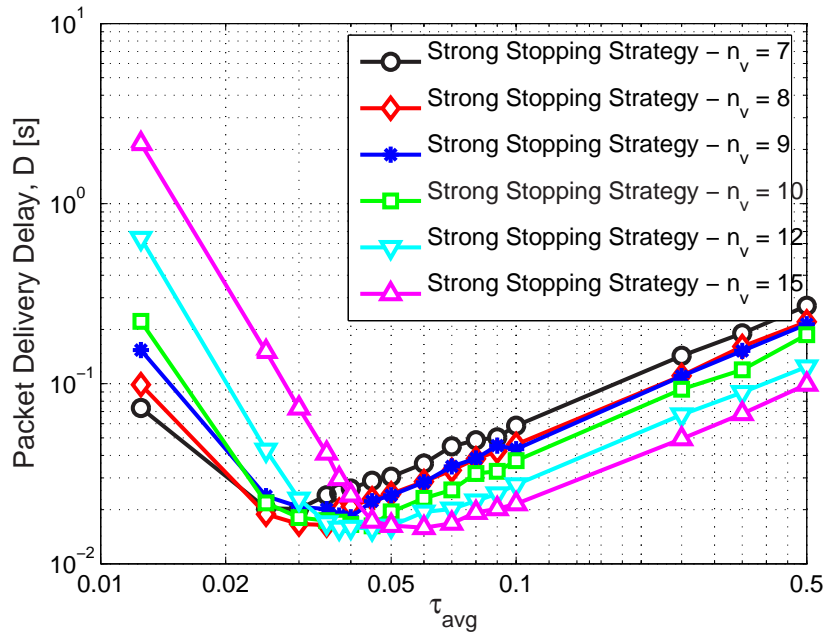
**Figure 9.6** Packet Delivery Ratio: Performance of *Weak ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .



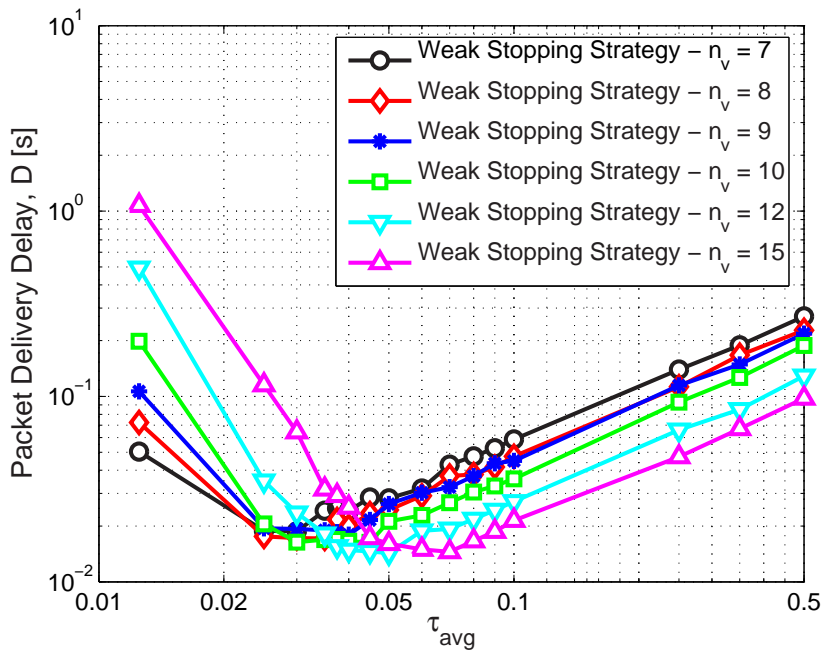
**Figure 9.7** Protocol Overhead: Performance of *Strong ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .



**Figure 9.8** Protocol Overhead: Performance of *Weak ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .



**Figure 9.9** Packet Delivery Delay: Performance of *Strong ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .



**Figure 9.10** Packet Delivery Delay: Performance of *Weak ProNC* as a function of the average insertion interval,  $\tau_{avg}$ .

transmissions when their packets are unlikely to be innovative for their respective neighbors. However, as soon as new information arrives, the nodes will go back to the active state. Note that this requires a continuous assessment of the *Stopping Condition*. Under a *Strong ProNC* this may not happen as the nodes remain in the active state until they receive a *SSM* from *all* their neighbors. However, packet collisions may prevent the reception of *SSMs* from all neighbors and in this case a node will unnecessarily continue transmitting.

In Figs. 9.9 and 9.10 we report the packet delivery delay. As expected, we observe that the delay curves have a minimum for specific values of  $\tau_{avg}$ . As discussed earlier, for small  $\tau_{avg}$  the delay increases due to the severe channel congestion. When  $\tau_{avg}$  is large, instead, the delay increases due to the long lapse of time between consecutive transmissions. We additionally observe that the values of  $\tau_{avg}$  minimizing the delay are slightly different for different node densities ( $n_v$  in the figures) and they do not always coincide with the values minimizing the protocol overhead. We finally note that the differences between *Weak* and *Strong* strategies in terms of minimum delays and optimal  $\tau_{avg}$  are very small.

### 9.2.3 A Possible Rate Adaptation Heuristic

As we mentioned in Section 9.1.3, due to the proactive nature of our scheme, we need to define a proper rate adaptation heuristic to guarantee good performance while making the data dissemination scheme dynamic and self-adaptable. The results in Section 9.2.2 show that an optimal value of  $\tau_{avg}$  exists and that it depends on the node density. Fig. 9.11 reports the optimal  $\tau_{avg}$  for different nodes densities. As done in [40], we assume a linear relationship between number of neighbors and transmission rate at any given node<sup>3</sup>. Accordingly,  $\tau_{avg}$  can be expressed as:

$$\tau_{avg} = \alpha(n_v + 1), \quad (9.1)$$

where  $\alpha$  is a constant,  $n_v$  is the average number of neighbors per node and  $n_v + 1$  the average number of nodes contending for the channel in a given neighborhood. Now, considering the optimal  $\tau_{avg}$ , which can be found by simulations as illustrated in the previous Section, we derive  $\alpha$  as:

$$\alpha = \frac{\tau_{avg}}{n_v + 1}. \quad (9.2)$$

In Fig. 9.11 we report the obtained values of  $\alpha$ . Notably, these values are very close for different node densities. Thus, to define our heuristic, we considered  $\alpha_{avg} = 0.004$ , obtained by averaging  $\alpha$  over all densities. Hence, during the dissemination phase, each node  $x$  calculates its  $\tau(x)$  as:  $\tau(x) = \frac{\alpha_{avg}}{n_v(x)+1}$ , where  $n_v(x)$  is the actual number of neighbors of node  $x$ . Note that the choice of  $\tau$  is

<sup>3</sup>In [40] the forwarding factor  $\rho$  implicitly defines the transmission rate at each node by modulating the transmission process.



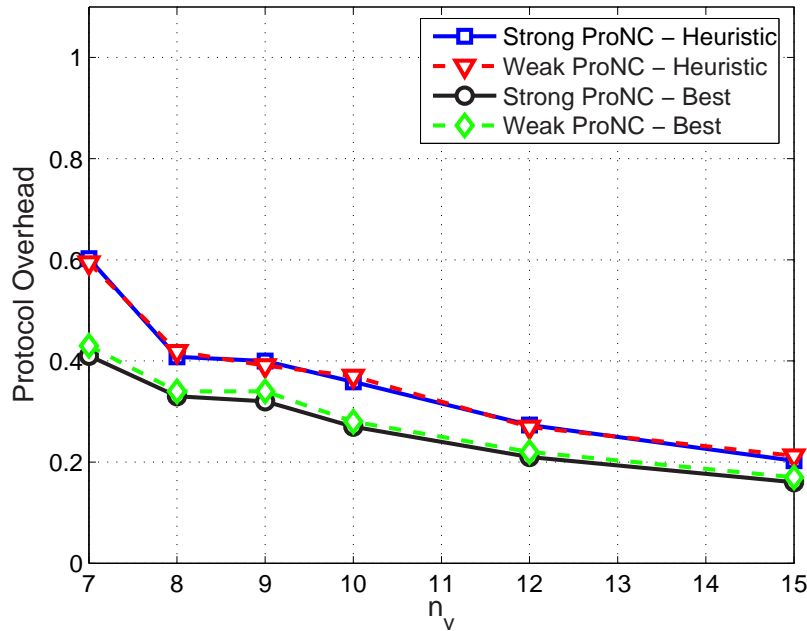
$n_v$	<i>Strong ProNC</i>		<i>Weak ProNC</i>	
	$\tau$ [s]	$\alpha$	$\tau$ [s]	$\alpha$
7	0.030	0.0037	0.030	0.0037
8	0.035	0.0039	0.035	0.0039
9	0.040	0.0040	0.040	0.0040
10	0.045	0.0041	0.045	0.0041
12	0.050	0.0038	0.060	0.0046
15	0.060	0.0037	0.070	0.0044

**Figure 9.11** Optimal  $\tau_{avg}$  and  $\alpha$ .

approximated but it can be derived in a distributed way (each node only requires a local estimation of the number of its own neighbors) and is allowed to be different for different nodes.

We evaluate the performance of the above *Rate Adaptation* heuristic in Figs. 9.12 and 9.13. We omit the packet delivery ratio as it is always equal to one. For comparison, in these figures we also plot two additional curves, for *Weak* and *Strong Stopping* policies, which are obtained by calculating the best performance in Figs. 9.5–9.10. From Fig. 9.12 we observe that the proposed heuristic leads to a small degradation with respect to the overall minimum overhead obtained by choosing the optimal value of  $\alpha$  for each node. From Fig. 9.13, we note that the gap with respect to the minimum achievable delay is larger. We finally observe that the degradation incurred in adopting our heuristic in place of a fixed  $\tau_{avg}$  is the price to pay to have a fully distributed and self-tunable scheme. In fact, if we used a fixed  $\tau_{avg}$ , we would have to know the average node density in advance to achieve optimal performance. However, this knowledge may be hard to obtain in practice. With our heuristic, the performance is slightly decreased but we gain in generality as the resulting scheme works for any topology and without any knowledge about the node density.

To summarize, we can state that both *Strong* and *Weak ProNC* show satisfactory performance in actual network settings. In particular, *Weak ProNC* with our rate adaptation heuristic is a completely distributed and self-adaptable algorithm. Moreover, it does not require any knowledge about the traffic and only requires a few local interactions among nodes to work properly.



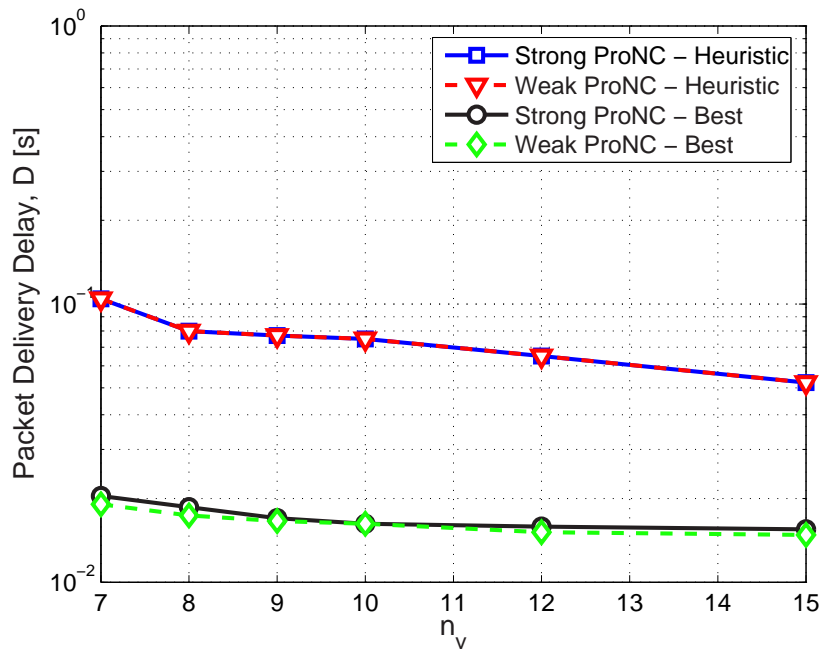
**Figure 9.12** Protocol Overhead: Comparison of the performance of the rate adaptation heuristic with the best achievable for *Weak* and *Strong ProNC*.

## 9.2.4 ProNC vs Reactive Network Coding Schemes

In this Section, we compare the *ProNC* scheme (with our rate adaptation heuristic) against the reactive probabilistic schemes proposed in [40] and the scheme based on complete knowledge about the network status, referred to as *innovation-based network coding*, (*INC*) and introduce in Chapter 8

In Fig. 9.14, we compare *Strong* and *Weak ProNC* against *INC* and reactive network coding with fixed and adaptive forwarding factor (see [40]). We only report the protocol overhead as we compare the schemes for the same value of the packet delivery ratio, i.e.,  $PDR = 1$ . We observe that *ProNC* performs closely to *INC*. Hence, having full knowledge of the network only gives marginal improvement in the considered cases. However, achieving this knowledge in practice may require the transmission of a substantial amount of control traffic, which may drastically reduce the benefits of *INC*. In addition, *ProNC* obtains substantial gains over reactive schemes in terms of protocol overhead (the overhead is roughly halved with *ProNC*). Note that, in this scenario, other schemes such as the probabilistic flooding can achieve a maximum  $PDR = 0.9$  due to the collisions leading to a protocol overhead around 2 [52].

Finally, in Fig. 9.15 we report the delivery delay performance. At low densities all schemes give similar results, whereas as the number of neighbors increases *ProNC* guarantees a packet delivery



**Figure 9.13** Packet Delivery Delay: Comparison of the performance of the rate adaptation heuristic with the best achievable for *Weak* and *Strong ProNC*.

delay that is almost one order of magnitude smaller than that of reactive solutions. In addition, it can be observed that with *ProNC* channel congestion is successfully mitigated for a wide range of node densities. This is possible thanks to the adaptation carried out by our heuristic. Reactive schemes, instead, heavily suffer from an increasing density, which ultimately leads to long delays.

### 9.3 Discussions and Conclusions

In this Chapter, we presented an original network coding scheme for data dissemination. In contrast to prior work, it exploits a proactive approach (*ProNC*), which solves some of the problems of network coding in realistic wireless environments. Our algorithm is distributed and self-adaptable. Also, the scheme requires some local coordination among nodes, which can be achieved through piggybacking control messages at a reasonable overhead. We evaluated the effectiveness of *ProNC* in distributed wireless settings, getting very good performance for all considered cases. Some issues are still open such as the evaluation/adaptation of our scheme in multicast/unicast scenarios with non-homogeneous traffic.

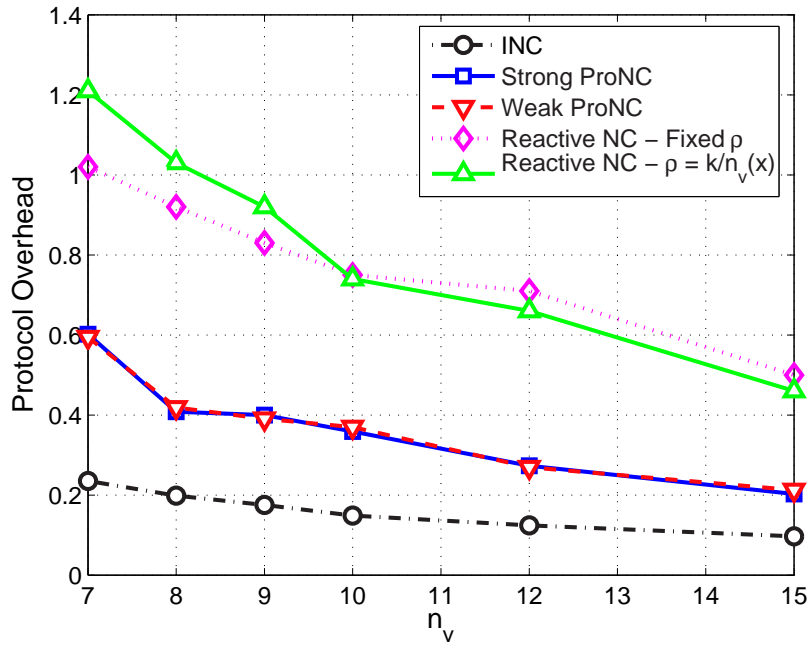


Figure 9.14 Weak and Strong ProNC against INC and reactive schemes

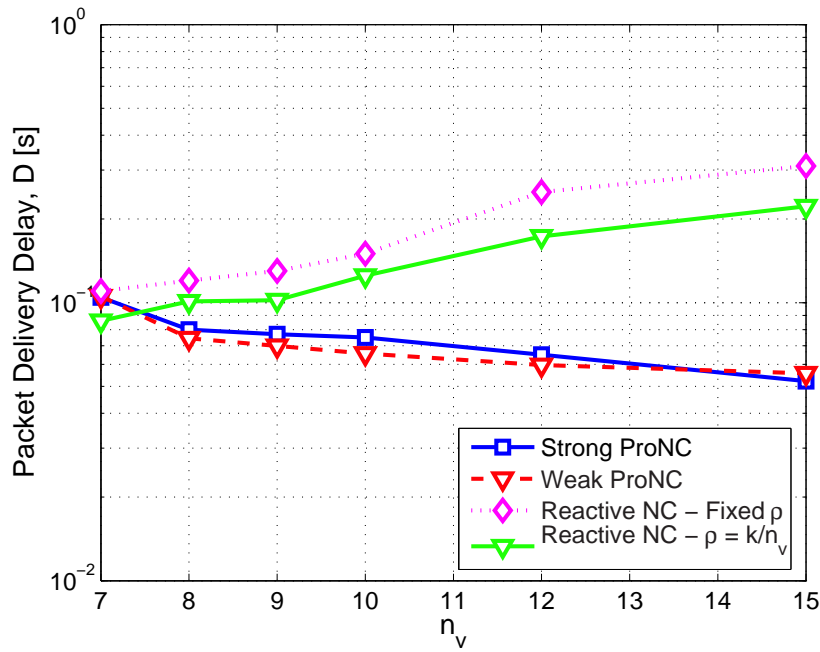


Figure 9.15 Weak and Strong ProNC against reactive schemes

# 10

## The MIMO\_NC Approach

---

*The important thing is not to stop questioning.*

(Albert Einstein)

In the previous Chapters, we discussed how network coding strategies can improve the network throughput, or equivalently reduce the number of transmissions, by transmitting coded packets. We studied such schemes in practical environments where collisions and interference produce a lot of packet losses. In particular, phenomena as fading, shadowing, multipath, collisions and interference can affect the packet transmissions leading to a high error probability. Note that a loss of data, when network coding is in use, can strongly affect the decoding phase. Indeed, as a single coded packet carries information related to multiple original packets, if it is lost, the matrix  $G$  within the receiver nodes could be not invertible. Thus, on the one hand, network coding is an effective strategy to disseminate data in a wireless environment as it guarantees that all the operations are implemented in a distributed way and they can be completed with limited energy consumption. On the other hand, these strategies are particularly sensitive to the packet losses. Hence, to guarantee satisfactory performance in terms of reliability we need to transmit more combined packets thus affecting the efficiency of network coding approach.

A lot of strategies, at the transport layer, have been proposed to alleviate the problem of packet losses. All of them try to increase the redundancy on the system by retransmitting, in some way, the packets. Unfortunately, this reduces the throughput and, again, affects the network performance. Thus, our aim is to find an approach to increase the network coding robustness to the packet losses without affecting its performance in terms of throughput and energy efficiency, i.e., keeping low the number of transmissions required to disseminate data.

In this Chapter, we propose a novel approach that jointly combines network coding and techniques whose purpose is to reduce the packet error probability.

We focus on wireless ad hoc networks affected by fading and interference. In this environments, it is well known that, for instance, MIMO techniques can achieve good performance as they exploit the so called *spatial diversity*. MIMO techniques represent nowadays one of the most studied topics in the radio communication field and great effort has been spent to develop efficient solutions. We believe that coupling MIMO and network coding could lead to optimal performance and represent an original

research topic.

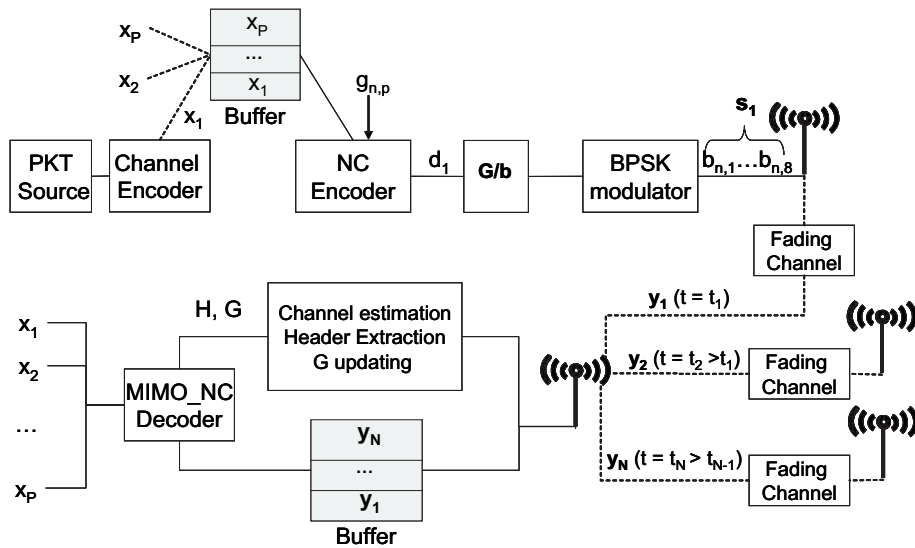
Recovering a vector of received information units from a vector of received samples is one of the key issues in MIMO technology [53]. In this case, the transmitter may send multiple symbols from different antennas, and the receiver must recover them. The transmitted symbols and the received samples are linked by the channel matrix, and the receiver must perform a vector symbol detection. For MIMO this means, for example, ML or layered detection [54]. MIMO detection can be also used, for instance, when different nodes with a single antenna send their information units to a single receiver. In all these cases, all the received energy by the destination node is used to better decode the original data in a noisy environment as the one we are considering.

Analogously, we can note that network coding approach inherently has, by nature, some spatial diversity. A node can receive some combined packets related to the same original information from different directions and over different channel. Actually, this fact is not exploited by network coding protocols as the coding and decoding phases are implemented at network layer, i.e. after the successfully reception of the packets.

The idea to jointly combine network coding and MIMO come from these observations and from the following intuition. Network coding paradigm and MIMO detection techniques are based on a similar description of the system. In both cases, the information units received by a destination are given by a linear system represented by a matrix and the vector of the original information units to be transmitted. In MIMO case the matrix is the channel matrix, in network coding the matrix is the coding matrix. However this equivalence is useful to develop a integrated system where MIMO and network coding coexist together at the same layer. In order to achieve this goal we need to move all network coding operations towards the physical layer and to design a different decoding phase based on soft decoding rather than on linear systems. In the rest of the document we refer to this approach as *MIMO\_NC*.

Such an approach can be used for different purposes.

- First, *MIMO\_NC* can be used instead of classic network coding to increase the robustness of network coding protocols by efficiently exploiting the spatial diversity offered by network coding. In the classical network coding approach, packets, coming from different positions over different channels, are separately received and decoded thus wasting a lot of redundancy and energy that could be used for a joint decoding in a noisy environment. These packets could be decoded together by the use of some MIMO techniques.
- Second, *MIMO\_NC* might increase the efficiency of cooperative communication schemes. Usually, in wireless networks, when an error occurs, the source node is required to retransmit the packet. When channel is affected by fading, exploiting spatial diversity can be convenient. Thus, the retransmission of a corrupted packet is carried out by a relay node on behalf of the source



**Figure 10.1** The MIMO\_NC encoding/decoding procedure.

node. These redundant transmissions reduce the network capacity but also increase the area involved by a single transmission. To make more efficient cooperative schemes we need to reduce the number of retransmissions or cope a retransmission phase with the transmission of a new packet. This last idea might be implemented by means of MIMO\_NC.

In this Chapter we focus only on the first possible application of MIMO\_NC as this thesis is mainly related to the data dissemination problems. To deepen the problems and the possible solutions related to the second application of MIMO\_NC see [P10].

## 10.1 Description of MIMO\_NC

The complete flow of the MIMO\_NC scheme is represented in Fig. 10.1. We want to point out that all nodes are equipped with a single antenna and this system is MIMO because multiple inputs (the IUs) are coded together by a NC matrix  $G$  to create the multiple outputs (the coded packets) that further result in multiple received packets at the destinations. The encoding process is performed by each node in a distributed fashion, thus potentially providing spatial diversity. MIMO provides advanced signal processing techniques for packet decoding, while a true antenna array is helpful but not necessary.

The encoding phase starts at the channel encoder, where the MAC layer packets are channel coded. The resulting bits are clustered and mapped into Galois symbols (here, the reference field is  $GF(2^8)$ ). We call the channel encoded PDUs *information units* (IUs). The IUs created by the node, along with those previously decoded, are stored in a buffer. Throughout the Chapter, for ease of notation, we

assume that each IU contains a single Galois symbol.<sup>1</sup> The number of available IUs in the network is  $P$ , and the symbol in the  $p$ -th IU is denoted as  $x_p, p = 1, \dots, P$ . These IUs are linearly combined so as to create a *coded packet* (CP) where the Galois symbol of the  $n$ -th CP is denoted by  $d_n, n = 1, \dots, N$ . The header of each CP includes the NC random coefficients  $g_{np}$  used to combine the IUs. The Galois symbols are turned into bits (G/b operation in Fig. 10.1), and to each Galois symbol  $d_n$  corresponds a vector  $\mathbf{s}_n$  of 8 BPSK symbols  $\{b_{n,1}, \dots, b_{n,8}\}$ . The corresponding waveforms are sent through the wireless channel, which is assumed to be frequency flat, and are collected by the receiver.

The destination performs coherent channel estimation and extracts the NC coefficients from the header. Should the header be corrupted, the packet must be discarded because the NC coefficients cannot be retrieved. In all the other cases, the receiver stores the packet into a buffer and updates its estimate of the  $G$  matrix. This buffer keeps all physical layer packets related to the same generation, i.e., the same set of IUs. These CPs have been received at different times and from different sources. Whenever early or full decoding is possible, the node starts the detection process. The number of received packets that can be used for detection is denoted as  $N$ . The received samples are gathered into column vectors of 8 elements  $\mathbf{y}_n$ . The  $N$  vectors  $\mathbf{y}_n$  with  $1 \leq n \leq N$ , that belong to the same generation are stacked on top of each other, so as to build an  $8N$  vector  $\mathbf{y}$ . The channel matrix,  $G$  and  $\mathbf{y}$  are passed to the MIMO\_NC decoder.

In conventional NC, each packet is separately demodulated and the NC coefficients are extracted from the packet header. Each node can thus form a system of equations with elements in  $\text{GF}(2^8)$ . If  $P = N = 2$ , the system looks as follows:

$$\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

which can be solved by the Gaussian elimination procedure. Note that the two problems of demodulation and NC decoding are carried out in separated stages. Classical NC can accept only packets successfully processed by PHY, whereas any other packets can not be used.

However, these operations can be jointly performed in order to exploit spatial diversity as much as possible. Since ML detection/decoding achieves optimal performance and is conceptually simple, the MIMO\_NC decoder adopts it. Let us focus on the  $N = P = 2$  case for ease of notation, and with no loss of generality. Then the system to decode is:

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix} \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} + \begin{pmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{pmatrix}$$

<sup>1</sup>This does not involve any loss in generality since different symbols in a packet are detected independently of each other, whereas joint detection is performed *across* coded packets.



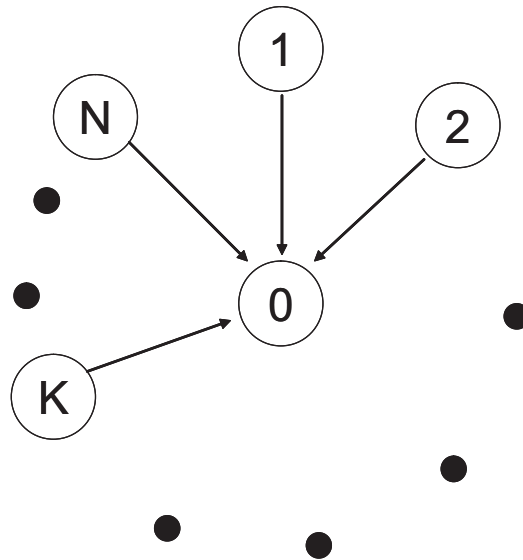
where  $H_n$  is the  $8 \times 8$  identity matrix multiplied by  $h_n$ , the gain of the  $n$ -th channel,  $\mathbf{s}_n$  is the vector of 8 BPSK symbols that represents the Galois symbol  $d_n$  and  $\boldsymbol{\eta}_n$  is a vector of 8 independent Gaussian random variables with zero mean and variance  $\sigma^2$ . Therefore, for any combination of input Galois elements  $x_1, x_2$ , there is a well defined set of output modulated waveforms. The ML criterion picks the  $x_1, x_2$  that minimize the distance between the expected received symbols  $[H_1\mathbf{s}_1(x_1, x_2); H_2\mathbf{s}_2(x_1, x_2)]$  and the actual samples  $\mathbf{y}_1, \mathbf{y}_2$ . For general  $N$  and  $P$ , an exhaustive search can be computationally infeasible, but past research has found ways to speed up this process. For instance, the NC matrix can be considered as the channel encoding matrix of a non binary system. Therefore the given problem can be cast as a joint MIMO demodulation (decode a vector of digital symbols from a vector of received samples) and channel decoding. An efficient, ML solution to this problem has been offered by [55], which is a modification of the famous sphere decoding algorithm (see [55] and references therein).

In order to describe this algorithm, we must first rewrite the above Galois system. Any Galois matrix  $G$  can be written as  $\Pi G = LU$ , where  $\Pi$  is a permutation matrix,  $L$  is lower triangular and  $U$  is upper triangular [56]. Since, for Galois fields,  $\Pi^{-1} = \Pi$ , it stems that  $G = \Pi LU$ . Therefore the problem can be decomposed into two subproblems:

$$\mathbf{y} = H\Pi LU\mathbf{x} + \boldsymbol{\eta} = H\Pi LZ + \boldsymbol{\eta}, \quad \mathbf{z} = U\mathbf{x} \quad (10.1)$$

where a dummy  $P \times 1$  vector  $\mathbf{z}$  is introduced, so that the easier problem  $\mathbf{y} = H\Pi LZ + \boldsymbol{\eta}$  needs to be solved. Given a solution  $\mathbf{z}^*$ ,  $\mathbf{x}$  is easily found by conventional backsubstitution [56], since  $U$  is upper triangular. The problem  $\mathbf{y} = H\Pi LZ + \boldsymbol{\eta}$  is easier than the full one because  $L$  is lower triangular.

We assume here, with no loss of generality, that the received packets are already ordered so that  $\Pi$  is the identity matrix. We recall that  $\mathbf{y}_n$  is the 8-element column vector that includes the components of  $\mathbf{y}$  whose index is between  $8(n-1)$  and  $8n-1$ ,  $1 \leq n \leq N$ . Moreover,  $\mathbf{L}_n$  denotes the row vector that is composed by the  $\min(n, P)$  leftmost elements of the  $n$ -th row of  $L$  and finally  $\mathbf{z}_n$  is the vector made up by the first  $\min(n, P)$  elements of  $\mathbf{z}$ . The algorithm picks the vector  $\mathbf{z}$  that minimizes the distance  $\|\mathbf{y} - HL\mathbf{z}\|^2$ , which can be written, with a slight abuse of notation, as the sum of  $N$  components:  $\sum_{n=1}^N \gamma_n^2 = \sum_{n=1}^N \|\mathbf{y}_n - h_n \mathbf{L}_n \mathbf{z}_n\|^2$ . The term  $h_n \mathbf{L}_n \mathbf{z}_n$  must be regarded as the multiplication of the scalar  $h_n$  and the BPSK symbols that stem from the Galois symbol  $\mathbf{L}_n \mathbf{z}_n$ . We note that the  $n$ -th component depends only on the first  $\min(n, P)$  symbols in  $\mathbf{z}$ . The sphere decoder finds a tentative solution for  $\mathbf{z}_1$  and computes  $\gamma_1^2$ . If this value is smaller than a certain threshold  $\rho^2$ , the *squared sphere radius*, it will proceed considering  $\mathbf{z}_2$  keeping the present estimate for  $\mathbf{z}_1$ . Otherwise, the next tentative value for  $\mathbf{z}_1$  will be considered. Given a tentative solution for the first  $k$  symbols, the decoder will proceed by decoding the  $(k+1)$ -st element if  $\sum_{n=1}^k \gamma_n^2 < \rho^2$ . The great advantage of the sphere decoder is that if the metric of a certain solution  $S$  is too large, all subsequent solutions which share  $S$



**Figure 10.2** The test network.

as a prefix need not be considered.

In summary, each node will collect the packets, decode the header, extract the NC coefficients and then keep the received soft samples. The node tries to decode as many transmitted packets as possible with the collected frames. Should it fail (because a packet has been corrupted by interference or noise), it will store the received samples and keep them so as to help the decoding of the next packets. To avoid error propagation, nodes are allowed to combine and retransmit only information units that have been successfully decoded. Finally, we note that in conventional MIMO the diversity is due to the presence of multiple antennas. MIMO\_NC, instead, may exploit three types of diversity: spatial due to the different positions of nodes, temporal due to the different transmission times and coding due to redundant linear combinations of IUs, if present.

## 10.2 Performance Analysis

Let us consider a simple case study that can be quite easily analyzed. The sample network is reported in Fig. 10.2 where nodes 1 through  $N$  have the same  $P$  IUs. Each of them transmits a coded packet (which is a random linear combination of the  $P$  original packets). Node 0 collects these  $N \geq P$  coded packets and tries to recover the original frames. This scenario can happen in a network where data dissemination has reached several nodes, thus many terminals can combine several packets at once. In this case it is common that some nodes transmit to the same receiver several coded packets based on the same information units [39].

### 10.2.1 Classical NC Performance

The analysis for conventional NC is quite straightforward in this scenario. We shall assume that if  $P$  CPs out of  $N$  are correctly decoded, the original IUs can all be recovered.<sup>2</sup> If fading is constant over a whole packet, and it is frequency flat and Rayleigh distributed, the packet error probability  $P_{pk}$  is inversely proportional to the SNR [54]. For NC,  $P$  out of  $N$  CPs must be correctly decoded. Therefore the error probability is the cumulative distribution function of the sum of  $N$  binary random variables evaluated at  $P - 1$ .

Let us consider the special case of all the channel gains  $h_n$  being independent and identically distributed. The probability of receiving fewer than  $P$  correct packets out of  $N$  is:

$$P_{err} = \sum_{k=0}^{P-1} \binom{N}{k} (1 - P_{pk})^k P_{pk}^{N-k} \quad (10.2)$$

For small  $P_{pk}$  the most likely error event is that exactly  $P - 1$  packets have been correctly decoded. In this case, the packet error probability is approximately:

$$P_{err} \simeq \binom{N}{P-1} P_{pk}^{N-(P-1)} = \binom{N}{P-1} P_{pk}^{N-P+1} \quad (10.3)$$

In Rayleigh fading,  $P_{pk} \propto 1/SNR$ , thus  $P_{err} \propto 1/SNR^{N-P+1}$  and the diversity order is  $N - P + 1$ .

### 10.2.2 MIMO\_NC Performance

The equivalent input/output relation for MIMO\_NC is reported in Eq. (10.1). The computation of the exact error probability is rather hard. Instead, we shall pursue the pairwise error probability.

For MIMO\_NC, let us call *codeword* the vector which contains the Galois symbols prior to the combination of network coding. We shall denote the codewords by the symbols  $\mathbf{c}_i$ , where  $i$  is an integer index. By definition  $\mathbf{c}_0$  is the all zero codeword and it is assumed to be the transmitted codeword in order to compute the pairwise error probability. This is not restrictive since the matrix  $G$  is a linear operator.

The pairwise error probability of deciding for another codeword  $\mathbf{c}_1$  instead of  $\mathbf{c}_0$  is the conditioned probability that:

$$\|HG\mathbf{c}_0 - \mathbf{y}\|^2 > \|HG\mathbf{c}_1 - \mathbf{y}\|^2 \quad (10.4)$$

given that  $\mathbf{c}_0$  was sent. After some algebra, Eq. (10.13) becomes (see [57]):

$$(HG(\mathbf{c}_0 - \mathbf{c}_1))^T \mathbf{y} < 0 \quad (10.5)$$

<sup>2</sup>This approximation does not consider the negligible probability that the NC matrix may not be inverted. This probability decays as  $1/(256^{(N-P+1)})$

The inner product  $(HG(\mathbf{c}_0 - \mathbf{c}_1))^T \mathbf{y}$  is the sum of  $8N$  terms. Let us define  $t_n, n \in \{1, \dots, N\}$  as the sum of the terms whose index goes from  $8(n-1) + 1$  to  $8n$ :

$$t_n = \sum_{k=1}^8 [h_n(b_{n,k}^{(0)} - b_{n,k}^{(1)})][h_n b_{n,k}^{(0)} + \eta_{n,k}] \quad (10.6)$$

where  $b_{n,k}^{(\ell)}$  is the  $8(n-1) + k$ -th modulated BPSK symbol of the  $\ell$ -th codeword ( $\ell \in \{0, 1\}$ ) and  $\eta_{n,k}$  is the  $k$ -th element of  $\eta_n$ ,  $k \in \{1, \dots, 8\}$ . Since  $\mathbf{c}_0 = \mathbf{0}$ , it follows that  $G\mathbf{c}_0 = \mathbf{0}$  and  $b_{n,k}^{(0)} = -1, \forall n, k$ . Clearly each of the terms that make up  $t_n$  is non zero if  $b_{n,k}^{(0)} \neq b_{n,k}^{(1)}$ .<sup>3</sup> Let  $w_n$  be the number of different bits in the  $n$ -th Galois symbol between  $\mathbf{c}_0$  and  $\mathbf{c}_1$  ( $w_n \in \{0, 1, \dots, 8\}$ ). After some algebra, the decision statistics  $t = \left(\sum_{n=1}^N t_n\right)/2$  is found as:

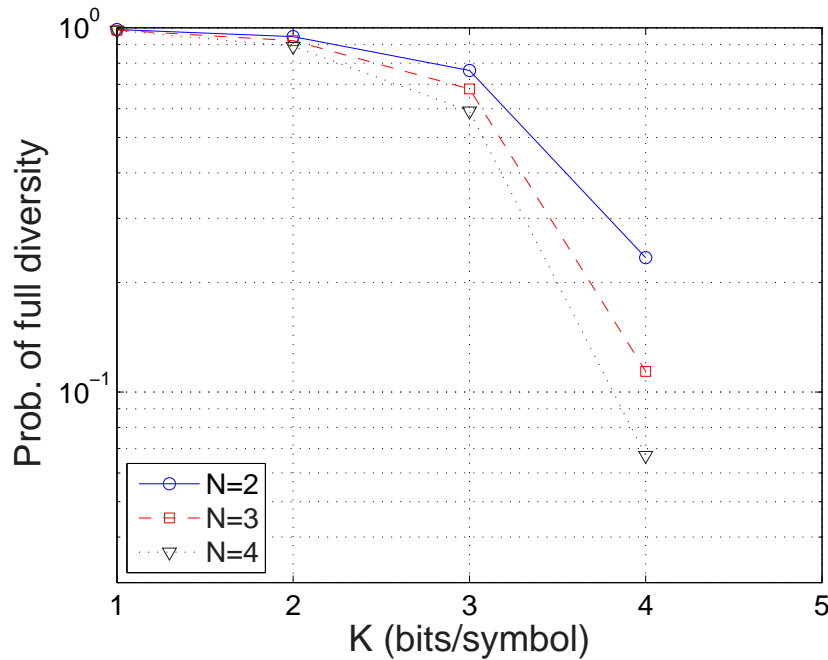
$$t = \sum_{n=1}^N h_n^2 w_n + \sum_{n=1}^N h_n \sum_{k=1}^8 ((b_{n,k}^{(0)} - b_{n,k}^{(1)})/2) \eta_{n,k} \quad (10.7)$$

There is a decoding error if  $t < 0$ . In Eq. (10.7), the first term  $\left(\sum_{n=1}^N h_n^2 w_n\right)$  is a deterministic number (since we assume the codewords to be known). Instead,  $\sum_{k=1}^8 ((b_{n,k}^{(0)} - b_{n,k}^{(1)})/2) \eta_{n,k}$  is the sum of  $w_n$  independent Gaussian random variables and its variance is  $w_n \sigma^2$ . Therefore  $\sum_{n=1}^N h_n \sum_{k=1}^8 ((b_{n,k}^{(0)} - b_{n,k}^{(1)})/2) \eta_{n,k}$  has zero mean and variance  $\sum_{n=1}^N h_n^2 w_n \sigma^2$ . The overall decision statistics is thus a Gaussian random variable with mean  $\sum_{n=1}^N h_n^2 w_n$  and variance  $\sum_{n=1}^N h_n^2 w_n \sigma^2$  [57]. Thus the error probability conditioned to the channel state is:

$$\begin{aligned} P_{block} &= Q \left( \sqrt{\frac{(\sum_{n=1}^N h_n^2 w_n)^2}{(\sum_{n=1}^N h_n^2 w_n) \sigma^2}} SNR \right) \\ &= Q \left( \sqrt{\frac{\sum_{n=1}^N h_n^2 w_n}{\sigma^2}} SNR \right) \end{aligned} \quad (10.8)$$

A few observations can be made. First of all, the error probability averaged on the fading statistics has diversity order equal to the number of non-zero  $w_n$ . If  $G$  is regarded as the generator matrix of a linear block code, the number of non-zero  $w_n$  is the minimum Hamming distance of the code. The best case occurs when the code achieves the Singleton bound [58], i.e., the minimum distance is  $N - P + 1$ . From a MIMO point of view, our system, in some sense, decodes a V-BLAST transmission with ML decoding. In this setting it is well known that the diversity order is  $N$ , not  $N - P + 1$ , where  $P$  now is the number of transmitted streams [54]. However, there is no real contradiction between these two facts. The intuitive reason is the following. In real MIMO systems, the channel matrix is real, and not

<sup>3</sup>A term may vanish also if  $h_n b_{n,k}^{(0)} + \eta_{n,k} = 0$ , but this is a zero probability event.

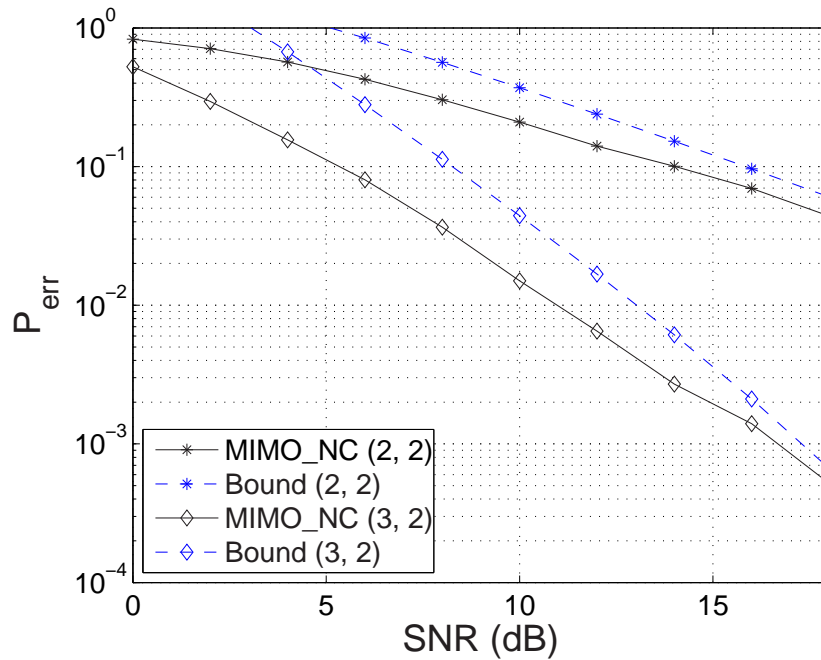


**Figure 10.3** Probability of full order diversity in NC by varying the input rate

a hybrid of Galois symbols and real numbers. Therefore the probability that a codeword may force to zero some received samples is negligible. Instead, with the Galois-valued matrix  $G$  there is with probability 1 a codeword that forces  $P - 1$  outputs to zero.

We also observe that our analysis predicts that the diversity order is one if  $N = P$ . Moreover, a conventional cooperative decode-and-forward system is a particular case of our system with  $P = 1$ . Our formula correctly states that the diversity order would be  $N - P + 1 = N$  [59].

It is clear that conventional NC encoding does not properly exploit the spatial diversity inherent in the system, because the sizes of the fields of NC coefficients and input symbols are equal, while in true MIMO this is not the case. Thus we have explored what would be the performance of a MIMO\_NC system whose input symbols are drawn from the field  $\text{GF}(2^K)$ ,  $1 \leq K \leq 8$ . This strategy effectively reduces the codebook and the rate. Since the codebook is smaller, there are fewer words that the ML decoder of MIMO\_NC may be confounded with. In particular, also the words that differ for  $N - P + 1 = N - 1$  elements from the correct codeword are fewer, and if there are none of them the diversity is  $N$ . Fig. 10.3 shows, for  $P = 2$ ,  $N \in \{2, 3, 4\}$ , what is the probability of having diversity  $N$  instead of  $N - 1$  by varying  $k$ . It is apparent that there is full diversity with high probability only for  $k \leq 3$ , which entails an unacceptable rate reduction. This shows that 1) the encoding phase of NC as it has been known so far is not suitable to exploit spatial diversity, and 2) there might exist a smart NC encoding scheme that could overcome this problem, but so far it implies heavy rate losses,



**Figure 10.4** Comparison of simulated MIMO\_NC and Union Bound when  $P = 2$  and  $N = 2, 3$ .

that diversity alone cannot justify. On the other hand, MIMO\_NC does achieve the maximum possible diversity order and outperforms classic NC, since it offers a SNR gain and can exploit packets that would not be considered by conventional NC (see the Results Section).

Even though the diversity order is the same for MIMO\_NC and NC, the former can successfully decode the transmitted data in many situations where NC would fail, because the joint detection and decoding can succeed even if the single packets are corrupted. In these cases NC could not even start recovering the data (see the next Section).

In order to check the correctness of our analysis, we have compared the Union Bound [58] for MIMO\_NC when  $P = 2$  and  $N = 2, 3$  with the simulated MIMO\_NC (Fig. 10.4). It turns out that 1) the analysis is validated since it correctly predicts the diversity order and 2) the union bound is quite accurate since it converges for high SNR towards the simulated curve.

### 10.3 Simulation Results

In this Section we prove the effectiveness of MIMO\_NC in different network configurations. Our aim is to show in which scenarios MIMO\_NC can achieve significant gains with respect to the classical network coding approach.

We analyze the MIMO\_NC approach and classical network coding in the simple topology described

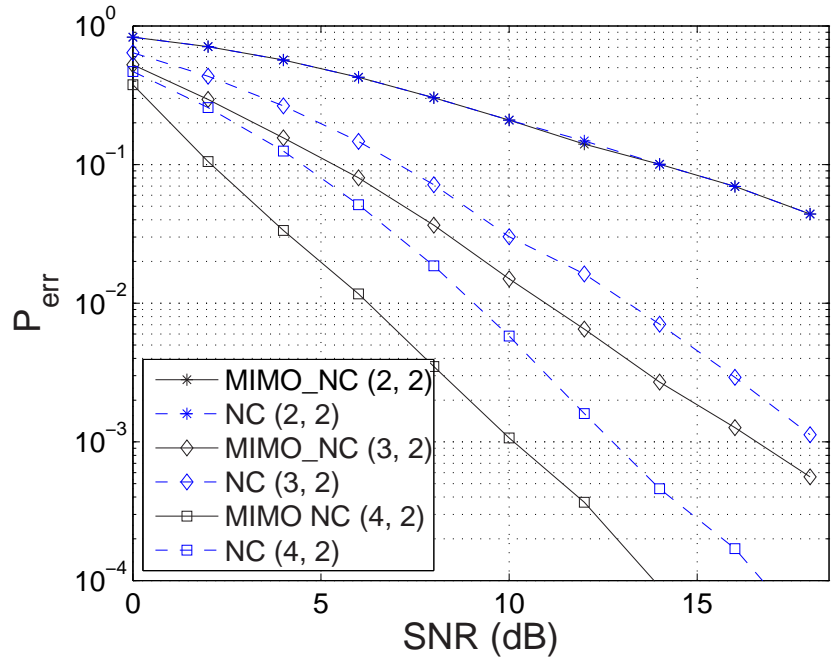
in Fig. 10.2 by varying  $P$  and  $N$ , where  $P$  is the number of original packets that can be combined together and  $N$  is the number of combinations of  $P$  original packets received by the destination. We name this scenario *Star Topology*, which represents a common situation when NC is used to disseminate data in a wireless distributed network [52]. Usually, a node can receive from its neighbors multiple packets which are linear combinations of the same IUs. In addition, it often receives more packets than it needs due to the data dissemination scheme in use. In these cases, according to the classical network coding approach, these redundant packets are dropped. On the contrary, they can be used by MIMO\_NC to better decode the original packets in case of errors.

The main difference between MIMO\_NC and the classical network coding algorithm is the ability of MIMO\_NC to exploit spatial diversity, thus decreasing the error probability. For this reason, we mainly focus on the system error probability which is defined as the probability that the destination cannot successfully decode all the  $P$  packets.

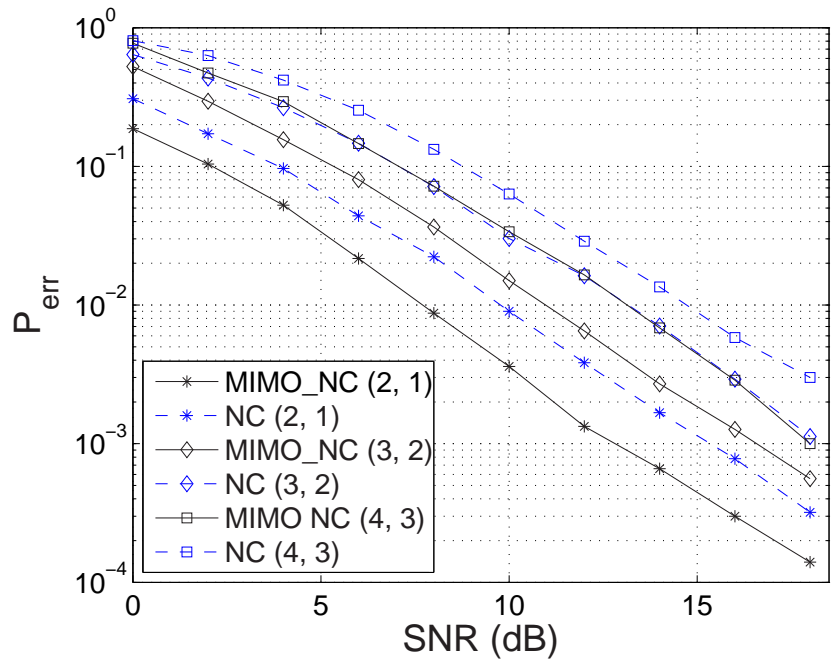
Fig. 10.5 considers different cases by varying  $N$ . The system error probability of MIMO\_NC and classical NC for fixed  $P = 2$  is shown. First of all, we observe that when  $N = P$  the two schemes achieve similar performance. In this case, the system introduces no redundancy and there is little diversity for MIMO\_NC to exploit. As  $N$  increases, the performance gap between the two schemes widens, being 2 and 3 dB for  $N$  equal to 3 and 4, respectively. This shows that for increasing  $N$  our scheme performs better and better, and thus is able to reap the advantages of joint demodulation and network coding. Moreover, note that the slope of the curves is exactly  $N - P + 1$  as predicted by the theoretical analysis.

Fig. 10.6 analyzes the performance when  $N = P + 1$  for different  $P$ . The performance gap between the two schemes may be expected to grow with the generation size. We note that the marginal improvement for  $P > 2$  is rather small, and therefore most of the benefits can be already reaped for small generation sizes, with low computational complexity.

Finally, we mention that the Star Topology represents a worst case scenario because we consider a destination node with an empty buffer. On the contrary, when data is disseminated via NC in a distributed network, nodes usually have some packets stored in their buffers. These packets could have been already decoded or not. In the former case they effectively reduce  $P$  as they do not need to be estimated in the MIMO\_NC decoder. In the latter, they increase  $N$  since they are an additional set of received samples. In both situations, the difference between  $N$  and  $P$  increases and therefore the gain of MIMO\_NC over NC is boosted.



**Figure 10.5** System error probability: performance comparison of MIMO\_NC and classical NC when  $P = 2$  and  $N = 2, 3, 4$ .



**Figure 10.6** System error probability: performance comparison of MIMO\_NC and classical NC when  $N = P + 1$ .



## 10.4 From MIMO\_NC to Super MIMO\_NC

MIMO\_NC is an important step to exploit the redundancy implicit in NC, but the quest for the diversity inherent in NC can succeed only if both the encoding and decoding phases are redesigned together. Thus, the previous Sections sparked the question of creating a NC/PHY layer that is as efficient as NC but can also exploit the diversity inherent in packet combining and retransmission. This Section deals with this problem, and proposes two novel systems. Firstly, we design a new encoding NC phase for MIMO\_NC, called Super MIMO\_NC, that achieves a higher diversity order than MIMO\_NC. Secondly, we will explore how the choice of the modulation affects the diversity order of the transmission scheme, by developing a rate-adaptive strategy called Adaptive MIMO\_NC. We remark that any solution requires knowledge from both networking and signal processing theory, and thus it is an interesting problem of cross-layer design.

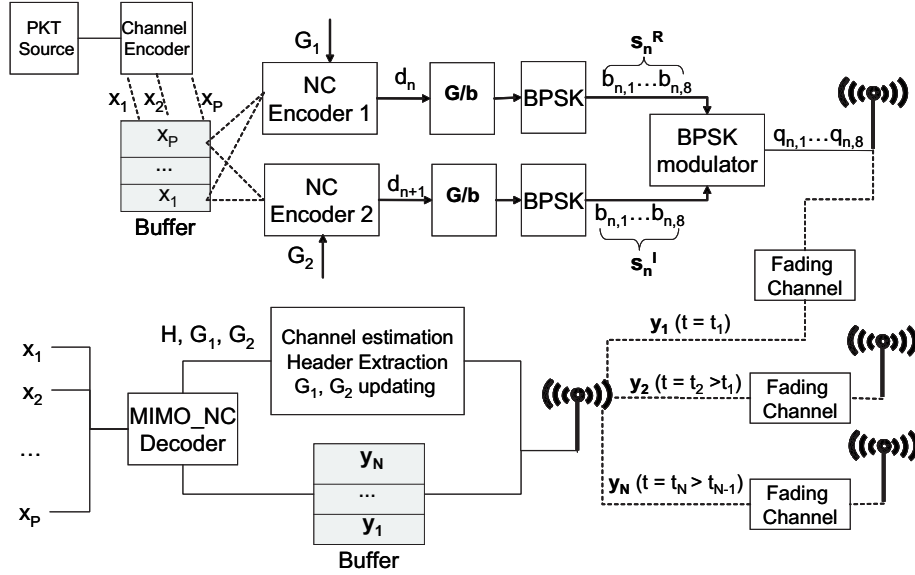
## 10.5 Super MIMO\_NC

The previous Sections highlighted the following conclusions about the diversity order of MIMO\_NC:

1. the diversity order of NC and MIMO\_NC is  $N - P + 1$  as long as the NC coefficients and the IUs are expressed by the same field;
2. MIMO\_NC can achieve higher diversity if the input symbols of the IUs belong to a smaller field than the NC coefficients, while standard NC cannot improve this metric under the same conditions;
3. higher diversity may imply heavy rate losses;
4. encoding and decoding must be jointly designed to achieve the spatial diversity inherent in NC.

One possible way to avoid this problem is to increase the number of symbols sent in each CP. In conventional NC, every CP is composed by one linear combination of the IUs. Instead, we propose each CP to include two linear combinations of the IUs. Such a system will be called Super MIMO\_NC.

The coding phase of Super MIMO\_NC is reported in Fig. 10.7. It is equal to that of conventional MIMO\_NC, but rather than a single NC matrix  $G$ , two random NC matrices  $G_1$  and  $G_2$  are used. Each of them combines the same IUs. In standard MIMO\_NC, the ensuing Galois symbols generate BPSK symbols; instead, in Super MIMO\_NC a Galois symbol from  $G_1$  creates the real parts of 8 consecutive QPSK symbols, while the outputs of  $G_2$  generate the imaginary parts. Super MIMO\_NC creates twice as many symbols as conventional MIMO\_NC, and this is the price to pay to improve the detection probability. All the NC coefficients are stored in the packet header.



**Figure 10.7** Super MIMO\_NC system overview.

The QPSK symbols  $q_n$  are sent through the wireless channel. Each CP undergoes flat fading, and the Rayleigh fading channel coefficient is a complex circularly symmetric Gaussian random variable  $h_n$ . Thus, let us focus on a generic received sample  $y_n^4$  of the  $n$ -th received coded packet:

$$\begin{aligned}
 y_n &= y_n^{(r)} + jy_n^{(i)} = \\
 &= (h_n^{(r)} + jh_n^{(i)})(q_n^{(r)} + jq_n^{(i)}) + \eta_n^{(r)} + j\eta_n^{(i)} = \\
 &= (h_n^{(r)}q_n^{(r)} - h_n^{(i)}q_n^{(i)}) + j(h_n^{(r)}q_n^{(i)} + h_n^{(i)}q_n^{(r)}) + \\
 &\quad + \eta_n^{(r)} + j\eta_n^{(i)}
 \end{aligned}$$

where  $\eta_n^{(r)} + j\eta_n^{(i)}$  is a complex valued, circularly symmetric Gaussian noise with variance  $\sigma^2/2$  per component. This relation can be turned into matrix form:

$$\begin{pmatrix} y_n^{(r)} \\ y_n^{(i)} \end{pmatrix} = \begin{pmatrix} h_n^{(r)} & -h_n^{(i)} \\ h_n^{(i)} & h_n^{(r)} \end{pmatrix} \begin{pmatrix} q_n^{(r)} \\ q_n^{(i)} \end{pmatrix} + \begin{pmatrix} \eta_n^{(r)} \\ \eta_n^{(i)} \end{pmatrix} \quad (10.9)$$

The received signal is complex valued. The decoding phase is carried out by means of Sphere Decoding (SD), which requires real signal processing in its most common version. Generally speaking, for  $N$  received CPs, the equivalent input-output relation can be written as [60]:

$$\begin{pmatrix} \mathbf{y}^{(r)} \\ \mathbf{y}^{(i)} \end{pmatrix} = \begin{pmatrix} H^{(r)} & -H^{(i)} \\ H^{(i)} & H^{(r)} \end{pmatrix} \begin{pmatrix} \mathbf{q}^{(r)} \\ \mathbf{q}^{(i)} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\eta}^{(r)} \\ \boldsymbol{\eta}^{(i)} \end{pmatrix} \quad (10.10)$$

<sup>4</sup>Throughout this Section, the real (imaginary) part of a complex number  $a$  is denoted by  $a^{(r)}$   $a^{(i)}$

where  $H^{(r)}$  and  $H^{(i)}$  are  $8N \times 8N$  diagonal matrices and  $\boldsymbol{\eta}^{(r)}$ ,  $\boldsymbol{\eta}^{(i)}$  are real-valued vectors made up by  $8N$  iid Gaussian noise samples. The diagonal of  $H^{(r)}$  ( $H^{(i)}$ ) is made up by  $N$   $8 \times 8$  matrices  $\{H_n^{(r)}\}$ ,  $0 \leq n \leq N - 1$  ( $H_n^{(i)}$ ). Each  $H^{(r)}$  ( $\{H^{(i)}\}$ ) is equal to the  $8 \times 8$  identity matrix multiplied by  $h_n^{(r)}$  ( $h_n^{(i)}$ ). We point out that  $H = \begin{bmatrix} H^{(r)} & -H^{(i)} \\ H^{(i)} & H^{(r)} \end{bmatrix}$  has the following property:

$$\begin{aligned} H' &= H^T H = \begin{pmatrix} H^{(r)} & H^{(i)} \\ -H^{(i)} & H^{(r)} \end{pmatrix} \begin{pmatrix} H^{(r)} & -H^{(i)} \\ H^{(i)} & H^{(r)} \end{pmatrix} = \\ &= \begin{pmatrix} H^{(r)2} + H^{(i)2} & 0 \\ 0 & H^{(r)2} + H^{(i)2} \end{pmatrix} \end{aligned} \quad (10.11)$$

which is a diagonal matrix. Each element in the diagonal of  $H^{(r)2} + H^{(i)2}$  is exponentially distributed, and the elements at index  $\{8n, \dots, 8(n+1) - 1\}$  are the squared envelopes of the channel seen by the  $n$ -th packet.

Therefore the received samples  $\mathbf{y}$  must be left-multiplied by the  $H^T$  matrix, so as to diagonalize the equivalent  $H$  matrix. At this point, the decoder has to solve a system of the type:

$$\mathbf{Y} = H' G \mathbf{x} + \boldsymbol{\eta}' \quad (10.12)$$

where  $\mathbf{Y}$  is the  $16N$  vector of the received samples (the first and last  $8N$  are respectively the real and imaginary parts of the complex-valued received samples),  $H' = H^T H$ ,  $G = [G_1; G_2]$  and  $\boldsymbol{\eta}' = H^T \boldsymbol{\eta}$ . The real (imaginary) part of  $k$ -th noise sample of the  $n$ -th CP  $\eta_{n,k}^{(r)}$  ( $\eta_{n,k}^{(i)}$ ) is a zero mean Gaussian random variable with variance  $|h_n^2| \sigma^2 / 2$ , since it can be expressed as  $h_n^{(r)} \eta_{n,k}^{(r)} \mp h_n^{(i)} \eta_{n,k}^{(i)}$ . This problem is formally identical to what has been solved in conventional MIMO\_NC by means of sphere decoding [55]. In this case, the matrix  $G$  has size  $2N \times P$  and the diagonal  $H'$  matrix is  $16N \times 16N$ .

### 10.5.1 Performance Analysis

We shall prove in this Section that Super MIMO\_NC can improve the diversity order of the system. The performance of the system can be analyzed by means of the pairwise error probability. Given that the all zero codeword  $\mathbf{c}_0$  was sent, the decoder decides for a different codeword  $\mathbf{c}_1$  if:<sup>5</sup>

$$\|H' G \mathbf{c}_0 - \mathbf{y}\|^2 > \|H' G \mathbf{c}_1 - \mathbf{y}\|^2 \quad (10.13)$$

given that  $\mathbf{c}_0$  was sent. After some algebra, Eq. (10.13) becomes:

<sup>5</sup>For ease of notation, as in 10.2 we assume in this analysis that each IU contains a single Galois symbol. This does not involve any loss in generality since different symbols in a packet are detected independently of each other, whereas joint detection is performed *across* coded packets.

$$(H'G(\mathbf{c}_0 - \mathbf{c}_1))^T \mathbf{y} < 0 \quad (10.14)$$

The passage stems from the fact that  $\|H'G\mathbf{c}_0\|^2$  is the energy of the vector of modulated symbols. Since each of them has constant envelope, the energy of  $\|H'G\mathbf{c}_0\|^2$  or  $\|H'G\mathbf{c}_1\|^2$  does not depend on the codeword  $\mathbf{c}_0$  or  $\mathbf{c}_1$ .

Let us assume that a QPSK symbol is received with average power  $P_r$ . Then, after some algebraic steps [57] and closely following the analysis for standard MIMO\_NC 10.2, the decision statistic is:

$$\begin{aligned} t &= \sum_{n=0}^{N-1} |h_n|^2 w_n \sqrt{\frac{P_r}{2}} + \\ &+ \sum_{n=0}^{N-1} \sum_{k=0}^7 ((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2) \eta_{n,k}'^{(r)} \\ &+ \sum_{n=0}^{N-1} \sum_{k=0}^7 ((b_{n,k}^{(i,0)} - b_{n,k}^{(i,1)})/2) \eta_{n,k}'^{(i)} \end{aligned} \quad (10.15)$$

where  $b_{n,k}^{(r,0)}$  ( $b_{n,k}^{(i,0)}$ ) is the in phase (in quadrature) bit of the  $k$ -th ( $0 \leq k \leq 7$ ) QPSK symbol in the  $n$ -th packet,  $\eta_{n,k}'^{(r)}$  ( $\eta_{n,k}'^{(i)}$ ) is the real (imaginary) part of the  $k$ -th noise sample for the  $n$ -th coded frame and  $w_n = w_n^{(r)} + w_n^{(i)}$ ,  $w_n^{(r)} = \sum_{k=0}^7 ((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2)$ ,  $w_n^{(i)} = \sum_{k=0}^7 ((b_{n,k}^{(i,0)} - b_{n,k}^{(i,1)})/2)$ .  $w_n^{(r)}$  ( $w_n^{(i)}$ ) is the number of non zero bits in the  $n$ -th component of  $G_1(\mathbf{c}_0 - \mathbf{c}_1)$  ( $G_2(\mathbf{c}_0 - \mathbf{c}_1)$ )

There is a decoding error if  $t < 0$ . In Eq. (10.15), the first term  $\sum_{n=0}^{N-1} |h_n|^2 w_n \sqrt{P_r/2}$  is a deterministic number (since we assume the codewords to be known). Instead,  $\sum_{k=0}^7 ((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2) \eta_{n,k}'^{(r)} + \sum_{k=0}^7 ((b_{n,k}^{(i,0)} - b_{n,k}^{(i,1)})/2) \eta_{n,k}'^{(i)}$  is the sum of  $w_n$  independent Gaussian random variables and its variance is  $w_n \sigma^2 / 2 |h_n|^2$ . Therefore  $\sum_{n=0}^{N-1} \sum_{k=0}^7 ((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2) \eta_{n,k}'^{(r)}$  has zero mean and variance  $\sum_{n=0}^{N-1} |h_n|^2 w_n \sigma^2 / 2$ . The overall decision statistics is thus a Gaussian random variable with mean  $\sum_{n=0}^{N-1} |h_n|^2 w_n \sqrt{P_r/2}$  and variance  $\sum_{n=0}^{N-1} |h_n|^2 w_n \sigma^2 / 2$ , and hence the error probability is:

$$\begin{aligned} P_{err} &= Q \left( \sqrt{\frac{(\sum_{n=0}^{N-1} |h_n|^2 w_n)^2 P_r / 2}{(\sum_{n=0}^{N-1} |h_n|^2 w_n) \sigma^2 / 2}} \right) = \\ &= Q \left( \sqrt{\sum_{n=0}^{N-1} |h_n|^2 w_n \Lambda} \right) \end{aligned} \quad (10.16)$$

where  $\Lambda = P_r / \sigma^2$  is the average Signal to Noise Ratio. A few observations can be inferred. First of all, the diversity order is equal to the number of different  $|h_n|^2$  that are present in the argument of the

Gaussian complementary cumulative distribution function (the Q function). On one hand, there can be at most  $N$  of them, because there are  $2N$  terms but each of them is repeated twice. On the other hand, if  $G$  is regarded as the generator matrix of a linear block code, the number of non-zero  $w_n^{(r)}$  and  $w_n^{(i)}$  is the minimum Hamming distance of the code. The best case occurs when the code achieves the Singleton bound [58], i.e., the minimum distance is  $2N - P + 1$ . Therefore,  $P - 1$  terms in Eq. (10.16) may disappear. The diversity order decreases by one if, for the same  $n$ ,  $w_n^{(r)} = w_n^{(i)} = 0$ . Since at most  $P - 1$   $w_n^{(r)}$  or  $w_n^{(i)}$  terms can vanish, the diversity order can be lowered at most by  $\lfloor (P - 1)/2 \rfloor$  and thus the slope of the Packet Error Rate vs SNR curve is at least:

$$D(N, P) = N - \left\lfloor \frac{P - 1}{2} \right\rfloor \quad (10.17)$$

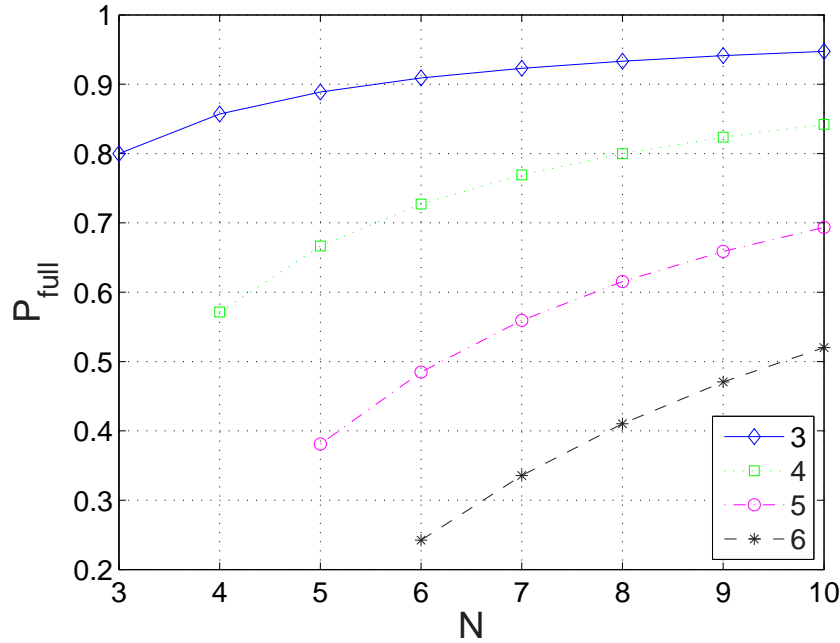
If  $P = 2$ , one term vanishes, but no diversity is lost, since each channel is present twice in Eq. (10.16). Therefore, the diversity order is always  $N$ . If  $P > 2$ , the diversity order will be smaller than  $N$  if the terms relative to the same channel vanish (i.e.,  $\exists n : w_n^{(r)} = w_n^{(i)} = 0$ ). This event can be analyzed by falling back on a closely related problem: given  $2N$  balls indexed from 0 to  $2N - 1$ ,  $P - 1$  of them are removed. What is the probability that one even index and the following odd index are drawn? The probability of not choosing any two balls in a forbidden configuration is as follows. Each time the  $k$ -th ball is moved out, the next ball (which can be drawn from  $2N - k$  positions) should not be picked from any of the  $k$  urns such that a ball with an even index is followed by an odd index. This probability is  $(2N - 2k)/(2N - k)$ . Thus the probability of having diversity  $N$  is:

$$P_{full} = \prod_{k=1}^{P-1} \frac{2N - 2k}{2N - k} \quad (10.18)$$

Fig. 10.8 plots this equation for different values of  $N$  and  $P$ . It is clear that for large  $N$  and fixed  $P$  the probability of having full diversity approaches 1, because all the factors in Eq. (10.18) go to one. However, the rate of convergence is quite slow. This speed can be estimated for large  $N$  as:

$$\begin{aligned} 1 - P_{full} &= 1 - \prod_{k=1}^{P-1} \left( 1 - \frac{k}{2N - k} \right) \simeq \sum_{k=1}^{P-1} \frac{k}{2N - k} \simeq \\ &\simeq \frac{1}{2N} \sum_{k=1}^{P-1} k = \frac{P(P - 1)}{4N} \end{aligned} \quad (10.19)$$

This equation reveals that the system approaches full diversity with a speed that decays inversely with  $N$ , but the number of combined packets has a quadratic weight  $P(P - 1)$ .

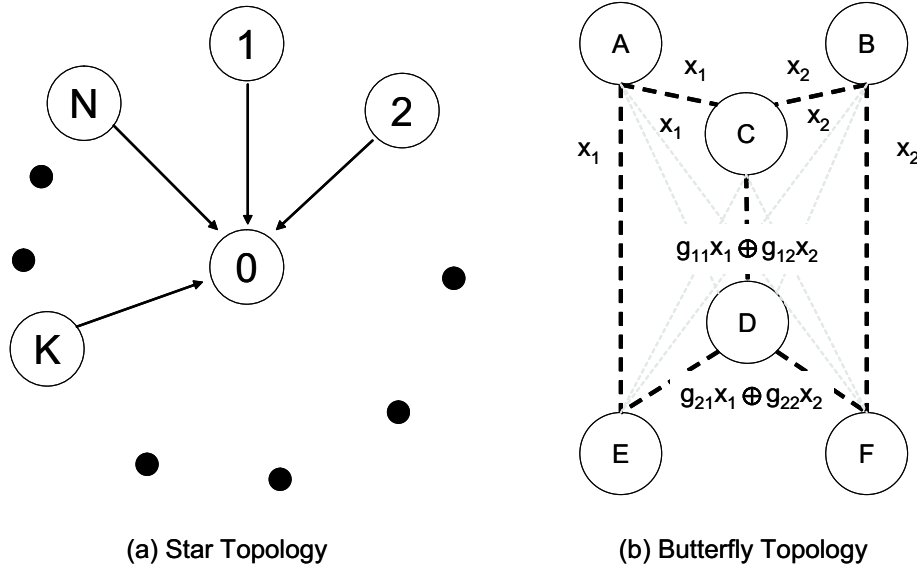


**Figure 10.8** Probability of having full diversity for Super MIMO\_NC for  $P \in \{3, 4, 5, 6\}$

### 10.5.2 Adaptive MIMO\_NC

The previous analysis has shown that Super MIMO\_NC can guarantee a diversity order of at least  $N - \lfloor \frac{P-1}{2} \rfloor$ . However, this comes at the price of a reduced transmission rate, since a more spectrally efficient constellation has been used but the effective data rate has not been changed. According to the situation, it may be desirable to have lower error probabilities or higher transmission rates. In particular, we note that when there is little redundancy at the receiver ( $N = P$ ), the error rates of standard MIMO\_NC can be quite high. Therefore it may be desirable to quickly reduce the error rate in the early data dissemination stages. Hence, we propose a simple rate adaptation scheme which works as follows. The SNRs of the received coded packets (even the corrupted or redundant ones) are stored and sorted. If the strongest  $P$  SNRs are larger than a threshold  $T$  (which is a design parameter), then the error rate is assumed to be sufficiently low and thus standard MIMO\_NC is used. If it is not the case, Super MIMO\_NC is employed, in order to reduce the error rate down to acceptable levels. We shall call such a scheme Adaptive MIMO\_NC.

We conclude this Section by noting an important fact: in adaptive MIMO\_NC, the CPs are transmitted according to either MIMO\_NC or Super MIMO\_NC, and the decoder can demodulate/decode a set of CPs which have been sent according to different schemes, as soon as the adopted modulation scheme is known. This enables the nodes to decide which strategy to employ in a completely dis-



**Figure 10.9** The test networks.

tributed fashion, without any exchange of information to coordinate them. Thus MIMO\_NC and Super MIMO\_NC can seamlessly coexist, and this is another reason that makes Adaptive MIMO\_NC viable.

## 10.6 Performance Results

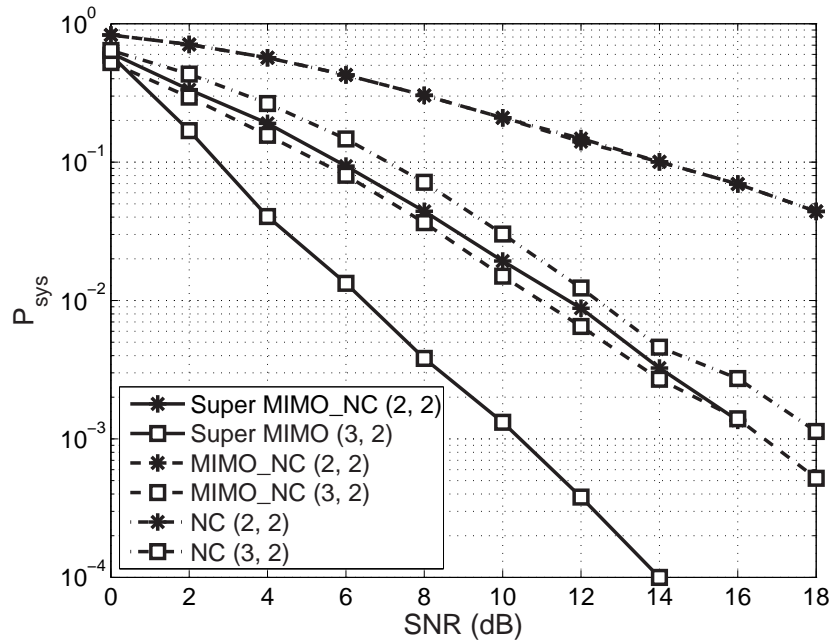
In this Section we prove the effectiveness of Super MIMO\_NC and Adaptive MIMO\_NC in different network configurations by comparing them with the basic MIMO\_NC scheme and the classical NC approach.

We focus on two different scenarios. First, we consider the simple topology described in Fig. 10.9a by varying  $N$  and  $P$ . This scenario will be named *Star Topology*. Second, we compare the performance of the different versions of the MIMO\_NC scheme in the well known network topology presented in [36], here referred to as *Butterfly Topology* (Fig. 10.9b).

The main difference between classical network coding, MIMO\_NC and Super MIMO\_NC is the ability of the schemes based on MIMO\_NC to exploit spatial diversity, thus decreasing the error probability. For this reason, we mainly focus on the system error probability  $P_{sys}$ , defined as the probability that at least one of the destination nodes does not receive one or more packets intended for it.

### 10.6.1 Star Topology

We report in this Section simulation results about the Super MIMO\_NC performance compared with the basic MIMO\_NC approach and classical NC in the Star topology which is representative of



**Figure 10.10** System error probability, star topology,  $P = 2$

common situations where, for instance, data is disseminated through random network coding protocols. Our main aim is to highlight the benefits obtained by Super MIMO\_NC in achieving a higher diversity order.

In the Star configuration,  $N$  nodes  $(1, \dots, N)$  share the same  $P$  IUs and they are charged to send them to the destination, i.e., the central node 0. In Fig. 10.10 we report the system error probability,  $P_{sys}$  (i.e., the probability that node 0 can not decode at least one of the  $P$  IUs) for the case  $P = 2$  and  $N = 2, 3$ . We point out that the diversity order changes according to Eq. (10.17). Let us focus on the case  $P = 2$  and  $N = 2$ . In this situation, it can be noted that MIMO\_NC and NC achieve the same performance while Super MIMO\_NC guarantees both a higher diversity order and substantial gains (e.g., about 8 dB at  $P_{sys} = 10^{-2}$ ) over the other schemes. This proves that Super MIMO\_NC can obtain significant performance improvements also when  $N$  and  $P$  are the same. The same behavior can be observed also when  $N = P + 1$ . In this case, MIMO\_NC guarantees a gain of about 2 dB over the classical NC approach but Super MIMO\_NC can achieve a lower  $P_{sys}$  with a gain of about 6 and 8 dB over MIMO\_NC and NC, respectively. This analysis proves that Super MIMO\_NC is a good solution to enhance the reliability of MIMO\_NC. In the next Section we evaluate its performance in a more realistic and complex scenario.



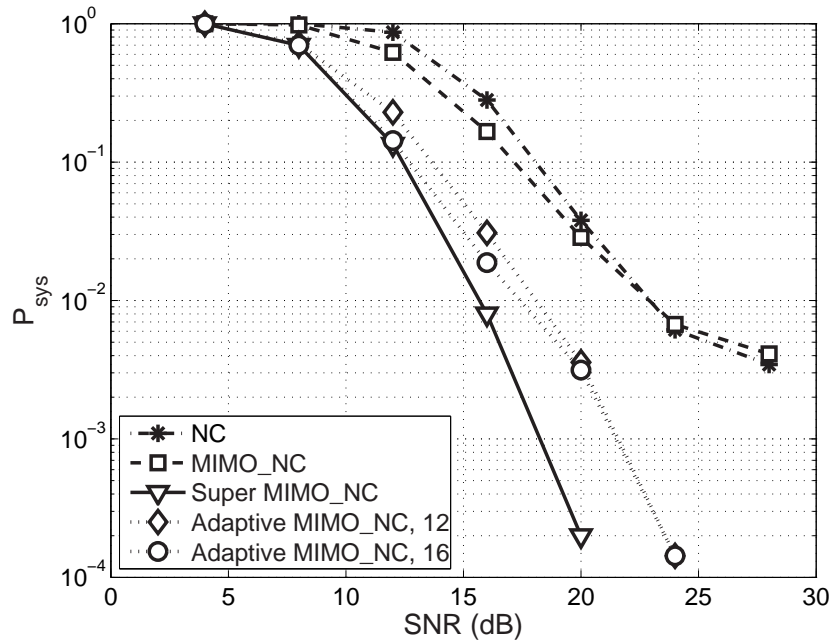
## 10.6.2 Butterfly Topology

In this Section, we consider the Butterfly Topology (see Fig. 10.9b) which is one of the best known reference scenarios for network coding [36]. Let  $A$  and  $B$  be two source nodes which generate two original packets  $x_1$  and  $x_2$ . Nodes  $E$  and  $F$  are the destinations and they want to successfully receive both  $x_1$  and  $x_2$ . Each of the intermediate nodes,  $C$  and  $D$ , transmits a packet which is a linear combination of  $x_1$  and  $x_2$ . Note that in this situation, intermediate nodes can transmit some packets only if they can successfully recover some of the IUs. This means that the destinations can receive between two and four combined packets depending on how many intermediate nodes retransmit. This scenario is a little bit more complex than the previous one as nodes are placed at different distances and only the two source nodes have the IUs at the beginning.

The system error probability, in this case, is defined as the probability that at least one of the two destinations does not successfully receive at least one of the two original packets. Fig. 10.11 compares the system error probability of Super MIMO\_NC, MIMO\_NC, NC and Adaptive MIMO\_NC with two values for the SNR threshold  $T$  (i.e.,  $T = 12$  dB and  $T = 16$  dB). We first observe that Super MIMO\_NC always guarantees better performance with a gain of 4 dB over the basic MIMO\_NC scheme and about  $1 \div 2$  dB over Adaptive MIMO\_NC. The plot shows that the error probability curves of Super MIMO\_NC and Adaptive MIMO\_NC are comparable and they are steeper than those of MIMO\_NC and NC. Finally, we note the both MIMO\_NC and NC show the same behavior for high SNR values. The slope decreases and the curves flatten to the constant value of  $1/256$ . This is due to the fact that in MIMO\_NC and classical NC, the two sources may send linearly dependent CPs. If it is the case, the received  $2 \times 2$   $G$  matrices are not invertible and neither the relays nor the destinations can decode any packet. Such an event happens with probability  $1/256$ , which is exactly the observed value.

The advantages of the Adaptive MIMO\_NC scheme are pointed out in Fig. 10.12 where the average transmission rates of all schemes normalized to the transmission rate of MIMO\_NC are shown. As expected, MIMO\_NC guarantees fastest transmissions and, on the contrary, the transmission rate of Super MIMO\_NC is halved. Therefore, Adaptive MIMO\_NC can achieve higher transmission rates than Super MIMO\_NC guaranteeing also good performance in terms of error probability.

Some observations related to the average transmission rate of Adaptive MIMO\_NC can be inferred. We observe that, in the Butterfly topology, there can be at most four transmissions. Let us assign to the rate of each transmission with MIMO\_NC the reference value of 1. Thus, the average transmission rate for MIMO\_NC is 1. According to this assumption, the average transmission rate of Super MIMO\_NC is 0.5 as each packet is transmitted at half the rate of MIMO\_NC. Using the adaptive scheme we can achieve at most an average transmission rate equal to 0.75 as the source nodes communicate at rate 0.5 while the relay nodes can use at most a rate equal to 1, therefore the maximum transmission rate

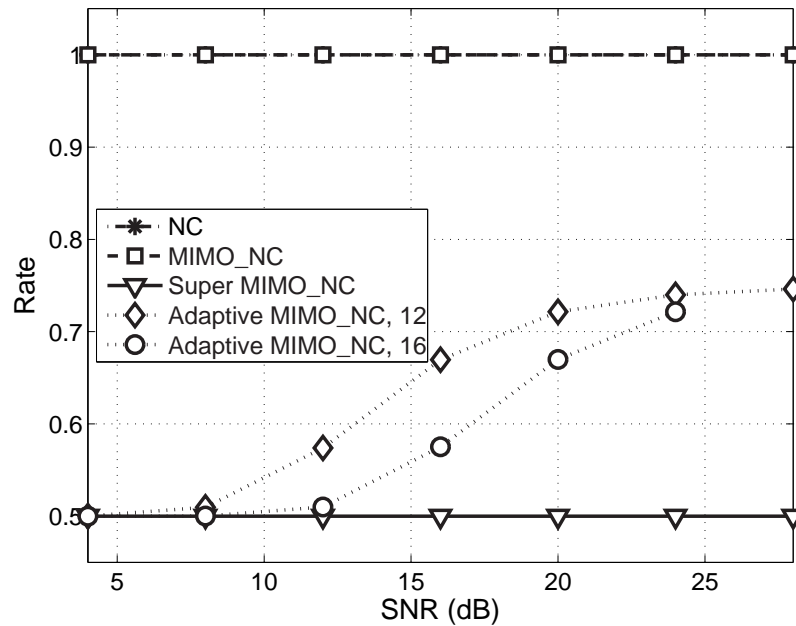


**Figure 10.11** System error probability, butterfly topology

averaged over all nodes is  $(0.5 + 0.5 + 1 + 1)/4 = 0.75$ . In addition, the average transmission rate achieved by the adaptive scheme depends on the selection of the threshold  $T$ . The higher the threshold value, the lower the transmission rate. Moreover, for high average SNR, the relay nodes never need to employ Super MIMO\_NC, therefore the top transmission rate (0.75) is reached. This proves that the idea of tuning the modulation scheme according to the number of collected packets is a promising approach to increase both the diversity gain and the transmission rates of MIMO\_NC systems.

## 10.7 Conclusions

In this Chapter, we proposed a scheme which jointly combines NC and MIMO in order to achieve more robustness with respect to packet losses. The basic idea comes from the fact that NC and MIMO systems can be described by similar equations and so they can be easily integrated. Nevertheless, to achieve this goal, we have to move NC functionalities towards the physical layer and implement a more sophisticated decoding phase in order to exploit spatial diversity. We name this scheme MIMO\_NC and we prove its effectiveness by both theoretical and simulation analysis. We focus on a simple network configuration which is significant in the context of data delivery via NC. The obtained results show that MIMO\_NC is a promising approach as it can strongly increase the system performance in terms of system error probability when communications are affected by fading.



**Figure 10.12** Achieved rate, butterfly topology

Secondly, we have proposed an improvement to the encoding procedure in the MIMO\_NC system. The main aim of such an approach is to increase the maximum diversity order achieved by MIMO\_NC in order to reduce the system error probability. We have named the novel scheme Super MIMO\_NC and we have compared its performance against classical NC and basic MIMO\_NC, showing that a higher diversity order is effectively achieved. In addition, we developed also a hybrid scheme, named Adaptive MIMO\_NC, which trades off reliability against transmission rate. Finally, we have proved its effectiveness in a well known network configuration.

Finally, we want to point out that the application of MIMO\_NC presented in this Section is only one of the possible uses of our approach. We give a tool which can be implemented in different environments for different purposes leading to interesting future approaches. We mention that, for instance, MIMO\_NC can be applied also in cooperative communications to increase the efficiency of the system.



# 11

## Conclusions

---

*Observe Everything.  
Communicate Well.  
Draw, Draw, Draw.*  
(Frank Thomas)

In this thesis we dealt with the problem to efficiently disseminate data in pervasive systems. To distribute a great amount of data to multiple users is an important service/application that modern networks have to offer at different network layers. At the application level there are a lot of interesting applications related to this: file sharing, files downloads, broadcast of video streams and so on. At lower levels, networking protocols often require to distribute control, topology, service information to all nodes in order to organize and maintain the network operability. Thus, data dissemination scheme is a crucial service which has to be implemented in an efficient way. Achieving this goal in wireless environments involves a lot of challenging issues due to the characteristics of the medium and to the lack of synchronization and organization among devices. In this thesis, we focused on two different applications which require a proper data dissemination scheme. The choice was aimed at pointing out the most interesting peculiarities, impairments and constraints related to the data dissemination problem. The first application deals with the data distribution of alert messages in inter-vehicular networks while the second one consists on spreading data over a large pervasive system via network coding.

The main outcomes of this study can be summarized as follows.

- *Dissemination of alert messages in inter-vehicular networks* - This kind of application requires the development of a reliable data dissemination scheme which can offer very low latency. Thus, we have designed a MAC protocol for disseminating data which minimizes the time required to deliver data to all interested nodes. The proposed protocol, named Smart Broadcast Protocol (SB), is a position-based scheme which does not require any network knowledge, it can be implemented in a distributed way and it is robust to the node mobility and to variable node density. We have analyzed its performance from both a theoretical point of view and via simulations. We have proposed also a way to optimize the design parameters of SB. Finally, we have compared SB with other existing schemes in order to point out that our optimization strategy leads to better

performance than the existing solutions. In particular, minimizing the time required to broadcast the message on one hop seems to be a better strategy than maximizing the maximum one hop advancement.

- *Data dissemination in pervasive system* - The main aim of a data dissemination scheme in such a context is to reduce as much as possible the number of transmission required to deliver a certain amount of data to all nodes. This means saving network resources in terms of energy, bandwidth, and so on. To achieve this goal in a distributed way we have decided to exploit the functionalities offered by network coding, a recent network paradigm whose aim is just the reduction of the number of transmission in order to increase the network throughput. It consists on the introduction of a data processing phase in addition to the simple data dissemination scheme. A lot of work has proved the effectiveness of network coding in ideal settings. Our contribution on this topic is manifold.

First, we dealt with practical aspects by analyzing data dissemination schemes based on network coding. This study has pointed out that network coding is particularly sensitive to realistic MAC and physical protocols thus, it can not achieve the theoretical performance also in practical settings. The main problems are related to the packet losses caused by interference and collisions. But also the random access mechanisms, such as the CSMA/CA, strongly affect the performance as they can not guarantee the good packet mixing needed by network coding strategy.

Second, we have proposed a different network coding approach to solve the problems highlighted by the previous analysis. Our proposal consists on a proactive network coding scheme, named *ProNC*, which outperforms the existing data dissemination protocols based on network coding and guarantees performance similar to those protocols which have a complete network knowledge.

Third, we have observed that network coding is particularly sensitive to the packet losses thus its robustness has to be increased to guarantee good performance also in noisy environments. We have focused on channel affected by fading and we have proposed a scheme which integrates network coding and MIMO techniques in a single communication system. The aim is to exploit both the benefits of network coding, i.e., the reduction of the number of transmissions and the potentiality of MIMO, i.e., the use of the spatial diversity. We have proved that the proposed scheme, namely *MIMO\_NC*, can strongly reduce the packet error probability with respect to the classical network coding approach. Hence, *MIMO\_NC* can represent a valid and more robust alternative to the classical network coding paradigm and it can further increase the performance of the data dissemination schemes.

The analysis of such application scenarios gave us the opportunity to face most of the interesting

problems related to the data dissemination. In particular, on the one hand we have investigated how it is possible to guaranteed reliability and low latency in a distributed way. On the other hand, we focused on the reduction of the network resource consumption to develop schemes with a low impact on the network itself. Both solutions can found interesting applications in different kind of networks such as WSNs or mesh networks. Our network coding schemes, for instance, can also be applied in the dissemination of multiple queries or they can be used to gather data from sensors. The MIMO\_NC approach can be useful not only to increase the data dissemination performance but also, as an example, to develop cooperative MIMO systems which exploit the benefit of network coding. The Smart Broadcast Protocol could be used also in real time applications and enhanced to work in more complex urban area by including a two dimensional management of nodes.





# 12

## Author's Publications

---

- P1** E. Fasolo, R. Furiato and A. Zanella, “Smart Broadcast for inter-vehicular communications”, in *Proc. of WPMC05, Sep.2005, Alborg Denmark*.
- P2** E. Fasolo, C. Prehofer, M. Rossi, Q. Wei, J. Widmer, A. Zanella and M. Zorzi, “Challenges and new approaches for efficient data gathering and dissemination in pervasive wireless networks”, in *Proc. of INTERSENSE06, May 2006, Nice, France*.
- P3** E. Fasolo and A. Zanella, “An effective broadcast scheme for alert message propagation in vehicular ad hoc networks”, in *Proc. of ICC06, Jun. 2006, Istanbul, Turkey*.
- P4** L. Badia, E. Fasolo, A. Paganini and M. Zorzi, “Data Aggregation Algorithms for Sensor Networks with Geographic Information Awareness”, in *Proc. of WPMC06, Sep. 2006, S.Diego, CA, US*.
- P5** E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, “On MAC Scheduling and Packet Combination Strategies for Practical Random Network Coding”, in *Proc of ICC07, Jun. 2007, Glasgow, UK*.
- P6** E. Fasolo, F. Maguolo, S. Ruffino, P. Stupar, A. Zanella and M. Zorzi, “VoIP Communications in Wireless Ad Hoc Network with Gateways”, in *Proc of MediaWin, Jul. 2007, Aveiro, Portugal*.
- P7** E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, “A Proactive Network Coding Strategy for Pervasive Wireless Networking”, in *Proc. of Globecom 2007, Nov. 2007, Washington, DC, US*.
- P8** E. Fasolo, F. M. Rossetto and M. Zorzi, “Network Coding meets MIMO”, in *Proc. of NetCod2008, Jan. 2008, Hong Kong*.
- P9** E. Fasolo, F. M. Rossetto and M. Zorzi, “On encoding and rate adaptation for MIMO\_NC”, *accepted to ISCCSP 2008, Feb. 2008, Malta*.
- P10** E. Fasolo, A. Munari, F. M. Rossetto and M. Zorzi, “Phoenix: A Hybrid Cooperative-Network Coding Protocol for Fast Failure Recovery in Ad Hoc Networks”, *submitted to Secon 2008*.
- J1** E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, “In-network aggregation techniques for wireless sensor networks”, *IEEE Wireless Communication Magazine, vol. 14, n. 2, Apr. 2007, 70-87*.



# Bibliography

- [1] S. Katti, H. Rahul, W. Huss, D. Katabi, M. Medard, and J. Crowcroft, “XORs in The Air: Practical Wireless Network Coding,” in *ACM SIGCOMM*, Pisa, Italy, Sep. 2006.
- [2] E. Fasolo, C. Prehofer, M. Rossi, Q. Wei, J. Widmer, A. Zanella, and M. Zorzi, “Challenges and new approaches for efficient data gathering and dissemination in pervasive wireless networks,” in *INTER-SENSE*, Nice, France, Apr. 2006.
- [3] IEEE LAN MAN Standards, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*”, ANSI/IEEE Std., March 1999.
- [4] IEEE LAN MAN Standards, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band*”, ANSI/IEEE Std., September 1999.
- [5] IEEE LAN MAN Standards, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*”, ANSI/IEEE Std., June 2003.
- [6] S. Ni, Y. Tseng, Y. Chen, J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proc. of ACM/IEEE MOBICOM’99*, Aug. 1999.
- [7] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [8] B. Williams and T. Camp, “Comparison of Broadcast Techniques for Mobile Ad-Hoc Networks,” in *MOBIHOC*, 2002.
- [9] A. Zanella and G. Pierobon and S. Merlin, “On the limiting performance of broadcast algorithms over unidimensional ad-hoc radio networks,” in *Proceedings of WPMC04*, Abano Terme, Padova, Sep. 2004.

- [10] Y. Tseng and S. N. E. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," in *The 21st International Conference on Distributed Computing Systems*, 2001.
- [11] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, November/December 2001.
- [12] G. Korkmaz, E. Ekici, F. Özgüner, and . Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *in Proc. VENET'04*, Oct. 2004.
- [13] M. Sun, W. Feng, T. Lai, K. Yamada, H. Okada, and K. Fujimura, "Gps-based message broadcasting for inter-vehicle communication," in *Proceedings of ICPP*, 2000.
- [14] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. of ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2000.
- [15] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks," in *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, Jan. 2002.
- [16] J. Wu, "An enhanced approach to determine a small forward node set based on multipoint relay," in *in Proc. VTC'03*, 2003.
- [17] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *MobiHoc 2002*, Lausanne, Switzerland, Jun. 2002.
- [18] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. IEEE ICDCSW'01*, 2001.
- [19] A. McDonald and T. Znati, "Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks," in *Proc. 34th Annual Simulation Symp.*, 2001.
- [20] J. Yu and P. Chong, "3hbc (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks," in *Proc. IEEE Pacrim'03*, Aug.2003.
- [21] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE JSAC*, vol. 15, pp. 1265–1275, Sept. 1997.
- [22] P. T. Eugster, R. Guerraoui, A. M. Kermarrec, and L. Massoulié, "Epidemic information dissemination in distributed system," *IEEE Computer Magazine*, vol. 37, no. 5, pp. 60–67, May 2004.
- [23] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, July 2003.

- [24] K. P. B. et al., “Bimodal multicast,” *ACM Trans. Computer System*, vol. 17, no. 2, 1999.
- [25] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” in *Technical Report CS-2000-06*, July 2000.
- [26] Y. Shavitt and A. Shay, “Optimal Routing in Gossip Networks,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 4, July 2005.
- [27] “Car 2 car communication consortium.” [Online]. Available: <http://www.car-to-car.org>
- [28] “Safespot integrated project.” [Online]. Available: <http://www.safespot-eu.org>
- [29] G. M. T. Abdalla, M. A. Abu-Rgheff, and S. M. Senouci, “Current trends in vehicular ad hoc networks,” in *UBIROADS' 2007*, Marrakech, Morocco, Jul. 2007.
- [30] H. Menouar, F. Filali, and M. Lenardi, “A survey and qualitative analysis of mac protocols for vehicular ad hoc networks,” *IEEE Wireless Communications*, vol. 13, no. 5, pp. 30–35, Mar. 2006.
- [31] J. Weatherall and A. Jones, “Ubiquitous network and their applications,” *IEEE Wireless Communications*, vol. 1, no. 9, pp. 18–29, Feb. 2002.
- [32] G. Korkmaz, E. Ekici, F. Özgüner, and . Özgüner, “Urban multi-hop broadcast protocol for inter-vehicle communication systems,” in *Proc. of the first ACM workshop on Vehicular ad hoc networks*, 2004.
- [33] M. Zorzi and R. Rao, “Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance,” *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, Oct.–Dec. 2003.
- [34] —, “Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, Oct.–Dec. 2003.
- [35] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network Aggregation Techniques for Wireless Sensor Networks: A Survey,” *IEEE Wireless Communication Magazine*, vol. 14, no. 2, pp. 70–87, Apr. 2007.
- [36] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, vol. 46, no. 4, July 2000.
- [37] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network Coding,” *IEEE Trans. on Information Theory*, vol. 49, no. 2, Feb. 2003.
- [38] P. A. Chou, T. Wu, and K. Jain, “Practical network coding,” in *41st Allerton Conf. Communication, Control and Computing*, Monticello, IL, US, Oct. 2003.

- [39] 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.
- [40] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, “A network coding approach to energy efficient broadcasting: from theory to practice,” in *IEEE Infocom*, Barcelona, Spain, Apr. 2006.
- [41] ———, “On the benefits of network coding for wireless applications,” submitted.
- [42] D. S. Lun, M. Medard, T. Ho, and R. Koetter, “Network coding with a cost criterion,” in *Technical Report P-2584*, April 2004.
- [43] S. Riis and R. Ahlswede, “Problems in network coding and error correcting codes,” in *Internal report*, 2004.
- [44] C. Gkantsidis and P. Rodriguez, “Network Coding for Large Scale Content Distribution,” in *IEEE Infocom*, Calcutta, India, Dec. 2005.
- [45] C. Fragouli and E. Soljanin, “Information Flow Decomposition for Network Coding,” *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 829–848, Mar. 2006.
- [46] D. Tuninetti and C. Fragouli, “Processing along the way: forwarding vs. coding,” in *ISITA*, Parma, Italy, Oct. 2004.
- [47] C. Fragouli and E. Soljanin, “On average throughput benefits of network coding,” in *42st Annual Allerton Conference on Communication Control and Computing*, Monticello, IL, US, Oct. 2003.
- [48] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. on Networking*, vol. 49, no. 11, Nov. 2003.
- [49] Y. E. Sagduyu and A. Ephremides, “Crosslayer Design for Distributed MAC and Network Coding in Wireless Ad Hoc Networks,” in *IEEE International Symposium on Information Theory*, Adelaide, Australia, Sep. 2005.
- [50] M. Wang and B. Li, “How Practical is Network Coding?” in *IWQoS*, New Haven, CT, USA, Jun. 2006.
- [51] D. Dubois-Ferrière, D. Estrin, and M. Vetterli, “Packet Combining in Sensor Networks,” in *ACM SenSys 2005*, San Diego, CA, USA, Nov. 2005.
- [52] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “On MAC Scheduling and Packet Combination Strategies for Random Network Coding,” in *IEEE ICC 2007*, Glasgow, UK, Jun. 2007.
- [53] D. S. P. J. S. D. Gesbert, M. Shafi and A. Naguib, “From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems,” *IEEE J-SAC*, vol. 21, no. 3, pp. 281–302, Mar. 2003.

- 
- [54] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, UK: Cambridge Univ. Press, 2005.
- [55] H. Vikalo and B. Hassibi, "On Joint Detection and Decoding of Linear Block Codes on Gaussian Vector Channels," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3330–3342, Sep. 2006.
- [56] G. H. Golub and F. V. Loan, *Matrix computations*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
- [57] E. Fasolo, F. Rossetto, and M. Zorzi, "The MIMO\_NC project," University of Padova, Tech. Rep. 110.
- [58] R. Blahut, *Theory and practice of error control codes*. Reading (MA, USA): Addison Wesley, 1983.
- [59] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 68–732, Oct. 2004.
- [60] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

