Università degli Studi di Padova

Dipartimento di Elettronica e Informatica

Scuola di Dottorato in Ingegneria dell'Informazione
Indirizzo in Ingegneria Elettronica e delle Telecomunicazioni

XXI Ciclo

Design of Multiple Access Techniques for Multihop Wireless Networks from a Physical Layer Point of View

Tesi di: *Francesco M. Rossetto*

Supervisore:
Chiar.mo Prof. Michele Zorzi

Direttore della Scuola:
Chiar.mo Prof. Matteo Bertocco

Anno Accademico 2008/2009

Ringraziamenti

Cari venticinque lettori,

(che nel frattempo sarete divenuti anche cinque e basta) dopo tre anni e mezzo ecco qui la mia nuova, colossale (almeno per me) fatica. Dopo essere passato attraverso tredici (e sperabilmente qualcuno in più) articoli, sei mesi a San Diego, quattro in Germania, una morosa, e soprattutto un dottorato, chi debbo ringraziare per tutto questo?

Un primo sentito grazie va indubbiamente al mio supervisore, Michele Zorzi. Il quale ha avuto non solo il merito di sopportare me e la mia pressapocaggine. Ma soprattutto di avermi fatto crescere come ricercatore. Se in questi anni ho esplorato molti campi di punti delle reti wireless, ho espresso idee anche originali in merito e mi sono, in un certo senso, divertito, lo devo anche lui. Ho innalzato un monumento (né più duraturo del bronzo, né più alto della regale mole delle piramidi) che porta il mio sigillo, e ne vado fiero. Ma questo monumento che ora si innalza si fonda sugli insegnamenti di passione, tenacia e bravura del mio supervisore. Grazie, Michele.

A ruota, i miei colleghi del Signet meritano un ringraziamento. Queste persone così diverse fra loro mi hanno tutte impressionato per la loro non comune preparazione ed inventiva. Nel desiderio di emularli, di provare a divenire un po' come loro, sono cresciuto ed ho imparato. Uno speciale pensiero va ad Elena e Andrea M., che mi hanno accompagnato per gran parte di questi quattro anni di ricerca; a Kostas, che oltre ad avermi stupito con le sue formule analitiche mi ha stupito dimostrando di avere un cuore, fatto poi non così comune, soprattutto a certe latitudini; a Nicola e Simone, che oltre che notevoli ricercatori si sono dimostrati strepitosi coinquilini, anche quando prendevano aceto per olio; ed infine a Federico M., che proprio non doveva andarsene anzitempo. Grazie, ragazzi.

Un grazie speciale va a molti dei miei amici (come Giovanni, Francesco, Elena, il mitico Alessandro ed il ricordo di Davide), il cui affetto, dimostratomi nei momenti meno banali, non ha prezzo. Se non sono crollato nel mezzo di questi anni così impegnativi e difficili, è in parte merito vostro. Ed un grazie va anche a Sabrina. Seppure le strade delle nostre vite si siano separate (e forse è meglio così), il mio debito è comunque grosso. Grazie, amici miei.

Ultimi, in ordine di apparizione, ma primi, in merito ed affetto, vanno i miei genitori. Con buona pace di Padoa-Schioppa (e detto da una persona che è uscita di casa a 22 anni), ne hanno il diritto, perchè hanno fatto più di quello che ogni essere (umano o divino) avrebbe loro potuto chiedere. Mi hanno confortato nei momenti più bui, mi hanno dato fiducia e mi hanno insegnato come tenere la testa alta di fronte alle avversità. Se ora sono un uomo che può tentare di affrontare il mondo e ridare i talenti che gli sono stati donati, ovvero compiere ciò per cui è stato creato, lo devo soprattutto a loro ed al loro sconfinato amore. Grazie, mamma e papà, sopra tutto e tutti.

Franz

Contents

Abstract **xi**

Sommario **xiii**

1 Introduction **1**

2 MIMO signal processing for ad hoc network MAC protocols **5**

2.1 Introduction 6

2.2 A low-delay MAC solution for MIMO Ad Hoc Networks 7

2.2.1 Antenna and Channel Models 9

2.2.2 Protocol Model 9

2.2.3 Performance Evaluation 13

2.2.4 Discussion 17

2.3 Network Coding meets MIMO: A look into the rate/diversity tradeoff in wireless Network Coding 17

2.3.1 The MIMO_NC schemes 18

2.3.2 Performance Analysis 22

2.3.3 Performance Evaluation 28

2.3.4 Discussion 33

2.4 Phoenix: A Hybrid Cooperative-Network Coding Protocol for Fast Failure Recovery in Ad Hoc Networks 34

2.4.1 Phoenix: A Cooperative - NC Protocol 35

2.4.2 Analytical Model 40

2.4.3 Performance Evaluation 42

2.4.4 Discussion 54

2.5 Chapter Conclusions 54

2.6 Acknowledgments 54

References 55

3 Interference models for Carrier Sense and Scheduling in ad hoc networks 59
3.1 Introduction . . . 60
3.2 Related Work . . . 61
3.3 A Robust approach to Carrier Sense for MIMO ad hoc networks . . . 62
3.3.1 System Model and Protocol Description . . . 63
3.3.2 The Carrier Sense System . . . 65
3.3.3 Performance Evaluation . . . 69
3.3.4 Discussion . . . 71
3.4 A low complexity Model for Interference in Wireless Networks . . . 72
3.4.1 Model Description . . . 72
3.4.2 Performance Evaluation . . . 79
3.4.3 Discussion . . . 80
3.5 Dynamic Carrier Sense Adaptation . . . 82
3.5.1 Dependence of the CS threshold on the node density . . . 82
3.5.2 A Distributed Algorithm for Threshold Optimization . . . 84
3.5.3 Performance Evaluation . . . 86
3.5.4 Discussion . . . 91
3.6 A Physical Model Scheduler for Multi-Hop Wireless Networks Based on Local Information . . . 93
3.6.1 Scheduler Design . . . 96
3.6.2 Implementation Issues . . . 97
3.6.3 Performance Evaluation . . . 100
3.6.4 Discussion . . . 106
3.7 Chapter Conclusions . . . 106
3.8 Acknowledgments . . . 107
References . . . 107

4 Delay and throughput characterization of random multi-hop networks 111
4.1 Introduction . . . 112
4.2 System model . . . 113
4.2.1 Network setting . . . 113
4.2.2 Physical layer . . . 115
4.2.3 Network metrics . . . 115
4.3 Analysis . . . 116
4.3.1 Queueing analysis . . . 116
4.3.2 Evaluation of packet success probabilities . . . 118
4.3.3 Examples . . . 122
4.4 Simulation Results . . . 125
4.5 Chapter Conclusions . . . 128
4.6 Acknowledgments . . . 129

References 129

5 Conclusions **131**

A Complete List of Papers **133**

A.1 Papers on MIMO signal processing 133

A.2 Papers on carrier sense 133

A.3 Papers on stochastic geometry and other topics 134

Published or submitted papers 134

List of Figures

2.1 BER Comparison for CRCM and STBC 13
2.2 Throughput vs effective load, 12 nodes 15
2.3 Throughput vs effective load, 12 nodes 15
2.4 Delay vs Throughput, 12 nodes 16
2.5 Delay vs Throughput, 12 nodes 16
2.6 Energy per information bit vs Throughput, 12 nodes 17
2.7 MIMO_NC system overview. 19
2.8 The test networks. 23
2.9 Comparison of simulated Basic MIMO_NC and Union Bound when $P = 2$ and $N = 2, 3$. 26
2.10 System error probability, star topology, $P = 2$ 29
2.11 System error probability, butterfly topology 31
2.12 Achieved rate, butterfly topology 31
2.13 Probability to deliver IUs to a node 32
2.14 Average time to deliver IUs to a node 32
2.15 Probability to deliver IUs to all network 33
2.16 Average time to deliver IUs to all network 33
2.17 Reference topology. 35
2.18 Markov chain for modelling the protocols in correlated fading. 40
2.19 Throughput vs average SNR (Markov analysis) 42
2.20 Packet Delivery Ratio 44
2.21 Aggregate Throughput 45
2.22 System delay per acknowledged packet 45
2.23 Transmit energy consumption per information bit 46
2.24 Percentage of times that cooperative behavior is shown 46
2.25 Packet Delivery Ratio according to hop distance from the sink 48
2.26 Aggregate Throughput according to hop distance from the sink 49
2.27 Total aggregate Throughput, multihop scenario 49

2.28 Jain's fairness index, multihop scenario 50
2.29 Packet Delivery Ratio 50
2.30 Aggregate Throughput 51
2.31 Impact of Cooperation and Network Coding 52
2.32 Energy Consumption 53
2.33 Network Capacity in a 10 cells scenario 53

3.1 Pictorial representation of the estimation process as a function of the thresholds and the Frobenius norm of the noise 67
3.2 Mass Distribution function of the number of estimated users 68
3.3 Probability of missed detection for $k_M = 4$ 69
3.4 Probability of missed detection for $k_M = 3$ 70
3.5 Comparison of the missed detections for $k_M = 3$ and $k_M = 4$, for different values of the SNR 70
3.6 Throughput of the original protocol ($k_M = 4$), for different values of the SNR, 10 pairs 71
3.7 Throughput of the modified protocol ($k_M = 3$), for different values of the SNR, 10 pairs 72
3.8 Semi-Markov model describing the protocol 73
3.9 Comparison of power histogram vs model PDFs 76
3.10 Throughput for the mixture model vs ns2 simulations 80
3.11 Throughput for the mixture model vs ns2 simulations 80
3.12 Throughput for the single mode interference model 81
3.13 Optimal carrier sense density with respect to the node density, free space path loss model 85
3.14 Optimal carrier sense density with respect to the node density, two rays path loss model 85
3.15 Aggregate throughput, 30 nodes. Data Rate is 6 Mbps. 87
3.16 Aggregate throughput, 50 nodes. Data Rate is 6 Mbps. 87
3.17 Aggregate throughput, 30 nodes. Data Rate is 12 Mbps. 88
3.18 Aggregate throughput, 50 nodes. Data Rate is 12 Mbps. 88
3.19 MAC delay. Data Rate is 6 Mbps 89
3.20 MAC delay. Data Rate is 12 Mbps 90
3.21 Average Hop Count Length vs time in the optimized case 90
3.22 Aggregate throughput, 30 node network. Data Rate is 6 Mbps. 91
3.23 Aggregate throughput, 50 node network. Data Rate is 6 Mbps. 92
3.24 MAC delay for the adaptive algorithms. Data Rate is 6 Mbps 92
3.25 A sample regular (a) and random (b) tree topology 100
3.26 Performance comparison for the regular topologies, $\Lambda_0 = 2.5$ dB, $K = 2$ 102
3.27 Performance comparison for the random topologies, $\Lambda_0 = 2.5$ dB, $K = 2$, $\alpha = 0$ . . 103
3.28 Performance dependence on the decoding threshold, $K = 2$, $\alpha = 0$ 103

3.29 Dependence on the number of dominant interferers, for regular (above) and random (below) topologies, $\Lambda = 2.5$ dB, $\alpha = 0$ 104
3.30 Performance of our scheduler with web-browsing and CBR flows, $\alpha = 0.5$. 105
3.31 Performance of our scheduler for different values of the suppression factor α. 105

4.1 A two-hop random network. So stands for source, Re for relay and De for destination. Note that the relay is not necessarily placed in the middle between source and destination. 114
4.2 An N-hop route. 115
4.3 D vs. R, for $N = 1, 2, \ldots, 7$ hops and backlogged sources ($p = p_R = 1$). For each R, D is minimized over all relay placements. 123
4.4 NT vs. R, for $N = 1, 2, \ldots, 7$ hops and backlogged sources ($p = p_R = 1$). 123
4.5 D vs. R for optimized/non-optimized MAP and backlogged sources. 124
4.6 D vs. r_1 for backlogged sources ($N = 2$, $p = p_R = 1$, $R = 20$ m). 124
4.7 D vs. p, for different relay placements and backlogged sources ($p_R = 1$, $R = 30$ m). 125
4.8 D vs. r_1, for different percentages of the maximum allowable p_A ($N = 2$, $p = p_R = 1$, $R = 20$ m). 125
4.9 Optimum number of hops vs. R, for different percentages of the minimum unstable RT ($N = 2$, $p = p_R = 1$). 126
4.10 Comparison of theoretical and simulated packet success probabilities for a two-hop network and backlogged sources ($p_R = 1,\ r_1 = 0.6R$). 126
4.11 Comparison of theoretical and simulated packet success probabilities for a three-hop network and backlogged sources ($p = p_R = 1,\ r_1 = 0.45R,\ r_2 = 0.8R$). 127
4.12 Comparison of theoretical and simulated success probabilities for a two-hop network and non-backlogged sources ($p_A = 0.3$, $R = 20$ m). 127
4.13 Simulated packet success probabilities for a static two-hop network and backlogged sources ($R = 20$m, $r_1 = 12$m). The theoretical values correspond to the system model described in Section 4.2. 128

Abstract

This thesis collects the key research results on wireless networking developed during the three years of the PhD. The approach that underlies all work has been the analysis and design of wireless network protocols together with their physical layer. The former have been created keeping in mind the features of the adopted physical layer techniques and, conversely, the physical layer has been chosen according to the employed MAC/routing and its necessities. The results focus on three main areas.

In the first branch, MIMO signal processing is applied in order to optimize broadcast in a wireless MIMO network, to improve the robustness of Network Coding to the vagaries of the wireless environment and finally to design cooperative protocols that reward nodes which help each other. In all these cases, great emphasis has been placed on signal processing and on its actual algorithmic implementation.

In the second part, Carrier Sense optimization for radio networks is studied. First a special type of carrier sense system for MIMO ad hoc networks is described. Then, attention is focused on single antenna terminals, and an analytical model for carrier sense optimization in static networks (i.e., topology is known) is developed so as to find the carrier sense threshold that maximizes aggregate throughput. Moreover, specific algorithms have been created also for dynamic networks (i.e., nodes are mobile or the topology is not known beforehand). In addition, the analytical model for static networks is applied also to design a low-complexity, high-performance scheduler for mesh networks. In any case, interference analysis and the characteristics of the propagation environment play a key role to study this problem.

In the last portion of this thesis, a more theoretical approach is undertaken: the performance of an Aloha multihop wireless network in terms of throughput and delay is analyzed, under saturation conditions or not. The impact of a variety of physical layer parameters (like rate, path-loss exponent or SNR decoding threshold) on network performance is analyzed, and we establish how certain parameters (e.g., transmission probability, rate, route length, admission control) must be tuned in order to optimize performance.

Sommario

In questa tesi si riunisce il cuore della ricerca svolta nei tre anni del dottorato nell'ambito delle reti wireless. L'approccio che accomuna tutti i risultati è stato lo studio di protocolli per reti wireless in congiunzione al livello fisico. Si è cercato di progettare i primi tenendo conto delle peculiarità del livello fisico adottato e viceversa si è scelto il tipo di livello fisico in funzione del protocollo di livello MAC/routing impiegato. I risultati si concentrano in tre grosse aree tematiche.

Nel primo filone, si studia come il signal processing MIMO possa essere impiegato per ottimizzare la disseminazione dati in una rete MIMO, per migliorare la robustezza del Network Coding in un ambiente wireless e infine per progettare protocolli cooperativi che ricompensino l'aiuto reciproco fra nodi tipico di questo approccio. In tutti questi casi, si pone una grossa enfasi sull'elaborazione del segnale e sulla implementazione effettiva degli algoritmi.

Nella seconda sezione, si analizza come il meccanismo del Carrier Sense possa essere ottimizzato in reti radio. Si sono costruiti algoritmi per trovare la soglia di carrier sense che massimizza il throughput aggregato in reti statiche (ovvero, la topologia è fissa e nota) ma anche quando la topologia è dinamica o non è nota a priori. Inoltre, il modello d'interferenza sviluppato per reti statiche è utilizzato per produrre uno scheduler per reti mesh a bassa complessità computazionale e ad alte prestazioni. In ogni caso, l'analisi dell'interferenza e le caratteristiche dell'ambiente di propagazione assumono un ruolo centrale nello studio del problema.

Nell'ultima parte, si intraprende un approccio più teorico e si analizza in forma chiusa la performance di una rete wireless Aloha multihop in termini di throughput e ritardo, in saturazione o meno. Si analizza l'influenza di una varietà di parametri di livello fisico (come il rate, path-loss o l'SNR minimo di decodifica) sulle prestazioni della rete, stabilendo come debbano essere regolati certi parametri di progetto (per esempio, la probabilità che un nodo trasmetta in un certo slot, il rate, la lunghezza della rotta, l'admission control) di modo da ottimizzare le performance.

Chapter 1

Introduction

Presently, all practical wireless networks mimic many aspects of their wired counterparts. For instance, a common design guideline is to decouple layers as much as possible. The physical layer (PHY) is indeed the best example, since any wired networks essentially consider the physical layer a reliable bitpipe. Any problem with wired networks arise and are dealt with at the upper layers, for instance congestion, presence of different traffic classes or security. As far as physical and MAC layers are concerned, the ethernet and optical fibers are effectively regarded as *de facto* standards for local networks and long distance communications, respectively.

However, the special features of wireless networks have disputed the foundations of such a strictly layered approach and especially whether PHY and the rest of the protocol stack should ignore each other's peculiarities. For instance, the broadcast nature of the radio propagation environment implies that the same packet is simultaneously sent to and received by many terminals at the same time. This is in stark constrast with wired networks, where signals are confined by the cable/optical fiber. This broadcast property has been object of much attention. It is naturally useful for multicast traffic but can also provide spatial diversity. However, it is also the fundamental cause of interference, which can be rather detrimental if proper care is not exercised.

Much effort has been put in so called cross-layer design, i.e., the idea of having the layers interact with each other so as to optimize performance. Like every ground-breaking idea, it has its positive and challenging elements. Among the former, cross-layer optimization has questioned many assumptions in the wireless world and has led to a whole range of new protocols and design issues. In addition, important performance improvements have been proved, in some cases also in testbeds. The chief consideration to oppose cross-layer design is that coupling different layers implies to reduce modularity among different protocols.

Being aware of these two points, the main idea that has led the research during the PhD has been to deal with wireless networking by keeping in mind that physical layer and the upper layers cannot be oblivious of each other. This does not mean that protocols at different points in the stack must be jointly optimized. Instead we mean that the physical layer must be tuned according to the purpose of the message it is carrying. Or, on the other hand, the upper layers must be aware of the capabilities of

the underlying physical level. An important example has been MIMO networking. This field, which has emerged in the past 5-6 years, focuses on the design of MAC and routing protocols for networks with multiple antennas. The new degrees of freedom brought by this hardware and the huge capacity gains of MIMO cannot be harnessed in a wireless network without proper MAC protocols, since also from an information theoretic point of view, multiuser MIMO is indeed different from point to point MIMO, and any protocol that wants to achieve a meaningful share of the MIMO gains must bear this in mind. This has been recognized by many researchers and there have been efforts in this direction. For instance, a fundamental result in point-to-point MIMO is that if transmitter and receiver have N antennas each, the ergodic capacity is proportional to $NW\log(1+\Lambda)$, where W is the bandwidth and Λ is the Signal to Noise Ratio. This capacity can be achieved by real-world architectures (i.e., spatial multiplexing at the transmitter and MMSE V-BLAST at the receiver). However, some data streams must propagate through poor channels and therefore they will be the detection bottleneck. Instead, in a multiuser setting, it is possible to improve the channel matrix properties by multiuser diversity, e.g., it is possible to select the users to transmit to and hence the channel matrix. This leads to more benign propagation channels where no data stream/user has to face a channel with low SNR.

This thesis reports the results developed in the three main areas that have been explored during the PhD. All these three areas have kept in mind the above mindframe, as the rest of the thesis will show:

- In the first area, the usage of MIMO signal processing for MAC protocols has been studied. In some cases, nodes may have multiple antennas, but in other cases they may not. We show that this type of physical layer techniques can be useful in a variety of environments. For instance, to efficiently distribute control packets, to improve the performance of Network Coding and finally to design innovative cooperative protocols. Work in this field is reported in Chapter 2.

- Carrier sense is the focus of the second part (Chapter 3). This technique is the foundation for all Carrier Sense Multiple Access (CSMA) MAC protocols. We have studied how this mechanism can be tuned, taking into considerations the properties of the wireless medium. Moreover, also carrier sense for MIMO networks is analyzed and discussed, with a specific proposal for these systems. Finally, an effective scheduler for mesh networks based on our algorithms is proposed.

- While the previous sections focus on algorithms and practical design of protocols, the last part (Chapter 4) analyzes the performance of a generic wireless network by means of tools taken from stochastic geometry. Again, physical layer does play its part, since the underlying model takes into account the properties of radio propagation and interference. The importance of this Chapter lies in its generality and in the design insight that can be gained by the simple analytical formulae that can be derived.

The thesis ends with Chapter 5, which draws the conclusions of this research project, and the list of all published and submitted papers to date (Appendix A).

In conclusion, this document deals with a variety of topics (MIMO ad hoc networks, mesh net-

works, network coding, carrier sense, stochastic geometry), but in all cases special attention has been devoted to the interactions between physical layer and MAC/routing.

Chapter 2

MIMO signal processing for ad hoc network MAC protocols

Contents

2.1 Introduction . . . 6
2.2 A low-delay MAC solution for MIMO Ad Hoc Networks . . . 7
2.2.1 Antenna and Channel Models . . . 9
2.2.2 Protocol Model . . . 9
2.2.3 Performance Evaluation . . . 13
2.2.4 Discussion . . . 17
2.3 Network Coding meets MIMO: A look into the rate/diversity tradeoff in wireless Network Coding . . . 17
2.3.1 The MIMO_NC schemes . . . 18
2.3.2 Performance Analysis . . . 22
2.3.3 Performance Evaluation . . . 28
2.3.4 Discussion . . . 33
2.4 Phoenix: A Hybrid Cooperative-Network Coding Protocol for Fast Failure Recovery in Ad Hoc Networks . . . 34
2.4.1 Phoenix: A Cooperative - NC Protocol . . . 35
2.4.2 Analytical Model . . . 40
2.4.3 Performance Evaluation . . . 42
2.4.4 Discussion . . . 54
2.5 Chapter Conclusions . . . 54
2.6 Acknowledgments . . . 54
References . . . 55

2.1 Introduction

Since a decade ago, the discovery of the huge capacity gains achievable by means of multiple antennas has spurred an unprecedented activity to harness the degrees of freedom in these systems [1–5]. So far, the attention devoted to point-to-point multiple-input multiple-output (MIMO) links has been extraordinary (see, for example, the tutorial paper [6] and the book [3] for an introduction on the matter). Multiple antennas can provide spatial diversity (by means of space time coding at the transmitter and signal combining at the receiver), power gain (if the transmitter or the receiver have some kind of channel state information), interference suppression (by means of directional transmissions) and finally spatial multiplexing (i.e., the ability to transmit multiple data streams at the same time). If N is the number of antennas at each node, it is possible to achieve an N-fold capacity increase, a power gain of $20\log_{10}(N)$ dB or a diversity order of N^2. These gains cannot be simultaneously achieved, as [7] has argued.

However, not so much attention has been dedicated to the interaction between MIMO and the upper layers [8]. On the one hand, the studies for the cellular case have achieved a relatively mature status [9–11], but on the other hand work on MIMO ad hoc networks is still incomplete. While some important efforts have been devoted to this area [11, 12, 14], there is not yet wide consensus on how to use these degrees of freedom in an interference limited environment without infrastructure like ad hoc networks.

One of the most important problems is the definition of the MAC protocol. Since the Distributed Coordination Function of the IEEE 802.11 is the *de-facto* standard MAC protocol for ad hoc networks [15], it has been taken as the basis for most MAC protocols in the context of MIMO ad hoc networks. This has been the case for terminals with directional antennas [16–21] but also for MIMO ad hoc networks. However, some studies [10–12] challenge this view and propose other schemes, more reliant on scheduling rather than carrier sense. Incidentally, work on routing, transport and upper layers for MIMO networks is still scanty [14, 23, 24].

A common problem to many papers (especially those on directional antennas) is the usage of some very simplified propagation and antenna models, e.g., cone–sphere models, with the main antenna lobe having constant amplitude over a certain angular extension, with negligible or even no side lobes. This may not be always the case, especially when performing array processing, i.e., when obtaining directivity through the superposition of omnidirectional signals sent from the array elements. Even if some beamforming technique is available that can steer a main beam of predefined width, and still guarantee secondary lobes to be under a given threshold [25], they may still radiate a significant amount of power, potentially reducing the accuracy of simplified models. A study by Takai *et al.* [26] has also highlighted the need to account for realistic physical layer models when evaluating ad hoc networks with multiple antennas.

The first part of this chapter deals with a modification of the IEEE 802.11 for MIMO ad hoc networks. In this section, it is investigated which physical layer can be most beneficial for broadcast packets. The goal is to distribute these vital packets as far as possible, but delay must be kept as short as possible at the same time. In addition, also unicast handshake packets (like RTS/CTS) fit into this

description, since they should be received by as many nodes as possible in the neighborhood of the transmitter. While the initial part of this work was developed during the Master Thesis, the second part (which evaluates the network performance) has been carried out during the PhD, and hence is reported here.

In the second part, we design a new physical layer for Network Coding [27], called MIMO_NC, which exploits MIMO signal processing to make Network Coding more robust to the heavy packet losses of the wireless propagation channel. Even though nodes are assumed to have just a single antenna, MIMO signal processing is fundamental, since MIMO_NC employs a parallel between Network Coding and MIMO to overcome the fragility of the former to packet losses. This scheme is tested in multiple access, multihop networks and compared against standard network coding.

In the third and final portion, MIMO_NC is used to build a highly efficient cooperative protocol. A key problem of cooperative protocols is the necessity of having a node (the relay) that devotes its own resources to help another node, without having a direct benefit. MIMO_NC is the key building block to solve this issue, since it allows the relay to transmit its own data along with the retransmitted packet. Such a feature is helpful in single hop networks (because it encourages cooperation) but it is even more important in multihop networks: the ability to recover a packet loss without wasting resources for ARQ can effectively reduce the congestion at bottlenecks.

2.2 A low-delay MAC solution for MIMO Ad Hoc Networks

Beamforming [28] has been the main ingredient of many multiantenna ad hoc network MAC protocols [12, 16, 18, 21, 29, 30]. Unicast data packets especially benefit from its ability to increase coverage and reduce interference. On the other hand, control packets (like RTS/CTS frames) or multicast/broadcast transmissions (such as routing information) need to be distributed omnidirectionally and with increased range, in order to reach as many neighbors as possible. If these packets are transmitted directionally, coordination between nodes can become very poor [21], [31]. This asymmetry between the possibility to achieve a high (but directional) gain and the need for an omnidirectional transmission has been called the gain asymmetry and, if not properly addressed, may lead to poor performance at the MAC layer, where directional transmission of data packets and omnidirectional transmission of control packets coexist [18,21,29,30]. One of the main ideas to solve this issue so far has been the concept of Circular RTS (C-RTS), [1] where a control packet is successively beamformed in adjacent sectors, so that the whole horizon is swept by means of multiple transmissions. This approach extends the transmission range while also retaining omnidirectionality, but comes at the price of additional delay. A longer handshake increases the contention time, thus reducing the efficiency of the handshake itself and the overall performance. Finally, the usage of multiple packets increases interference and energy consumption.

Moreover, a closely related problem is that the majority of the proposed MAC protocols for

[1] The technique of C-RTS was originally proposed in [18] and later extended (under the name of Circular RTS and CTS MAC, CRCM) in [30]. While our comparisons refer to the latter and more advanced version, we will use both terms (CRCM and C-RTS) interchangeably.

MANET with smart antennas [16, 18, 21, 31], as well as the studies of broadcast techniques [32, 33], assume that omnidirectional reception must have unit gain, so that the transmitter-receiver pair is effectively turned into a MISO (rather than MIMO) system. This is due to the erroneous belief that there is always a trade-off between beamforming gain and beamwidth. While this is certainly true at the *transmitter*, where sending on two beams simultaneously requires twice the power and is therefore impossible in the presence of a power constraint, at the *receiver* this is not the case, because starting from the baseband samples at the output of the antennas it is possible to form as many receiving beams as desired by just using signal processing (i.e., no additional power is needed to implement more beams, as long as the processing power consumption can be considered as negligible). This misinterpretation of the beamforming process at the receiver has led some authors to designing schemes that take into account a constraint that does not exist.

While the asymmetry in gain has been identified and addressed for directional antennas [18, 21, 29, 30], protocols designed for MIMO networks do not deal with it [12, 34, 35]; these papers explore the interaction between MIMO PHY and MAC, and they assume that the coverage of directional and omnidirectional communications is the same, because the network is geographically small. Or in some other cases [35] unicast and broadcast packets are transmitted by the same space time architectures (STBC in that case). This approach is suboptimal, because it does not consider the possible directionality inherent in antenna arrays even when it could be useful. On the other hand, studies for broadcast in multiantenna networks [32, 33] focus on designing MAC protocols that exploit directivity, but again directional and omnidirectional transmission ranges are the same, because the power for directional communications is lowered.

The problem that we want to solve is to employ a known MIMO technique for packets that have to be processed by all neighbors, such that it 1) achieves the same performance as [18] in terms of increased coverage, 2) provides a major reduction of the delay (sending the packet as few times as possible), 3) is an open loop technique, and 4) is not applicable to directional data transmissions. The last condition can be motivated as follows. A solution suitable for transmission of both control (omnidirectional) and data (directional), such as for example traditional channel coding or TCM, would benefit them equally and hence would not reduce the asymmetry. Therefore our goal has been to design an efficient protocol component by means of existing PHY techniques.

The contribution of this part is a novel control packet exchange method that provides extended as well as omnidirectional coverage while not suffering from the long delays incurred by C-RTS, and the design and performance evaluation of a new MAC protocol based on it. Our goal is not to propose a new MIMO technique, but rather to use existing multiantenna schemes in the design of high-performance protocols which exploit the opportunities offered by antenna arrays. Our scheme includes for the first time some MIMO techniques that, while well established in the PHY community, have not yet been considered in the *protocol design* for MIMO ad hoc networks [12, 34, 35]. We show that some existing and well studied MIMO techniques are sufficient to overcome the asymmetry in gain. The upshot is a 40% delay reduction and 15% throughput improvement over state-of-the-art protocols.

2.2.1 Antenna and Channel Models

In this Section, plain letters (i.e., x) denote scalars, overlines (i.e., $\overline{x}$) denote column vectors and capital bold letters with an overline (i.e., $\overline{\mathbf{X}}$) denote matrices.

The MIMO channel is regarded as Ricean flat fading, slowly varying in time. The baseband (complex) channel gain from the α-th input to the β-th output is denoted by $h_{\alpha,\beta}, \alpha, \beta \in \{1, 2, ..., N\}$, where N is the number of antennas at each node. We consider here a Space-Time Block Code (STBC) model [4], in which a block of K complex data symbols (arranged in the vector $\overline{x} = (x_1, x_2, ..., x_K)^T$, $x_i = x_i^r + jx_i^i$) is transmitted in $L \geq K$ symbol intervals (K/L is the code rate). In the ℓ-th interval of each L-symbol group, the N antennas send linear combinations of the K data symbols, computed according to two weighing $N \times K$ matrices: $\overline{\mathbf{W}}_\ell^r$ for the real part and $\overline{\mathbf{W}}_\ell^i$ for the imaginary part. The output of the baseband equivalent of the receiver antennas in the ℓ-th symbol interval (neglecting interference from other users) can be written as:

$$\overline{r}(t) = \overline{\mathbf{H}} \left(\overline{\mathbf{W}}_\ell^r \overline{x}^r + j \overline{\mathbf{W}}_\ell^i \overline{x}^i \right) + \overline{n}(t) \tag{2.1}$$

where $\overline{n}(t)$ is a zero-mean, spatially and temporally white Gaussian noise. A set of complex matrices $\overline{\mathbf{W}}_\ell^r, \overline{\mathbf{W}}_\ell^i, \ell = 1, \ldots, L$ defines an STBC. For instance, for the Alamouti code [5] and QAM modulation, two data symbols are transmitted every two symbol intervals, so that $K = L = 2$ and $\overline{\mathbf{W}}_1^r = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \overline{\mathbf{W}}_2^r = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \overline{\mathbf{W}}_1^i = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \overline{\mathbf{W}}_2^i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

Finally, the Ricean statistics describes each channel gain by a constant coefficient and a complex circularly symmetric Gaussian random variable:

$$\overline{\mathbf{H}} = \sqrt{\frac{\kappa}{\kappa+1}} \overline{\mathbf{H}}_{LoS} + \sqrt{\frac{1}{\kappa+1}} \overline{\mathbf{H}}_w = \sqrt{\frac{\kappa}{\kappa+1}} G \overline{a}(\theta_t) \overline{a}(\theta_r)^T + \sqrt{\frac{1}{\kappa+1}} \overline{\mathbf{H}}_w \tag{2.2}$$

where κ is the Rice parameter, and θ_t and θ_r are respectively the direction of the receiver with respect to the line perpendicular to the transmitter's array and viceversa. The first part is the LOS component. The coefficients $\overline{a}(\theta_t)$ and $\overline{a}(\theta_r)$ can be derived by geometric reasoning ([28], [9], Section 2.2.2) and depend only on the array geometry and the transmission/reception directions. $\overline{\mathbf{H}}_w$ is a matrix composed by complex circularly symmetric Gaussian random variables, and models fading.

We assume that the receiver has perfect Channel State Information (CSI, e.g., obtained by training based estimation), while the transmitter only knows the function $\overline{a}(\theta_t)$ (which depends only on θ_t and requires no feedback from any receiver). This asymmetry is due to the fact that in multicast it is typically impossible for the transmitter to have CSI for all the receivers (whose number and identity may even be unknown).

2.2.2 Protocol Model

In the following we outline our protocol and a modified CRCM that is more robust to fading.

STBC based broadcast

In our previous research [1], we showed that (in agreement with other results from array processing theory [28] or information theory [1]) in a MISO setting the asymmetry in gain cannot be eliminated, because there is an intrinsic tradeoff between coverage and packet delivery delay. Thus the receiver's degrees of freedom must also be used in order to overcome this impairment. The performance gap with respect to CRCM can be bridged using Maximum Ratio Combining (MRC) at the receiver and a space time block code [4] with full rate (to keep the delay as short as possible) and full diversity (to improve the outage behavior of the scheme). These features can be found in the STBC proposed in [37] (called ABBA), refined by constellation rotations [38], which also provides some coding gain. The decoding is performed by a twice iterated MMSE-PIC. That is to say, all symbols are first decoded by an MMSE filter, then each stream is decoded after the contribution from the other symbols is cancelled. The computational complexity is about twice as large as that of linear MMSE detection, but the performance in terms of BER vs SNR is close to optimal. We opted for an already existing coding strategy, because our purpose is to use known PHY solutions to enhance a MAC protocol, rather than to advance space time processing theory itself.

Therefore, our proposal is to employ this class of STBC to transmit control packets; we prove by Bit Error Rate (BER) simulations (reported at the end of this Section) that a suitable MIMO technique (the ABBA code properly enhanced) can provide the same performance as the beamforming technique of [30] in terms of range extension and omnidirectionality, without the delay and energy costs associated to multiple transmissions. Thus, a brief description of the protocol is as follows:

- Medium access is controlled by conventional carrier sense;
- RTS/CTS packets are transmitted by means of the ABBA STBC. The CTS includes the channel right eigenvector corresponding to the largest singular value;
- Unicast packets (data/ACK) are sent by closed loop beamforming. The weight vector was included in the CTS;
- If no feedback is received (i.e, CTS or ACK), another attempt is performed after a binary exponential backoff, or the packet is discarded if the maximum number of retransmissions has been reached.

Unicast packets may employ spatial multiplexing or beamforming. While an application of the former is, for example, [34], we focus on the latter. In conclusion, we study CSMA/CA where control packets use the ABBA STBC, and evaluate the network performance benefits provided by this space time architecture.

Modified CRCM

The original CRCM [30] would follow the same protocol model outlined in the previous subsection, except for the transmission scheme for RTS/CTS packets. In [30] an RTS/CTS frame would be

sent by a sweep of directional communications in order to emulate a broadcast transmission, while the Data/ACK exchange is directional. The radiation pattern adopted in studying CRCM followed the pie-slice model, and selection diversity was assumed at the receiver [18]. While this might be quite realistic for LOS conditions, the presence of a highly scattering medium would break the regularity of the pattern, so that CRCM would present unacceptable performance. Instead, one of the desirable features of our system is that it offers robust performance with respect to the degree of fading. Therefore it is a suitable choice for both LOS environments and Rayleigh fading conditions. [2] Therefore, for a fair comparison, we have modified the scheme in [30] by the inclusion of an STBC. [3] In addition, the required method must be flexible enough to reduce to a conventional beamforming scheme if CSI is available at the transmitter. The scheme proposed in [40] associates the Alamouti STBC [5] with equal gain combining in a frequency flat environment, and has been chosen because of its simplicity and effectiveness. The difference between the modified CRCM and our scheme lies in the type of STBC and in the number of transmissions for each RTS. In its original form, the scheme is designed for a system with 2 transmit antennas, but it can be readily generalized to encompass any number N of antennas. In this type of STBC, the transmitter is assumed to know the phase difference between the channel coefficients, so as to perform equal gain combining beamforming. In [40] this quantity is fed back by the receiver. However, in a broadcast environment this kind of information is not available, thus the transmitter simply assumes the channel vector $(h_1\, h_2\, ...\, h_N)$ towards direction θ_t (with respect to the direction orthogonal to the array axis) to be proportional to $(\hat{h}_1\, \hat{h}_2\, ...\, \hat{h}_N) = (1,\, e^{j\phi},\, e^{2j\phi}, ..., e^{j(N-1)\phi})^T$ where $\phi = \pi d\ sin\theta/\ell$, which are the coefficients of classic linear beamforming schemes [28] (d is the array pitch, ℓ is the carrier wavelength, θ is the actual direction of the receiver with respect to the transmitter). The choice of θ_t is exactly the same as in [18]: $2\pi k/M$, where M is the total number of control packets needed to sweep the whole horizon and k is the number of control packets sent so far in the sweep. Thus also this scheme is open-loop. Since the ABBA code does not require any feedback from the receiver, both systems are open-loop.

Moreover, the original scheme in [40] includes a parameter, λ, which is a distribution coefficient for allocating power between the two transmitted symbols. The choice of this parameter should mirror the correlation between the estimated and the actual coefficients. In our case, the closer the scenario to the LOS condition, the higher the correlation. Since the power of the LOS component is proportional to $\kappa/(\kappa+1)$, λ is taken to be equal to $1-\kappa/(\kappa+1)$. Provided that the Rice constant is known, [4] the beamforming matrix can be built as follows:

$$\begin{pmatrix} 1 & 0 \\ v\sqrt{1-\lambda} & \sqrt{\lambda} \end{pmatrix}$$

where v is equal to $e^{-j\cdot arg(h_2 h_1^*)}$, that is to say the phase difference between the first and the second

[2] Most of the cited protocols work in LOS conditions only. However, there are some protocols designed for highly scattering conditions, e.g., [12]- [39]

[3] We remark that our previous work [1] simulated C-RTS by a highly idealized model, where the transmitter had perfect CSI toward any receiver. The modified version here is a more realistic implementation of C-RTS for a fading channel.

[4] It may be estimated by averaging the channel coefficients in time and space, because the channel statistics is assumed to be time invariant in both dimensions.

columns of the channel matrix. In a broadcast scenario, this (unknown) information is replaced by $e^{-j \cdot arg(\hat{h}_2 \hat{h}_1^*)}$. If perfect CSI were available at the receiver, this scheme would yield an equal gain combining beamforming. Therefore, for a circular broadcast of RTS/CTS packets, each time a certain direction is taken as reference and the channel coefficients are estimated according to that bearing.

For $N > 2$ antennas at the transmitter, the system can be generalized as follows. The STBC is still Alamouti, but each symbol is sent by a block of $N/2$ contiguous antennas. Finally, this system could be extended to include a higher diversity STBC, but it would lose orthogonality and that would entail more complicated signal processing.

A brief comparison of the BERs achieved by the two systems is reported in Fig. 2.1 when both transmitter and receiver have 4 antennas. Fig. 2.1 shows that the usage of the ABBA STBC with MRC does provide enough coding gain to overcome the asymmetry in gain. The powerful ABBA code yields a very robust behavior with respect to different channel variabilities (the two curves for STBC basically overlap). Even in the AWGN case (the scenario CRCM is designed for) STBC outperforms CRTS. The reason is as follows: the performance would be very close if the receiver were always at the center of the main beam. However, this may not be the case, since its position is random inside the beam. Therefore, the C-RTS may not always provide as high a gain as possible, and this causes the 1 dB performance loss observed in Fig. 2.1 for the AWGN case. The STBC does not rely on any LOS component or directionality, and this offers a smooth performance with respect to the transmitter-receiver mutual position.[5] These BER simulations show that the STBC method achieves at least the same coverage as CRCM for the shown array sizes, since for the same SNR $\text{BER}_{STBC} \leq \text{BER}_{CRCM}$. For even larger array sizes (10 antennas or more) the beamforming array gain cannot be provided by modified STBC. In this case, the packet has to be transmitted twice or more to match the power gain. However, these large arrays are not of practical importance for ad hoc networks. In our setting, a single transmission is enough to provide as good a coverage as CRCM.

In conclusion, the main differences between our scheme and CRCM are:

- Our scheme uses STBC ABBA, CRCM employs beamforming (hybrid STBC/beamforming in its modified version);
- Our scheme needs to transmit a packet only once for moderate array sizes N, CRCM needs at least N transmissions;
- Our scheme is completely open loop, CRCM needs to estimate the Rice constant.
- Our scheme uses the ABBA STBC, while CRCM employs beamforming (or hybrid STBC/ beamforming in its more advanced version) for handshake packets;
- Our scheme manages to broadcast a packet in a single transmission for moderate array sizes, CRCM needs at least N transmissions;

[5] The value of about 1 dB may be predicted by computing the average gain of a 4 antenna linear uniform array in broadside configuration inside its 3 dB beamwidth. This integral mean is equal to 3.24, which corresponds to a loss of 0.91 dB with respect to the peak value (equal to 4).

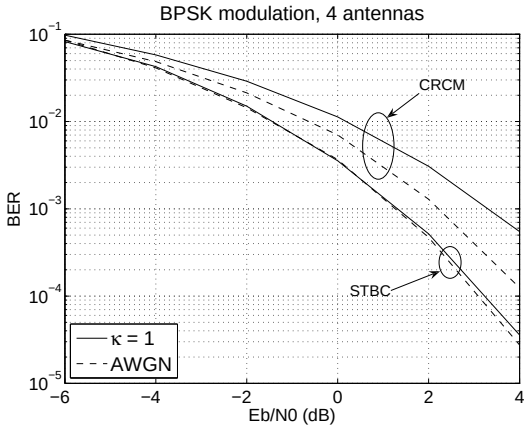

Figure 2.1. *BER Comparison for CRCM and STBC*

- Our scheme is completely open loop, modified CRCM needs to estimate the Ricean constant.

2.2.3 Performance Evaluation

The two schemes in Section III can be employed in modified IEEE 802.11 MAC protocols, where the control packets are sent according to one of these two methods and the data packets by directional beamforming. In order to assess their actual impact on network behavior and performance, CRCM and our protocol have been compared using OPNET 11.5. The network is a 200m $\times$ 200m square, where 12 nodes are uniformly randomly placed. The simulation time is 240 seconds, long enough to stabilize the metrics, and the results are averaged over 30 independently generated random topologies, which provide the desired statistical confidence.

The maximum transmission power (0.25 mW) has been chosen large enough to let every node transmit a data packet (8336 bits) to any other node in an AWGN channel with an outage probability of 10% only if they both transmit and receive directionally with 4 antennas each. [6]

Therefore, this is a single-hop network with no hidden nodes. This scenario, where all nodes may simultaneously contend for the channel, is designed to test the ability of the protocol to exploit spatial re-use and to reduce channel contention. MIMO ad hoc networks can provide increased parallelism, but the physical layer capacity improvement must be coupled with an adequate degree of coordination, or otherwise performance may be even worse than conventional 802.11 [31]. Our study has focused on the use of STBC to improve channel access. The impact of these techniques for multihop networks is an interesting problem.

There are four types of packets: RTS, CTS, DATA and ACK. Their sizes are, respectively, equal to 240, 240, 8336 and 120 bits per packet. The CTS is as large as the RTS because it carries the estimated coefficients of the transmit beamforming vector. In CRCM, each RTS/CTS is sent 5 or 10 times, for a 4 or 8 antenna array [1]. The packet arrivals are described by a Poisson process, whose rate takes values between 10 and 100 packets/s. This rate has been increased until saturation.

[6] Our simulations show that both approaches make the network fully connected. This indirectly proves that they achieve the same coverage, since this network is single hop only if sufficient array gain can be obtained also for handshake packets.

Each node is equipped with a linear antenna array, comprising 4 or 8 antennas. The Rice factor κ is equal to 1 or ∞ (AWGN). These two values have been chosen to test almost opposite environments, i.e., nearly Rayleigh or LOS. For our system, the rotation angle of the STBC has been chosen to be half the characteristic angle of the PSK modulation. While this is not always optimal, the performance is often close to the maximum [38]. Moreover, the channel is never purely Rayleigh. If fading followed a Rayleigh statistics, CRCM performance would drop to a very low level, because no LOS component is present, while CRCM needs some predictable LOS component to beamform and achieve array gain. We have avoided this scenario, in which CRCM would be too penalized, and haven chosen $\kappa = 1$ instead as representative of a heavily faded channel. Finally, the channel bandwidth is 1 MHz and the modulation is BPSK, and thus the data rate is equal to 1 Mbps.

Three metrics (aggregate throughput, packet delivery latency and transmit power consumption) will be evaluated. The first metric is the aggregate throughput, defined as the total number of data bits successfully acknowledged, normalized by the product of the channel bandwidth and the simulation duration. This quantity is depicted against the aggregate load,[7] which is computed as the total number of transmitted bits, including both data and control packets, also normalized by the bandwidth-duration product. Whenever this quantity is larger than 100%, then spatial reuse is effectively in place, because more bits than the duration-bandwidth product are sent. Hence this load (and not the nominal load) actually proves whether antenna arrays enable spatial reuse, and is thus more informative for this class of MACs. The results are reported in Figs. 2.2 and 2.3. A few observations are in order. First of all, Figs. 2.2 and 2.3 analyze the same networks; in the former, the Rice constant is 1, in the latter ∞. For a scenario close to Rayleigh fading, our system outperforms CRCM for both array sizes (gaining 15% and 25% in throughput, respectively). In addition, the performance of CRCM actually becomes worse for a larger number of antennas (losing 5% in throughput), because of the increased RTS/CTS overhead and interference. In such a case, handshakes fail more often and since their duration is longer the loss in throughput is more noticeable. The phenomenon is particularly remarkable in the AWGN case: in the 4 antenna case CRCM delivers the same performance as STBC but with slightly smaller load. However, in the 8 antenna case the relative order is reversed, with a performance gain as large as 25%, showing that our scheme can better exploit the benefits of antenna arrays. With 4 antennas, CRCM slightly outperforms our protocol because each hadshake packet is sent out 4 times, and thus these frames can enjoy a limited amount of time diversity. For a small number of antennas and the AWGN channel, CRCM does not incur a large overhead and its RTS/CTS have additional robustness due to the repeated retransmissions. However, CRCM performs better than the STBC solution only in terms of throughput and only for LOS channels, that is to say the situation it has been designed for. On the other hand, our approach is more robust to the channel environment parameters, since it is not greatly affected by the specific value of the Rice constant. Incidentally, we point out that the metric shown on the x-axis of the plots is the effective load, and therefore the expected saturation effect is not shown in the graphs, because for higher nominal loads

[7] We shall use the expression effective load as a synonym. This term is meant to also highlight that we take into consideration the bandwidth actually used by the users, not the nominal load given by the traffic generated at the packet sources.

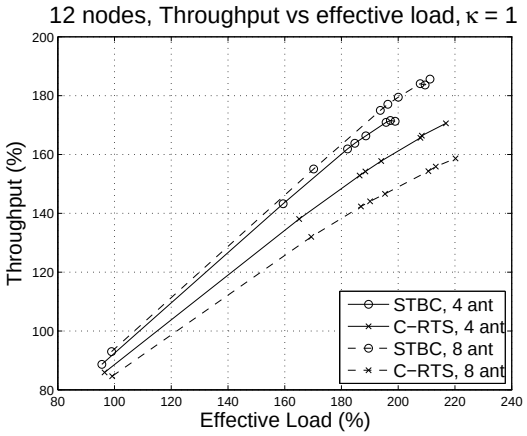

Figure 2.2. *Throughput vs effective load, 12 nodes*

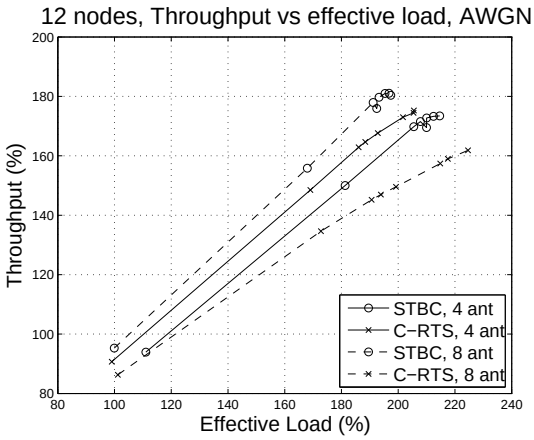

Figure 2.3. *Throughput vs effective load, 12 nodes*

both aggregate load and throughput decrease.[8] Therefore, points in the graphs reach a maximum in the top-right corner and then both coordinates scale down.

More importantly, thanks to the much more efficient RTS/CTS exchange, our scheme achieves a significantly reduced packet delivery delay (Figs. 2.4 and 2.5). In the comparison, we considered the latency between the start of MAC contention and the correct reception of the ACK. The delay reduction is between 17 and 20% for 4 antennas and between 32 and 37% for 8 antennas. In addition, this advantage is more noticeable for a larger number of antennas, as expected, and in the AWGN case the 8-antenna system outperforms the 4-antenna network because of the increased interference suppression, which leads to fewer retransmissions. Instead, CRCM's latency is affected by the circular RTS delay. Doubling the array size from 4 to 8 antennas results in twice as many transmissions of RTS and CTS packets (from 5 to 10 times each). Each of them takes 240 μs, leading to an additional delay of 2.4 ms. In fact, the CRCM curves in Fig. 2.4 are approximately the same but shifted upwards by about 2.8 ms, which is in fair agreement with this estimate. The difference is due to a worse

[8]We note that when the generated traffic is large, collisions become more likely and nodes spend a significant fraction of their time in backoff. This fact reduces the available time for transmission, and thus the effective load.

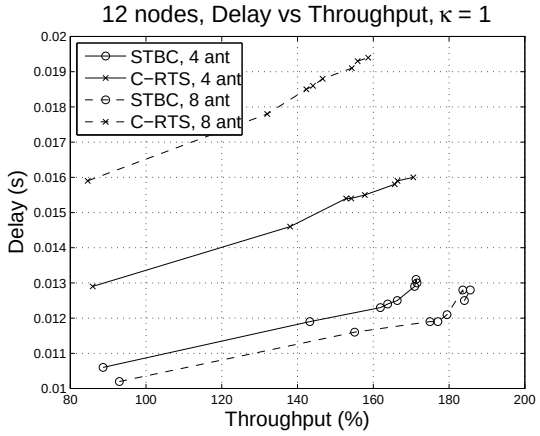

Figure 2.4. *Delay vs Throughput, 12 nodes*

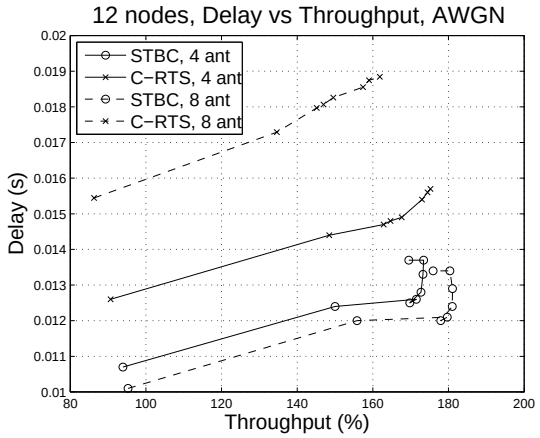

Figure 2.5. *Delay vs Throughput, 12 nodes*

protocol efficiency, as the longer handshake time slightly exacerbates the problems outlined before (such as increased collisions). In conclusion, the STBC based packet distribution takes advantage of the greater number of antennas in any propagation environment, which is not the case for CRCM; moreover the shorter handshake eases problems of coordination, whereas the opposite happens in CRCM.

Finally, another important metric is transmit energy consumption, computed as the total transmit energy divided by the number of information bits successfully acknowledged, and plotted against throughput in Fig. 2.6. We assumed that handshake packets are sent at the maximum power, while unicast frames (data and ACK) are subject to power control. This fact is essential because otherwise a data exchange (which lasts for a long time with respect to a control packet) would create a great deal of interference and would also capture many receivers in the area, preventing further communications. Therefore unicast frames have a limited weight in this metric, and the control (broadcast) packets contribute to the majority of the energy consumption. Since our scheme needs significantly fewer transmissions for a control packet, savings can be very high, as Fig. 2.6 shows. In addition, the increased number of collisions in CRTS makes the gap between the two sets of curves even wider.

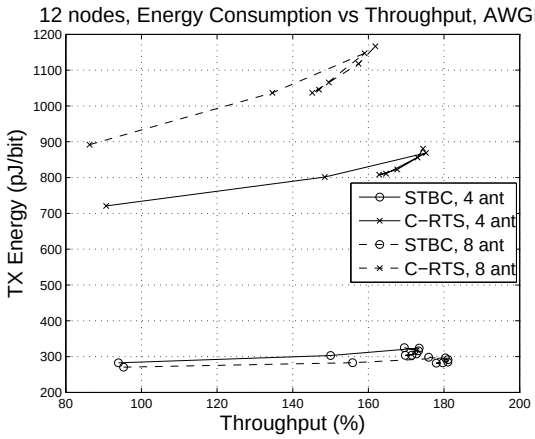

Figure 2.6. *Energy per information bit vs Throughput, 12 nodes*

2.2.4 Discussion

The issue of the gain asymmetry in ad hoc networks has been described and some of its consequences have been discussed. The solution proposed in [18, 30] has been reviewed. The major problem of this technique is the significant delay and energy consumption due to the circular distribution of the control packets. Building on theoretical considerations, in this work we have applied Space Time Block Codes combined with Maximal Ratio Combining for the transmission of handshake packets, and have shown that a suitable STBC can provide the required performance in terms of coverage and omnidirectionality without any delay penalty. The performance benefits when this scheme is incorporated into a MAC protocol for MIMO ad hoc networks have been documented by simulation results. Interesting extensions of this work include rate adaptation and multihop.

2.3 Network Coding meets MIMO: A look into the rate/diversity tradeoff in wireless Network Coding

Network Coding (NC) has been proposed as a throughput efficient system for data dissemination [27]. Since then, extensive theoretical work [**?**, 41–46] and practical implementations [47–51] have proved its effectiveness for real-world networks. NC delivers unquestionable benefits in wired networks, where the packet error probabilities are generally very low. However, wireless networks impose the far more hostile radio propagation environment. In this setting, NC may lose its efficiency, because a higher redundancy may be required to deliver the data, thus reducing the advantage over standard routing [?]. Moreover, NC is known to exhibit a threshold-like behavior with respect to packet losses: if P information bearing packets (called Information Units (IUs)) are coded together, P linearly independent combinations must be correctly received. Otherwise, only if early decoding is possible can some of the IUs be retrieved. On the other hand, conventional routing may enable to decode some of the IUs, thus reducing the latency for some packets.

In order to fight the erratic behavior of the wireless medium, diversity is often employed. MIMO

systems are extremely effective in achieving this goal [52]. The key mechanism of MIMO is to have the same information transmitted and/or received at different antennas, i.e., different locations in space. However, NC can itself be seen as an inherently multiple input, multiple output system, and an Information Unit is naturally present in most (if not all) the received Coded Packets. Thus a natural question arises: is it possible to take advantage of this diversity? Is there any way to have Network Coding exploit such an inherent redundancy to improve the decoding process? The idea comes from the observation that MIMO detection techniques and NC are based on a similar description of the system. In both cases, the transmitted data are the solution of a linear system $Ax = b$, where b contains the received data or samples and A is the channel matrix $\mathcal{H}$ in MIMO and the coding matrix G in network coding. This similarity is useful to develop an integrated system where MIMO signal processing and NC coexist at the same layer [4]. The key goals of this research effort are:

1. to propose a family of practical, computationally efficient schemes (collectively called MIMO_NC) based on NC that can tradeoff rate for diversity,

2. to show how much diversity can be recovered from Network Coding,

3. to gain some understanding on the tradeoffs between throughput and diversity in NC.

Recently, the interaction between network coding and other important communication techniques (especially channel coding and cooperation) has received some interest. The idea of moving network coding towards the physical layer has been first proposed in [54]. Relations between network coding and cooperation have also been investigated [55–58]. In [55], the authors propose a way to implement a physical network coding scheme over a two–way relay channel. Instead [56, 57] propose two approaches that exploit NC during a relaying phase, while [58] explores the rate regions for some NC/cooperative strategies.

Some research [59–63] deals with the integration of channel coding and network coding. These studies exploit the efficiency of network coding in order to both increase the performance of channel coding and reduce the channel decoding complexity. All these papers exploit in a smart way the diversity that joint channel coding and network coding can offer. However, both network coding and channel coding are applied across nodes and provide redundancy spread through the network, and thus it is not straightforward to understand which technique is eventually responsible for these performance improvements. Hence our aim is different, namely to analyze the performance of network coding alone, without the aid of channel coding, in order to find the fundamental performance limits in network coding *per se*. Moreover, many of these papers are focused on some specific scenarios (e.g., the relay-aided delivery of some packets), while our techniques can be applied in a generic network.

2.3.1 The MIMO_NC schemes

The complete flow of MIMO_NC is represented in Fig. 2.7. We want to point out that although all nodes are equipped with a single antenna, this system is MIMO because multiple inputs (the information packets) are coded together by a NC matrix G to create the multiple outputs (the coded

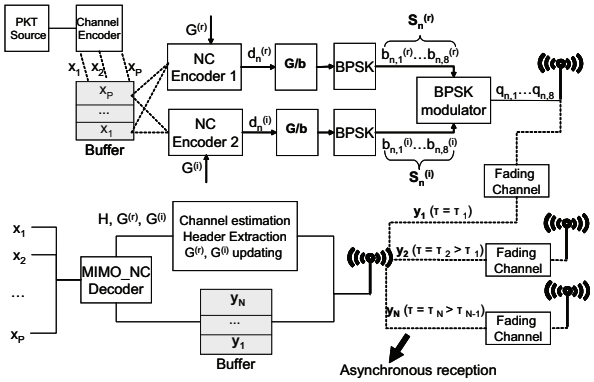

Figure 2.7. *MIMO_NC system overview.*

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 offering 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 coded. These bits are clustered into Galois symbols (here, the reference field is GF(2^8)). We call the channel encoded PDUs *information unit*s (IUs). The IUs created by the node, along with those previously decoded, are stored in a buffer.[9] The number of available IUs is P, and the IU symbols are denoted as x_p, where $1 \leq p \leq P$ is the packet index. These IUs are linearly combined so as to create a *coded packet* (CP). In MIMO_NC, each CP may include one or two linear combinations of the IUs, and the NC mixing matrices are denoted as $G^{(r)}$ and $G^{(i)}$. We shall assume for the time being that there is some rule that decides whether one or two combinations should be produced, but some examples of actual, distributed decision criteria will be described later. If only one combination is computed every time, this scheme will be referred to as Basic MIMO_NC, while if the converse happens (all CPs contain two linear combinations) the system will be called Super MIMO_NC. In the latter case, the rate is halved, but it will be shown that this rate decrease leads to a diversity order gain. Incidentally, a receiver may get packets with different modulation formats, but the decoding process can support a mixed set of CPs. The n-th CP ($1 \leq n \leq N$) contains one or two linear combinations of the P IUs. In the latter case, the output can be represented as:

$$d_n^{(r)} = \sum_{p=1}^{P} g_{np}^{(r)} x_p, \quad d_n^{(i)} = \sum_{p=1}^{P} g_{np}^{(i)} x_p \tag{2.3}$$

while in the former only $d_n^{(r)}$ is created.

If only one combination is produced, BPSK is chosen as the modulation format, otherwise a QPSK constellation is employed in order to keep the same transmission time. For the rest of this section, unless otherwise stated, two linear combinations per CP and QPSK constellation will be

[9] In this Section, it is assumed that the payload of each packet includes only one Galois symbol. This comes at no loss of generality because each set of symbols is decoded independently.

assumed because this is the more general case. For BPSK, only the real part of the received signal is processed. The header of each CP includes the NC random coefficients $g_{np}^{(r)}, g_{np}^{(i)},\ p = 1, \ldots, P$ used to combine the IUs. The Galois symbols $d_n^{(r)}, d_n^{(i)}$ are turned into bits (G/b operation in Fig. 2.7), and the bits are mapped into QPSK symbols $q_{n,k}, 1 \leq k \leq 8$. To each pair of Galois symbols $d_n^{(r)}, d_n^{(i)}$ corresponds a vector $\mathbf{q}_n$ of 8 modulated symbols. The corresponding waveforms are sent through the wireless channel and are collected by the receiver. The channel is Rayleigh distributed and frequency flat, and can be represented by a complex circularly symmetric Gaussian random variable $h_n = h_n^{(r)} + jh_n^{(i)}$.[10] Thus, the real and imaginary part of the received complex sample $y_n = y_n^{(r)} + jy_n^{(i)}$ can be written as:

$$\begin{pmatrix} y_n^{(r)} \\ y_n^{(i)} \end{pmatrix} = \begin{pmatrix} h_n^{(r)} & -h_n^{(i)} \\ h_n^{(i)} & h_n^{(r)} \end{pmatrix} \begin{pmatrix} q_n^{(r)} \\ q_n^{(i)} \end{pmatrix} + \begin{pmatrix} \eta_n^{(r)} \\ \eta_n^{(i)} \end{pmatrix} \tag{2.4}$$

where $\eta_n^{(r)} + j\eta_n^{(i)}$ is a complex valued, circularly symmetric Gaussian noise with variance $\sigma^2/2$ per component.

The destination estimates the channel (we shall assume that perfect channel state information is available at the receiver) 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 $G = [G^{(r)}; G^{(i)}]$[11] matrices. 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 real and imaginary parts of the samples of the n-th coded packet are gathered into the column vectors $\mathbf{y}_n^{(r)}, \mathbf{y}_n^{(i)}$, respectively. The $2N$ vectors $\mathbf{y}_n^{(r)}, \mathbf{y}_n^{(i)}$ that belong to the same generation are stacked on top of each other, so as to build a $16N$ vector $\mathbf{y} = [\mathbf{y}^{(r)}; \mathbf{y}^{(i)}] = [\mathbf{y}_1^{(r)}; \mathbf{y}_2^{(r)}; ...; \mathbf{y}_1^{(i)}; ...; \mathbf{y}_N^{(i)}]$, that is passed to the MIMO_NC decoder along with the coding matrix, G.

In conventional NC, each packet is separately demodulated and the NC coefficients are extracted from the packet header. Note that the two problems of demodulation and NC decoding are carried out in separated stages. Classical NC can accept only non-redundant packets successfully processed by PHY, whereas any other packets cannot 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, MIMO_NC adopts it. The equivalent input-output relation for N received CPs can be written starting from Eq. (2.4) (see also [64]):

$$\begin{pmatrix} \mathbf{y}^{(r)} \\ \mathbf{y}^{(i)} \end{pmatrix} = \begin{pmatrix} H^{(r)} & -H^{(i)} \\ H^{(i)} & H^{(r)} \end{pmatrix} \begin{pmatrix} \mathbf{q}^{(r)} \\ \mathbf{q}^{(i)} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\eta}^{(r)} \\ \boldsymbol{\eta}^{(i)} \end{pmatrix} \tag{2.5}$$

[10] Throughout this Section, the real (imaginary) part of a complex number a is denoted by $a^{(r)}$ $\left(a^{(i)}\right)$, and similarly for vectors

[11] In the rest of this Section, when we use the notation [x;y], we refer to the vectors (or matrices) x and y which are stacked on top of each other, i.e., $\begin{bmatrix} x \\ y \end{bmatrix}$.

where $H^{(r)}$ and $H^{(i)}$ are $8N \times 8N$ diagonal matrices, $\mathbf{q}^{(r)}(\mathbf{q}^{(i)})$ is an $8N$ vector whose $8(n-1)+1, ..., 8n, 1 \leq n \leq N$ elements are equal to $q^{(r)}_{n,k}(q^{(i)}_{n,k})$ 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 8×8 matrices $H^{(r)}_n(H^{(i)}_n), 1 \leq n \leq N$. Each $H^{(r)}$ $\left(H^{(i)}\right)$ is equal to the 8×8 identity matrix multiplied by $h^{(r)}_n$ $\left(h^{(i)}_n\right)$. We point out that $H = \begin{bmatrix} H^{(r)} & -H^{(i)} \\ H^{(i)} & H^{(r)} \end{bmatrix}$ is orthogonal and $H' = H^T H$ is a diagonal matrix. Each element in the diagonal is exponentially distributed, and all the elements with index in $\{8(n-1)+1, .., 8n\}$ and $\{8(N+n-1)+1, .., 8(N+n)\}$ are equal to the squared envelope 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:

$$\mathbf{Y} = H'G\mathbf{x} + \boldsymbol{\eta}' \tag{2.6}$$

where $\mathbf{Y}$ is the $16N$ vector of the processed received samples, $H' = H^T H$, $G = [G^{(r)}; G^{(i)}]$ and $\boldsymbol{\eta}' = H^T\boldsymbol{\eta}$. We point out that Eq. (2.6) implies a slight abuse of notation, because $G\mathbf{x}$ is a vector of Galois numbers and H' is a matrix of real numbers. It is understood that the CPs are modulated and thus H' actually multiplies the baseband equivalent of the modulated waveforms (which can be represented by complex numbers). Eq. (2.6) is useful because it explicitly shows the parallel between a MIMO system and NC. The real and imaginary parts of the k-th noise sample of the n-th CP $\eta'^{(r)}_{n,k}(\eta'^{(i)}_{n,k})$ are zero mean Gaussian random variables with variance $|h^2_n|\sigma^2/2$, since they can be expressed as $h^{(r)}_n\eta^{(r)}_{n,k} - h^{(i)}_n\eta^{(i)}_{n,k}$ and $h^{(i)}_n\eta^{(r)}_{n,k} + h^{(r)}_n\eta^{(i)}_{n,k}$, respectively.

Therefore, for any Galois input $\mathbf{x}$, there is a well defined set of modulated waveforms. The ML criterion picks the $\mathbf{x}$ that minimizes the distance between the expected postprocessed received symbols $[H'\mathbf{q}^{(r)}; H'\mathbf{q}^{(i)}]$ and the actual processed samples $\mathbf{Y}$. 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 code. 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 [65], which is a modification of the sphere decoding algorithm (see [65] and references therein).

In order to describe this algorithm, the above Galois system must be rewritten. Any Galois matrix G can be expressed as $\Pi G = LU$, where Π is a permutation matrix, L is lower triangular and U is upper triangular [66]. Since, for Galois fields, $\Pi^{-1} = \Pi$, it stems that $G = \Pi LU$. Therefore the problem can be decomposed into two subproblems:

$$\mathbf{y} = H'\Pi LU\mathbf{x} + \boldsymbol{\eta} = H'\Pi L\mathbf{z} + \boldsymbol{\eta}, \;\; \mathbf{z} = U\mathbf{x} \tag{2.7}$$

where a dummy $P \times 1$ vector $\mathbf{z}$ is introduced, so that the easier problem $\mathbf{y} = H'\Pi L\mathbf{z}+\boldsymbol{\eta}$ needs to be solved. Given a solution $\mathbf{z}^*$, $\mathbf{x}$ is easily found by conventional backsubstitution [66], since U is upper triangular. The problem $\mathbf{y} = H'\Pi L\mathbf{z}+\boldsymbol{\eta}$ 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 Π is the identity matrix. We call $\mathbf{y}_{n'}$ the 8-element column vector that includes the components of

$\mathbf{y}$ whose index is between $8(n'-1)+1$ and $8n'$, $1 \leq n' \leq 2N$. 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 column vector made up by the first $\min(n', P)$ elements of $\mathbf{z}$. The algorithm picks the vector $\mathbf{z}$ that minimizes the distance $\|\mathbf{y} - H'L\mathbf{z}\|^2$, which can be written, with a slight abuse of notation, as the sum of $2N$ components: $\sum_{n'=1}^{2N} \gamma_{n'}^2 = \sum_{n=1}^{2N} |\mathbf{y}_{n'} - h_{n'}\mathbf{L}_{n'}\mathbf{z}_{n'}|^2$. 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 γ_1. If this value is smaller than a certain threshold ρ^2, called the *squared sphere radius*, it will proceed to consider $\mathbf{z}_2$, 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 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. On the other hand, 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.

As a concluding remark, we highlight that MIMO_NC is based on soft decoding, while standard NC employs hard decoded bits. Thus MIMO_NC may use corrupted or redundant packets and provides an SNR gain over NC. Thus it can benefit from all the received packets, and is less vulnerable to corrupted packets. However, if the rate of the transmitted CPs is low (in a sense to be specified in the next Section), MIMO_NC can achieve a diversity increase that neither soft decoding nor NC attain.

2.3.2 Performance Analysis

In order to gain some initial understanding, consider a simple case study that can be quite easily analyzed. The sample network is reported in Fig. 2.8 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 occur in a network where data dissemination has reached several nodes, thus many terminals can combine multiple packets at once. In this case it is common that some nodes transmit to the same receiver different coded packets based on the same information units [43].

In this Section, Basic MIMO_NC is analyzed before Super MIMO_NC. The reason is that in Basic MIMO_NC, the matrices G and H are half as large as the Super MIMO_NC matrices. Therefore, the analysis is easier and clear-cut and moreover the analysis for Super MIMO_NC stems as a straight-

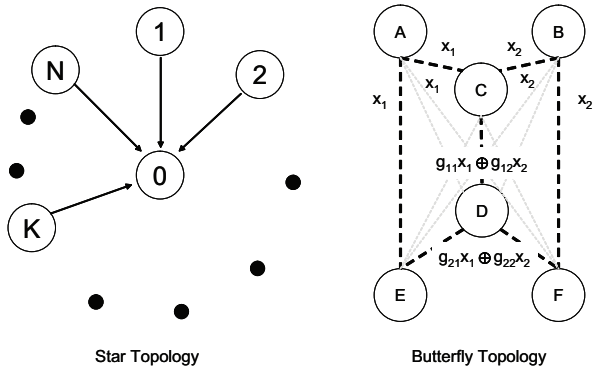

Figure 2.8. *The test networks.*

forward extension of that for Basic MIMO_NC. Finally, only in this section, the channel coefficients h_n represent only Rayleigh fading for ease of notation; the path loss will be included in the average received power P_r.

Classical NC performance

The analysis of 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.[12] If fading is constant over a whole packet, and it is frequency flat and Rayleigh distributed, the average packet error probability P_{pk} is inversely proportional to the SNR [52]. 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 fading coefficients h_n^2 being iid. The probability of receiving fewer than P correct packets out of N is:

$$P_{err} = \sum_{k=0}^{P-1} \binom{N}{k} (1-P_{pk})^k P_{pk}^{N-k} \tag{2.8}$$

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:

$$P_{err} \simeq \binom{N}{P-1} P_{pk}^{N-(P-1)} = \binom{N}{P-1} P_{pk}^{N-P+1} \tag{2.9}$$

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

Basic MIMO_NC performance

The equivalent input/output relation for Basic MIMO_NC was reported in Eq. (2.7). We recall that in Basic MIMO_NC, BPSK is employed and thus only the real part of the received signals is

[12] This approximation does not consider the negligible probability that the NC matrix may not be invertible. This probability decays as $1/(256^{(N-P+1)})$

considered. Hence, all vectors and matrices have half the size as in Section 2.3.1. In addition, since only the real part of the incoming signals is considered, there is no need to left-multiply the received vectors by H^T in order to decouple real and imaginary parts, and thus only matrix H, rather than H', is needed by the ML decoder. Nonetheless, the computation of the exact error probability is rather hard and the pairwise error probability will be pursued instead, because it reveals the diversity order of the system.

Let us call *codeword* the vector of the CPs. Since it is assumed that each CP includes only one Galois Symbol, here coding is applied across packets rather than inside a single packet, as is usually done in channel coding. We shall denote the codewords by the symbols $\mathbf{c}_i$, where i is an integer index, and $\mathbf{c}_i = G\mathbf{x}_i$, where $\mathbf{x}_i$ is the set of IUs that generate that codeword. By definition $\mathbf{c}_0$ is the all zero codeword and it is assumed to be the transmitted codeword. 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 conditional probability that:

$$\begin{aligned} \|HG\mathbf{x}_0 - \mathbf{y}\|^2 &> \|HG\mathbf{x}_1 - \mathbf{y}\|^2 \\ \|HG\mathbf{x}_0\|^2 - 2(HG\mathbf{x}_0)^T\mathbf{y} + \|\mathbf{y}\|^2 &> \|HG\mathbf{x}_1\|^2 - 2(HG\mathbf{x}_1)^T\mathbf{y} + \|\mathbf{y}\|^2 \\ (HG(\mathbf{x}_0 - \mathbf{x}_1))^T\mathbf{y} &< 0 \end{aligned} \tag{2.10}$$

given that $\mathbf{c}_0$ was sent and all transmitted symbols are BPSK, thus $\|HG\mathbf{x}_i\|^2$ does not depend on the codeword $\mathbf{c}_i,\ i = 0, 1$.

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

$$t_n = \sum_{k=1}^{8} [h_n(b_{n,k}^{(0)} - b_{n,k}^{(1)})][h_n b_{n,k}^{(0)} + \eta_{n,k}] \tag{2.11}$$

where $b_{n,k}^{(\ell)}$ is the $8(n-1)+k$-th modulated BPSK symbol of the ℓ-th codeword ($\ell \in \{0, 1\}$) and $\eta_{n,k}$ is the k-th element of $\boldsymbol{\eta}^{(n)},\ k \in \{1, \ldots, 8\}$. Since $\mathbf{c}_0 = 0$, it follows that $b_{n,k}^{(0)} = -1, \forall n, k$. Clearly each of the terms that make up t_n is non zero if $b_{n,k}^{(0)} \neq b_{n,k}^{(1)}$.[13] Let w_n be the number of different bits in the n-th Galois symbol between $\mathbf{c}_0$ and $\mathbf{c}_1$ ($w_n \in \{0, 1, .., 8\}$). After some algebra, the decision statistics $t = \left(\sum_{n=1}^{N} t_n\right)/2$ is found as:

$$t = \sum_{n=1}^{N} h_n^2 w_n + \sum_{n=1}^{N} h_n \sum_{k=1}^{8} ((b_{n,k}^{(0)} - b_{n,k}^{(1)})/2)\eta_{n,k} \tag{2.12}$$

There is a decoding error if $t < 0$. In Eq. (2.12), 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

[13] A term may vanish also if $h_n b_{n,k}^{(0)} + \eta_{n,k} = 0$, but this is a zero probability event.

$\sum_{n=1}^{N} h_n \sum_{k=1}^{8}((b_{n,k}^{(0)} - b_{n,k}^{(1)})/2)\eta_{n,k}$ 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 $\sum_{n=1}^{N} h_n^2 w_n P_r$ and variance $\sum_{n=1}^{N} h_n^2 w_n \sigma^2$ [67]. Thus the error probability conditioned to the channel state is:

$$P_{block} = Q\left(\sqrt{\frac{(\sum_{n=1}^{N} h_n^2 w_n)^2 P_r}{(\sum_{n=1}^{N} h_n^2 w_n)\sigma^2}}\right) = Q\left(\sqrt{\sum_{n=1}^{N}(h_n^2 w_n) SNR}\right)$$

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, that is to say the Hamming distance between $\mathbf{c}_0$ and $\mathbf{c}_1$. If G is regarded as the generator matrix of a linear block code, the minimum Hamming distance of the code is the diversity order of the symbol error rate of the code. The best case occurs when the code achieves the Singleton bound [68], 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 [52]. 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 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, i.e., we can find a $\mathbf{x}_1$ such that $\mathbf{c}_1 = G(\mathbf{x}_1 - \mathbf{x}_0)$ has $P - 1$ zeros. For instance, consider a vector $\mathbf{d}$ whose first $P - 1$ components are 0 and the last one is an arbitrary non zero Galois symbol. If the matrix with the first P rows of G is denoted $G(1 : P, :)$, the system $\mathbf{d} = G(1 : P, :)\mathbf{x}_1$ has a unique solution if $G(1 : P, :)$ is full rank (which happens with high probability), and thus it is possible to find many such codewords that effectively reduce the system diversity. Diversity can be regained if no such codewords exist.

Therefore, 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. This answers the second question raised in the Introduction. This problem leads directly to the third question at the beginning of this section: what would be the performance of a MIMO_NC system whose input rate is changed? Let us assume that the input symbols are drawn from 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 $N - P + 1$ elements from the correct codeword are fewer, and if there are none of them the diversity is N. Table 2.1 shows, for $P = 2, N \in \{2, 3, 4\}$, 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 is an inherent tradeoff between rate and performance (in terms of diversity and SNR gain). As Section 2.3.2 and the Simulation Results Section will show, a rate adaptation scheme (called Adaptive MIMO_NC) can explot this tradeoff. On the other hand, Basic MIMO_NC does achieve the maximum possible diversity order for full rate communications and outperforms classic NC, since it offers a SNR gain and can exploit packets that would not be considered by conventional NC. Even

Table 2.1. *Probability of having full diversity in a $P = 2$ Star system on varying the input rate*

k	1	2	3	4	≥ 5
N=2	100%	94.6%	76.4%	23.4%	0%
N=3	98.6%	92.2%	65%	11.4%	0%
N=4	98.5%	88.9%	59.1%	6.7%	0%

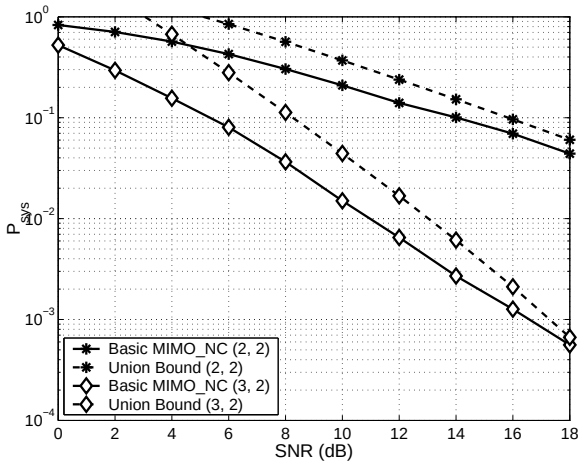

Figure 2.9. *Comparison of simulated Basic MIMO_NC and Union Bound when $P = 2$ and $N = 2, 3$.*

though the diversity order is the same for Basic 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. The results Section will confirm that this feature leads to faster data dissemination and higher reliability.

In order to check the correctness of our analysis, we have compared the Union Bound [68] for Basic MIMO_NC when $P = 2$ and $N = 2, 3$ with the simulated Basic MIMO_NC (Fig. 2.9). 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.

Super MIMO_NC

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. By following very similar steps as for Basic MIMO_NC, the decoder decides for a codeword $\mathbf{c}_1$ if $(H'G(\mathbf{x}_0 - \mathbf{x}_1))^T\mathbf{y} < 0$, given that the all zero codeword $\mathbf{c}_0$ was sent.

Let us assume that a QPSK symbol is received with average power P_r. Then, after some algebraic steps that closely follow those of the Section 2.3.2.B, the decision statistic turns out to be:

$$t = \sum_{n=1}^{N} |h_n|^2 w_n \sqrt{\frac{P_r}{2}} + \sum_{n=1}^{N}\sum_{k=1}^{8}((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2)\eta_{n,k}^{\prime(r)} + \sum_{n=1}^{N}\sum_{k=1}^{8}((b_{n,k}^{(i,0)} - b_{n,k}^{(i,1)})/2)\eta_{n,k}^{\prime(i)} \quad (2.13)$$

where $b_{n,k}^{(r,0)}$ ($b_{n,k}^{(i,0)}$) is the in phase (in quadrature) bit of the k-th ($1 \leq k \leq 8$) QPSK symbol in the n-th packet, $\eta_{n,k}^{\prime(r)}$ ($\eta_{n,k}^{\prime(i)}$) is the real (imaginary) part of the k-th noise sample for the n-th coded frame and $w_n = w_n^{(r)} + w_n^{(i)}$, with $w_n^{(r)} = \sum_{k=1}^{8}((b_{n,k}^{(r,0)} - b_{n,k}^{(r,1)})/2)$ and $w_n^{(i)} = \sum_{k=1}^{8}((b_{n,k}^{(i,0)} - b_{n,k}^{(i,1)})/2)$ the numbers of non zero bits in the n-th component of $G^{(r)}(\mathbf{x}_0 - \mathbf{x}_1)$ and $G^{(i)}(\mathbf{x}_0 - \mathbf{x}_1)$, respectively.

A decoding error happens if $t < 0$. By similar arguments to Section 2.3.2.B, the error probability is:

$$P_{err} = Q\left(\sqrt{\frac{\left(\sum_{n=1}^{N}|h_n|^2 w_n\right)^2 P_r/2}{\left(\sum_{n=1}^{N}|h_n|^2 w_n\right)\sigma^2/2}}\right) = Q\left(\sqrt{\sum_{n=1}^{N}\left(|h_n|^2 w_n\right) SNR}\right) \quad (2.14)$$

The diversity order is equal to the number of non zero terms in the sum in the argument of the Gaussian complementary cumulative distribution function (the Q function). 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 [68], i.e., the minimum distance is $2N - P + 1$. 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:

$$D(N, P) = N - \left\lfloor \frac{P-1}{2} \right\rfloor \quad (2.15)$$

If $P = 2$, one term vanishes, but no diversity is lost, since each channel is present twice in Eq. (2.14). 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 1 to $2N$, $P-1$ of them are removed. What is the probability that one odd index and the following even one 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 indices such that a ball with a odd index is followed by one with an even index. This probability is $(2N - 2k)/(2N - k)$. Thus the probability of having diversity N is:

$$P_{full} = \prod_{k=1}^{P-1} \frac{2N - 2k}{2N - k} \quad (2.16)$$

For large N and fixed P the probability of having full diversity approaches 1, because all the factors in Eq. (2.16) go to one.

Adaptive MIMO_NC

The previous analysis has shown that Super MIMO_NC can guarantee a diversity order of at least $N - \left\lfloor \frac{P-1}{2} \right\rfloor$. 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 Basic 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, but with correct header in any case) 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 Basic MIMO_NC is used. If this 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 Basic 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 distributed fashion, without any exchange of information to coordinate them. Thus Basic MIMO_NC and Super MIMO_NC can seamlessly coexist, and this is another reason that makes Adaptive MIMO_NC viable.

2.3.3 Performance Evaluation

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 three different scenarios. First, we consider the simple topology described in Fig. 2.8 by varying N and P. This scenario will be named *Star Topology*. Second, the performance of the different versions of MIMO_NC are compared in the well known *Butterfly Topology* (Fig. 2.8) [27]. Finally, we test MIMO_NC in random networks.

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} which is defined as the probability that at least one of the destination nodes does not receive one or more packets intended for it.

Star Topology

We report in this Section simulation results about the Super MIMO_NC performance compared with Basic MIMO_NC approach and classical NC in the Star topology which is representative of common situations like data dissemination through random network coding, where the same information is present at many neighboring nodes.

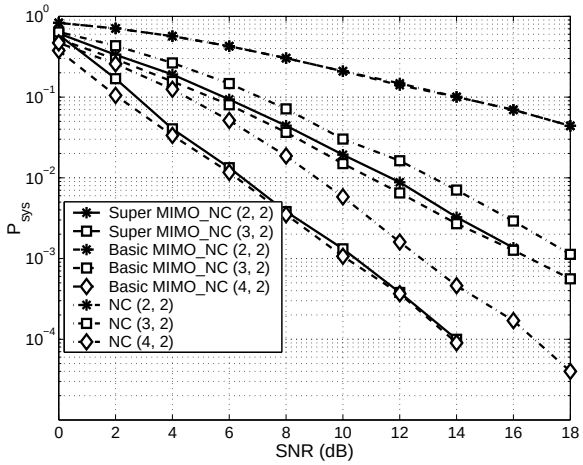

Figure 2.10. *System error probability, star topology,* $P = 2$

In the Star configuration, N nodes $(1, \cdots, N)$ share the same P IUs and they are charged to send them to the destination, i.e., the central node 0. In Fig. 2.10 we report the system error probability, P_{sys} (i.e., the probability that node 0 can not decode all P IUs) for the case $P = 2$ and $N = 2, 3, 4$ for Basic MIMO_NC and standard NC, while $N = 2, 3$ for Super MIMO_NC. We point out that the diversity order changes according to Eq. (2.15). Let us focus on the case $P = 2$ and $N = 2$. In this situation, it can be noted that Basic MIMO_NC and NC achieve the same performance while Super MIMO_NC trades off efficiency (measured as N over the total number of transmitted linear combinations) for higher diversity and power gains over the other schemes (e.g., about 8 dB at $P_{sys} = 10^{-2}$).

It is also possible to compare Basic MIMO_NC and Super MIMO_NC with equal bandwidth requirements. This can be done by considering the curves for Basic MIMO_NC ($P = 2, N = 4$) and Super MIMO_NC ($P = 2, N = 2$). In this case, Basic MIMO_NC outperforms Super MIMO_NC, but it comes at the price of twice as many nodes involved. The conclusion that may be drawn is the following: as Section 2.3.2 proved, some redundancy is needed to achieve additional diversity in wireless NC. When this redundancy can be attained by having more nodes transmitting the same information, Basic MIMO_NC is the system to be preferred. Instead, when the only viable way is to reduce the spectral efficiency of each packet, Super MIMO_NC provides a more significant reliability improvement than Basic MIMO_NC.

Butterfly Topology

In this Section, we consider the Butterfly Topology (see Fig. 2.8) which is one of the best known reference scenarios for network coding [27]. 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 CP which combines x_1 and x_2. Note that, in this situation, intermediate nodes can transmit some packets only if they can successfully recover some 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 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. 2.11 compares the system error probability of Super MIMO_NC, Basic 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 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 Basic MIMO_NC and NC. Finally, we note that both Basic 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 Basic MIMO_NC and classical NC, the two sources may send linearly dependent CPs. If this is the case, the received 2×2 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 the value observed in Fig. 2.11.

The advantages of the Adaptive MIMO_NC scheme are pointed out in Fig. 2.12, where the transmission efficiency is shown. This metric is defined as the ratio between N and the number of transmitted linear combinations of the IUs. The lower the transmission efficiency, the lower the error probability (because more redundancy is available), but also the more the required bandwidth. By definition, Basic MIMO_NC has an efficiency of 1, since each CP contains only one linear combination. Instead, Super MIMO_NC implies a value of 1/2. Adaptive MIMO_NC can achieve a higher efficiency than Super MIMO_NC guaranteeing also good performance in terms of error probability (see Fig. 2.11). We also observe that, in the Butterfly topology, there can be at most four transmissions. By the adaptive scheme a ratio of at most 0.75 can be achieved, as the source nodes transmit two linear combinations each while the relay nodes may send just one, therefore the maximum efficiency averaged over all nodes is $4/(2+2+1+1) = 0.75$. In addition, the average efficiency achieved by the adaptive scheme depends on the selection of the threshold T. The higher the threshold value, the lower the efficiency ratio. Moreover, for high average SNR, the relay nodes never need to employ Super MIMO_NC, therefore the upper bound for the efficiency (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 efficiency of MIMO_NC systems.

Random Topology

In this Section, we study the performance of MIMO_NC when the node positions are random. Ten nodes are uniformly distributed in a 90m$\times$ 90m square and two source nodes have one IU each and the goal is to have all nodes decode both IUs. A TDMA scheduler arranges transmissions according to the following rules. All nodes have an index, running from 1 to 10. The time frame is slotted and in the first slot node 1 (one of the sources) transmits its IU, while in the second slot terminal 2 (the

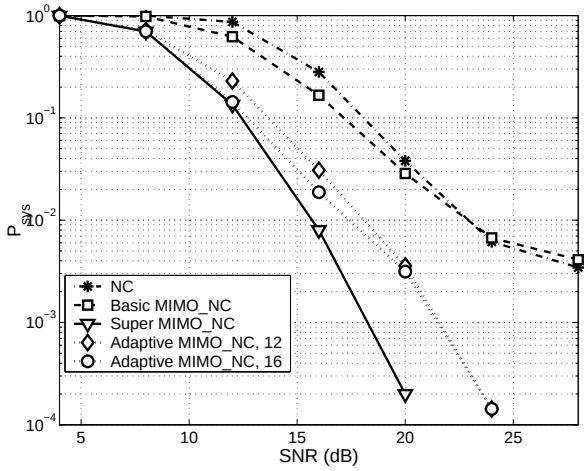

Figure 2.11. *System error probability, butterfly topology*

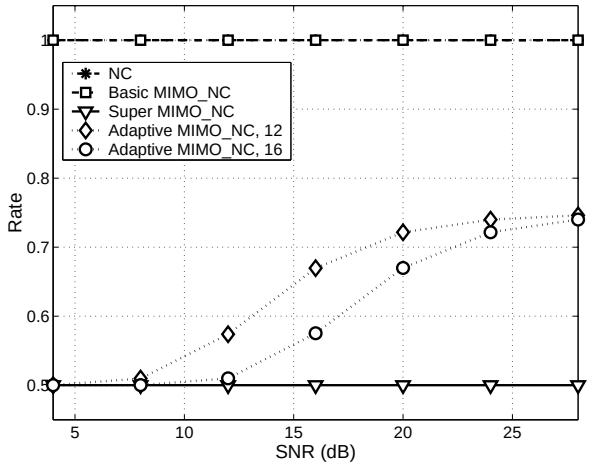

Figure 2.12. *Achieved rate, butterfly topology*

other source) does the same with its IU. Then, in the following slot, the node with the smallest index that has not yet transmitted and has decoded some IUs will broadcast a CP and so on. This scenario is interesting because, while rather idealized, it mimics a real world data distribution system. Moreover, NC performance is known to suffer because of high PERs, and therefore if the transmit power is low traditional NC may no longer be effective. Instead the ability of MIMO_NC of employing discarded or redundant packets can lead to significant performance improvements.

We shall focus on four metrics of interest: the average time for a terminal (different from the sources) to decode all IUs, the probability that a node may decode the IUs, the average time for all nodes in the network to decode all IUs (if it happens) and finally the probability that all the nodes in the network decode all IUs. All metrics are plotted against the average SNR at a distance of 90m (that is to say, between two nodes located on two adjacent vertices of the square). At low SNR (-8 dB) the network is about three hops wide. At high SNR (2 dB) a packet needs at most two hops.

The first metrics deal with the performance of these schemes from the point of view of the average terminal. They are shown in Figs. 2.13 and 2.14. As is clear, MIMO_NC outperforms NC in terms

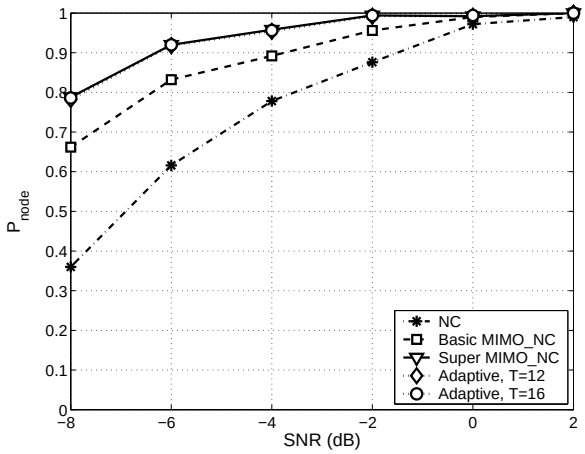

Figure 2.13. *Probability to deliver IUs to a node*

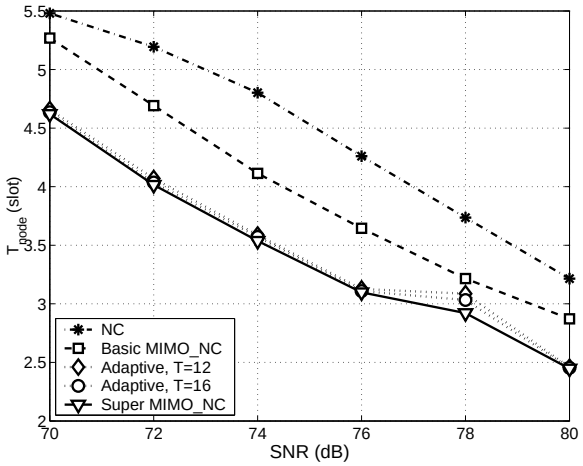

Figure 2.14. *Average time to deliver IUs to a node*

of decoding probability, especially in the low SNR regime. On the other hand, Super MIMO_NC or Adaptive MIMO_NC can further improve reliability or delivery time over Basic MIMO_NC, albeit at a lower transmission efficiency. This confirms that MIMO_NC trades off rate for reliability, and it can be adapted to achieve different performance targets.

However, the most interesting results hold for the last two metrics, which refer to the possibility that all nodes may decode all IUs (Fig. 2.15-2.16). These metrics measure the outage behavior of the transmission scheme. Clearly, it is harder to guarantee delivery of all IUs to all the network nodes, and the resilience of MIMO_NC to channel errors and fading makes an impressive difference. As far as the network-wide delivery probability is concerned, even Basic MIMO_NC outperforms NC by a factor of 4 or more at low SNR, let alone the other more conservative schemes. The average time for this delivery is as much as 25% lower. Finally, note the increased gap between MIMO_NC curves and standard NC from Fig. 2.13 to 2.15, which proves that the outage behavior can be greatly improved. This shows that MIMO_NC is more effective in real world propagation environments than NC and is able to guarantee remarkably higher performance standards than conventional NC.

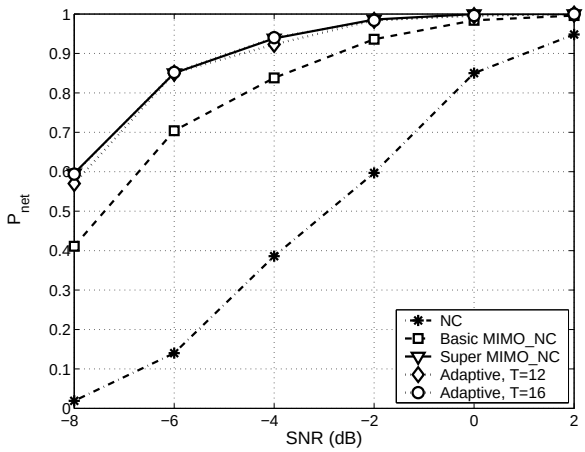

Figure 2.15. *Probability to deliver IUs to all network*

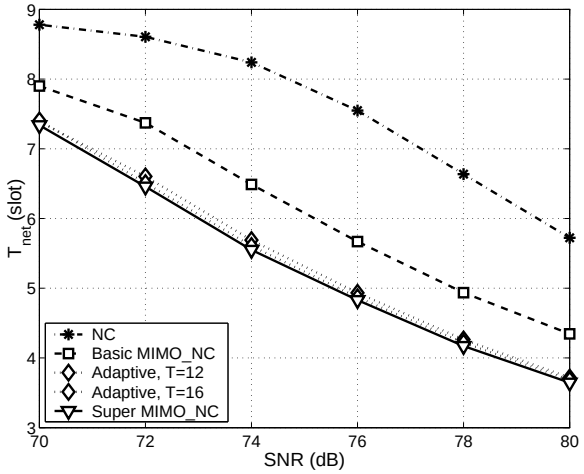

Figure 2.16. *Average time to deliver IUs to all network*

In conclusion, MIMO_NC can achieve much better reliability than standard NC, and is able to trade off delivery time for reliability. In these discussions, we have assumed that two IUs are injected ($P = 2$). We expect that the gap between MIMO_NC and NC would not decrease as P is increased, because MIMO_NC is less sensitive to the threshold reaction of NC to packet losses, and it would show a softer performance degradation as the SNR is reduced. Results in [4] confirm this idea.

2.3.4 Discussion

In this Section a new joint demodulation/NC decoding system, called MIMO_NC, has been explored. Its aim is to reap the spatial diversity inherent in NC, and it has been shown that this diversity cannot be achieved without rethinking both the coding and the decoding phases of NC. MIMO_NC is flexible enough to be throughput efficient or benefit from spatial diversity, without requiring multiple antennas at each node. Theoretical analysis and simulation results have proved its effectiveness in many network scenarios.

2.4 Phoenix: A Hybrid Cooperative-Network Coding Protocol for Fast Failure Recovery in Ad Hoc Networks

Cooperative communication techniques [69] have recently attracted much interest in the wireless research community. The possibility of achieving a low bit error rate without the need for special hardware (e.g., multiple antennas) has been the focus of a significant deal of research, especially at the physical layer. More recent studies have proposed medium access strategies that try to take advantage of cooperation. For instance, in [70] the authors propose a cooperative MAC scheme based on IEEE 802.11 where each node proactively selects a relay for cooperation and lets it transmit simultaneously when beneficial in mitigating interference from nearby terminals. The authors of [71] design a protocol, named CoopMAC, in which each node maintains a table, called CoopTable, of potential helper nodes. At any communication, a terminal can select either direct transmission or transmission through a helper node in order to minimize the total delivery time. In [72] a source assisted way to select the relay node is proposed. In the area between the transmitter and the destination, the source identifies a relaying region. Any eligible terminal, when a cooperative phase is required, contends to become a relay by starting a timer. The node with the shortest backoff actually performs the ARQ transmission. These approaches introduce some additional overhead and require the source to coordinate the cooperators' activity.

The cooperative paradigm introduces new issues also at the network layer.

For instance, in all the classical cooperative schemes, a relay that performs a retransmission on behalf of another node must delay its own frames. So far only a limited amount of attention has been devoted to this aspect (e.g., [73] studies cooperative routing), and one of the main aims of our work is to address these challenges.

Another approach that improves efficiency is Network Coding (NC), whose aim is to reduce the number of transmissions by efficient packet coding. NC and cooperation have different but complementary goals: NC increases the efficiency of each transmission by sending multiple packets combined together, while cooperation tries to make a smart use of this redundancy in order to increase spatial diversity. While in the former technique several terminals send different packets to different terminals, in the latter several nodes send copies of the same packet to the same destination.

Recently, the interaction between NC and cooperation has received some interest [54–57,59]. The idea of moving network coding towards the physical layer has been first proposed in [54]. In [55], the authors propose a way to implement a physical network coding scheme over a two–way relay channel. Also in [57] an approach that exploits network coding during a relaying phase is proposed. Some research [56,59] deals with the integration of channel coding and network coding. These studies exploit the efficiency of network coding in order to both increase the performance of channel coding and reduce the channel decoding complexity.

Network coding and cooperation have been recently joined in [4] in a completely new way. This work has shown how to get the diversity gain of cooperation with the throughput efficiency of NC by means of signal processing techniques borrowed from MIMO, hence its name MIMO_NC. The basic property of MIMO_NC is that N nodes may send combinations, called coded packets, of the

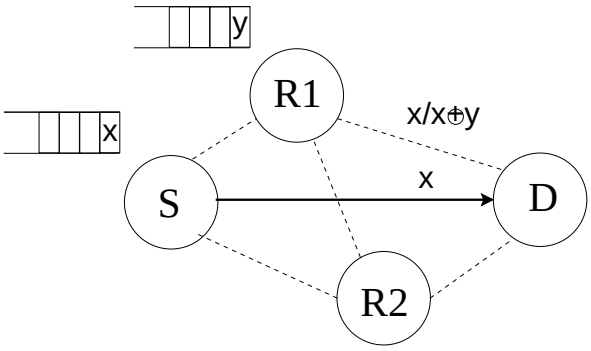

Figure 2.17. *Reference topology.*

same P original information units, IUs, and simultaneously achieve a diversity order of at least $N - \lfloor (P-1)/2 \rfloor$, as well as a coding gain with respect to conventional NC. For instance, if a relay sends a combination of a retransmitted packet and a packet of its own ($N = 2, P = 2$), the diversity order is still two as in a conventional cooperative protocol, but the relay has been able to cooperate in the classical sense, while at the same time also serving its own traffic.

As highlighted by the above example, an approach that combines the advantages of diversity (via cooperation and MIMO processing) and packet coding (via the network coding paradigm) makes it possible for cooperating nodes to also send their own traffic at the same time, thereby removing a significant drawback of traditional cooperative schemes where a cooperating node typically has to delay its own packets. However, due to their main focus on physical layer issues, most of the existing papers on this topic (including the original work that proposed MIMO_NC [4]) have not addressed the issues and the tradeoffs that arise in this case, which mostly relate to the MAC and network layers. An in-depth study of such protocol design issues in the context of MIMO_NC and for realistic network scenarios is the purpose of this branch.

More specifically, the contribution of our work is twofold. i) We propose a fully distributed relay election procedure suitable for a decode-and-forward protocol. This solution is simple and easily implementable with off-the-shelf WLAN cards. ii) We design a novel cooperative MAC protocol based on MIMO_NC called Phoenix. Its main feature is to allow cooperators to code their own packets together with the ones to be relayed, with no additional cost in time and energy.

2.4.1 Phoenix: A Cooperative - NC Protocol

In this paragraphs we propose a novel MAC protocol called Phoenix that relies on the MIMO_NC physical layer (PHY) in order to leverage cooperative relaying and network coding techniques. Our solution suitably extends the basic IEEE 802.11 CSMA medium access policy without channel negotiation [15], which is based on carrier sensing and Binary Exponential Backoff (BEB). In the remainder of the Section we will first describe a completely distributed mechanism to perform cooperation among nodes and then extend it to the full Phoenix protocol by introducing the contention procedures needed to successfully combine relayed and original packets.

CCSMA: A Cooperative CSMA

Consider the topology depicted in Fig. 2.17, and suppose that S sends a packet to D. Once the transmission is accomplished, the source waits for a feedback from the destination of the frame in order to determine how to behave. Three conditions can occur: i) the recipient has correctly decoded the whole data packet, ii) the header of the packet has been received but the payload could not be decoded, iii) the destination has not decoded the header. We point out that condition ii) can hold because the header is assumed to have a stronger FEC than the data and a separate CRC. If condition i) happens, an acknowledgment message (ACK) is sent. On the other hand, if the destination fails to receive the data but is aware of the attempt performed by the source (i.e., condition ii) holds), a Not ACKnowledged message (NACK) is sent, asking for a retransmission. Finally, condition iii) may occur because the destination has not been able to synchronize to the header (either because the received power was too low or the destination was already engaged in another reception) or because the SNIR has prevented a correct reception not only of the payload but also of the header. In any case, the destination is not able to ask for a retransmission, as no information on the source is available, and no feedback is sent.

Let us now assume that the communication between S and D fails. A pure CSMA protocol in this situation would require S to retransmit the frame after a suitable backoff interval. However, if the communication has failed because of harsh channel conditions between S and D, the following attempt is not likely to succeed unless performed after a period long enough for the channel to decorrelate. [14] On the other hand, thanks to the broadcast nature of the wireless medium, other terminals (e.g., R_1 and R_2) may have decoded the packet sent by S, even though it was not intended for them. The cooperative paradigm proposes that one of these nodes [15] acts as relay by immediately retransmitting the original frame in place of S. Two copies of the same packet are then available at the destination, sent over spatially independent channels. The receiver can therefore exploit spatial diversity by performing Maximum Ratio Combining, and the decoding probability is strongly enhanced. Cooperative relaying, in brief, substitutes the time diversity offered by pure CSMA with spatial diversity, potentially offering a significant reduction of the duration of failure recovery procedures.

The main challenge in defining a cooperative system is to determine a strategy to select nodes that have to act as relays. In this case we consider a fully distributed approach based on carrier sense. A node is eligible as relay for the S-D communication if two conditions are met: i) the node has decoded the data packet sent by S and ii) a frame requiring a retransmission coming from the destination has been received. The latter condition prevents nodes from transmitting useless and potentially harmful (in terms of interference) cooperative packets when they are not needed (i.e., when an ACK is missed by the terminal due to poor channel conditions towards D or when D has not decoded the header of the data packet from S, being unable to perform any kind of recombination). A terminal that satisfies both requirements starts a backoff whose length n is drawn in the interval $[0, CW_{rel}/2]$ if the received

[14] The more correlated the channel, the longer the required period. Therefore, this problem may be particularly severe in low mobility scenarios.

[15] In our work we consider decode-and-forward cooperation, i.e., only nodes that have correctly decoded the packet sent by the source can act as relays.

power P_{rec} for the NACK frame is higher than a given threshold $\bar{P}$, and in $[CW_{rel}/2+1, CW_{rel}]$ otherwise. During the first $n-1$ slots, the node does not perform carrier sense. On the contrary, during the last slot, the power level on the medium is checked as in the normal contention phase. If the channel is sensed busy, the terminal assumes that another node has won the contention to relay data (i.e., another potential cooperator has chosen a shorter backoff window) and gives up its attempt, going back to its own activity. If, instead, the countdown expires with an idle medium, the node actually transmits to D a copy of the original packet received from S. After the transmission is performed, the relay returns to its previous activities without waiting for any feedback from the destination, as it has no need to be informed on the outcome of the communication. If D successfully combines the original and the relayed packet, an ACK message addressed to S is sent. On the contrary, if the cooperative phase fails (either because the relayed packet is not sufficient to recover the original failure or because no packet is relayed at all), the destination sends another NACK frame addressed only to S and therefore not able to trigger another cooperative communication. In this condition, S decides whether to perform another attempt or not, according to the Short Retry Limit (SRL) value. [16] In conclusion, a transmission between S and D is composed of up to SRL phases, separated in time by the usual BEB mechanism, and each phase is made up by an original transmission from S and a (potential) retransmission from a relay.

Let us now make some observations on CCSMA. First of all, we notice that the relaying phase could fail for three main reasons: i) no cooperators may be available, ii) some collisions among relayed packets may occur, and iii) a potential relay could sense the medium busy and give up the contention because of aggregate network interference rather than because of an actual cooperative transmission. The first factor is related to both network density (i.e., topological availability of relays) and overall network activity (i.e., potential cooperators may not decode data coming from S or a NACK, due to interference). This issue is not specific to our strategy but rather affects all the decode-and-forward relaying procedures. As far as collisions among relayed packets are concerned, they are a drawback of the completely distributed approach that we employ. It is clear that the shorter CW_{rel} the higher the probability of having two or more cooperators choosing the same value for n. On the other hand, a long contention window would increase the delay, thereby reducing the advantages of fast failure recovery of cooperative ARQ. In our protocol we have chosen to use a maximum contention period much shorter than the backoff window employed by a pure CSMA strategy while still keeping sufficiently low the probability of collision for the considered network densities (see Section 2.4.3). Moreover, we try to enhance the probability of success for a relaying phase by favoring nodes that experience good channel conditions towards the destination. In fact, the threshold $\bar{P}$ represents the average power of all data packets received from D. Therefore, if a NACK is decoded with a power higher than $\bar{P}$, the fading between the potential cooperator and D is favorable.

Let us now focus on the criterion chosen to give up the contention. The described carrier sense mechanism aims at informing all the contending terminals as soon as a relay accesses the medium. Nevertheless, as pointed out earlier (iii), a node always overhears some amount of overall interference

[16] The Short Retry Limit represents the maximum number of attempts to deliver a packet performed at the MAC layer before dropping it [15].

due to current transmissions in the network, and there is no way to distinguish a specific message out of it. Therefore, if the noise and interference level exceeds the threshold, the contention may be abandoned even if no other node is acting as a relay. In order to reduce the impact of this factor, we ask a node to sense the medium only in the last slot of its backoff window.[17] Suppose, in fact, that a neighboring communication is in place when the potential cooperator receives the NACK frame but ends before the countdown is over. With the usual carrier sensing scheme, the terminal would unnecessarily abandon the contention, while with our approach cooperation may still take place. On the other hand, we would like to observe that giving up cooperation if the overall interference level is still high when the contention window actually expires may be a good solution. This stems from the spatial correlation of interference. In fact, when a potential relay decides to abort its transmission, the destination, which is typically one of its neighbors, is likely to be affected by unfavorable conditions as well. Should the node cooperate anyway, not only would its transmission experience a very low SNIR at the receiver but also the other ongoing communications might experience a collision, with detrimental effects on the network performance.

In conclusion, although the protocol could be easily extended to more cooperative retransmissions per each phase, we have decided not to follow this approach, as the decoding performance would not significantly increase. Actually, a relay phase typically fails because the overall interference at the receiver is too high (see Section 2.4.3). In this situation, another immediate attempt is going to experience similar conditions and therefore it is not likely to succeed, wasting resources and increasing network congestion.

The Phoenix Protocol

Cooperative techniques, as discussed, are able to shorten failure recovery procedures with respect to pure CSMA protocols by exploiting spatial diversity. On the other hand, when a node acts as a relay it has to both delay it own transmissions and spend some of its resources (e.g., energy) in order to help another terminal. These drawbacks may deter potential cooperators from taking part in communication recovery phases, thus limiting the potential advantages of such approaches. We propose a novel MAC protocol, called Phoenix, that is able to go beyond the disadvantages of classic cooperation by letting relay nodes transmit a combination of a cooperative packet and a packet taken from their own queue. This solution is enabled by the MIMO_NC scheme described in Section 2.3. The idea that underpins Phoenix can be explained referring to Fig. 2.17. In CCSMA, node R_1 retransmits S's packet x on its behalf. With MIMO_NC, R_1 can send a frame $x \oplus y$ that combines x and a packet y of its own. Any node that has at least an estimate of x (i.e., it has correctly received x's header) and of $x \oplus y$ is potentially able to decode both frames. With the novel PHY, the relaying phase still provides spatial diversity to the destination of packet x (i.e., cooperative advantages) but at the same time a relay can exploit the cooperative communication to serve its own traffic. The Phoenix approach therefore encourages nodes to cooperate by offering them an advantage and has the potential to sig-

[17] This solution does not worsen the probability of collision among relays. In fact, if a terminal accesses the medium, its transmission lasts much longer than the relay backoff window chosen by any other contending node.

nificantly improve overall network metrics such as throughput and delay by using ARQ procedures to deliver novel packets.

In order to make the most out of the MIMO_NC PHY, however, the CCSMA protocol described earlier has to be extended into the Phoenix solution. In particular, additional coordination among nodes is required to properly handle cooperative-network coded transmissions (called NC phases throughout the rest of this Section). First of all, observe that the success rate of a combined packet is lower than that of a pure cooperative retransmission. Therefore, if an NC phase is performed, the destination of the relayed packet (cooperative destination, that is to say D) experiences a degraded success rate. This higher packet error rate can be justified only if the destination of the novel frame (NC destination) has some chance to decode y (i.e., two successes with only one transmission are possible). Such a recovery is possible when the NC destination has cached an estimate of x. According to this remark, a cooperator shall perform a coded transmission only if the aforementioned condition is met. To this aim, Phoenix modifies the CCSMA contention strategy. Once a node receives a NACK asking for a retransmission and is eligible as a relay, it checks its own queue to determine if it has packets to code with the relayed frame. If not, the usual CCSMA behavior is followed, and if the node wins the contention a simple cooperative phase takes place. If the queue is non-empty, the potential cooperator checks the IDs of the destination for the NC packet and acts accordingly, distinguishing two conditions.

A) *NC and cooperative destinations coincide*. In this case, the node is sure that the addressee of its packet (D) has a cached version of x, although corrupted. [18] The usual CCSMA contention is then performed and if the terminal is elected as relay, a coded $x \oplus y$ frame is sent.

B) *NC and cooperative destinations are different*. The backoff phase proceeds as in CCSMA. If the node wins the contention, an RTS is sent. This is addressed to the destination of the relay's own frame and contains the ID of packet x. If the destination of the RTS receives it and has cached a version of x, a CTS is sent in reply. If the negotiation is successfully completed, the relay performs the NC transmission. Otherwise, the node switches back to the pure cooperative scheme and sends only x. We remark that the benefits offered by the RTS/CTS exchange are twofold. On one hand this procedure avoids useless and less effective NC phases, as discussed earlier. On the other hand, the collision avoidance strategy protects the relay communication because nodes that receive one of the handshake frames update their NAV as in the usual IEEE 802.11 approach [15]. This may be useful, as the destination of the NC packet could be located out of the regions blocked by the simple carrier sense mechanism for the original S-D communication.

If a pure relay transmission takes place, the cooperator, as discussed for the CCSMA protocol, does not need any feedback and goes back to its activity. On the contrary, if an NC phase was performed, the relay node has to know whether its own destination has decoded the packet or not. Therefore, Phoenix has to handle two potential ACK messages: one from the addressee of x and one from the addressee of y. To this aim, a simple slotted solution is employed. Once the reception of the data frame has ended, the destination of x starts sending its feedback (ACK/NACK) after a SIFS

[18] Because of the structure of MIMO_NC, also corrupted or redundant packets are cached and employed in the decoding process.

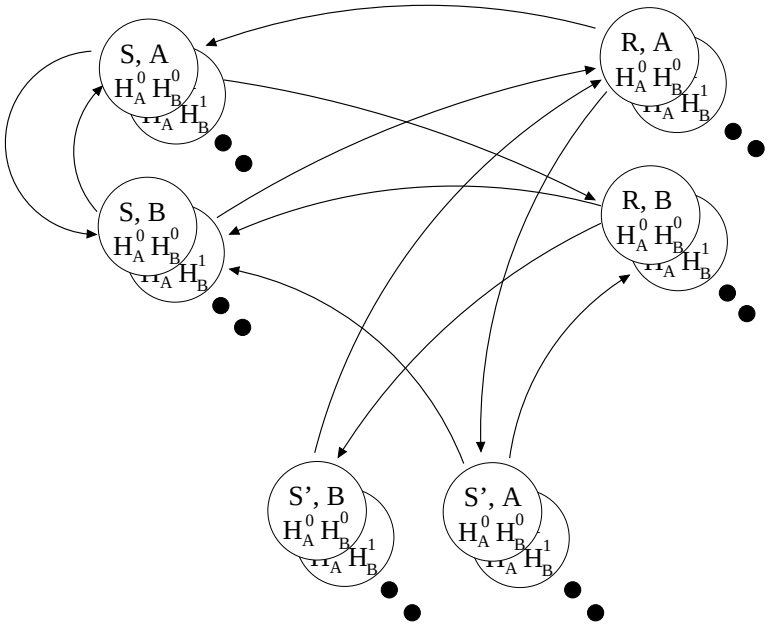

Figure 2.18. *Markov chain for modelling the protocols in correlated fading.*

interval, while the destination of y waits. After this transmission, if the secondary destination has successfully received its packet, a second ACK (addressed to the relay) is sent after SIFS seconds. Otherwise, the node keeps silent. In the event of a success, the relay goes back to its activities having processed one of its frames. If no ACK is received, the terminal puts packet y back into its queue and leaves the cooperative phase.

2.4.2 Analytical Model

In order to obtain some basic understanding on the possible gains that Phoenix can yield with respect to CSMA and CCSMA, we have developed a Markov model to compare the throughput of these protocols in a simplified environment. The test scenario is a three node network, composed by two saturated nodes A and B that send their packets to a central terminal T, which acts as a packet sink. Nodes A and B are at the same distance from T. Time is slotted, and only one node can transmit a single packet per slot. At the end of the slot, an instantaneous, error free ACK/NACK is sent to both terminals. The channel between the nodes A/B and T is assumed to be subject to time-correlated fading. Instead, the channel between A and B is error free. The correlation is modelled by a Markov chain [74], in which the SNRs of the A-T and B-T channels are tracked. The SNR is quantized into a set of values (from 0 to 18 dB, with step 1 dB, in our case), and the chain may transition only between neighboring states/SNRs. This is due to the fact that low speeds are considered and therefore transitions towards further states are unlikely. [19]

All the studied systems employ a slotted time frame and the medium access is scheduled by T. Nonetheless, the first protocol will be called CSMA, to be consistent with the other sections, and also

[19] More sophisticated models are available [75], however they are not needed in the low mobility environment considered here.

because it represents a benchmark protocol that uses plain, conventional ARQ techniques. In this case, if a node (say A) fails to deliver its packet, it transmits it again in the next slots (without any backoff) until the SRL is reached. In our model, the SRL is 2 (i.e., only one retransmission is allowed). In the second protocol, called CCSMA, if a transmission fails, B is assumed to have perfectly received A's packet and it performs the retransmission on behalf of A. No matter whether it succeeds or not, in the next slot the relay (B) transmits one of its own packets. Finally, in the third protocol, Phoenix, the relay node B transmits a coded packet which combines A's corrupted frame (which is again correctly received by the relay) and one of its packets. In the next slot, B transmits another packet of its own. This model makes some simplifying assumptions. The channel between the relays is error free, so the relay can always retransmit the packet on behalf of its neighbor. In addition, the relay is always automatically elected without any contention phase, at no cost in time. No outside interference can prevent the relay phase from acting. Therefore this environment represents an upper bound on the achievable performance of our protocols. Furthermore, the theoretical model is easy and quick to derive, it provides an insightful qualitative comparison of the three protocols and it suggests under which conditions Phoenix offers the highest benefits.

The network is described by a Markov chain for CSMA and a more elaborated one for CCSMA/Phoenix. In the former case, the state is described by two elements: a variable $X \in \{S, R, S'\}$ and the A-T channel SNR, H_A. Whenever a new packet is transmitted, $X = S$ if in the previous slot a correct transmission occurred; on the other hand $X = S'$ if 1) the previous slot experienced an error and no retransmissions are possible and 2) a new packet is sent. Instead, if a retransmission is performed, $X = R$. Let us explore the chain from state (S, H_A) (a new frame is sent after a successful transmission). If the packet is correctly received, the chain evolves into state (S, H'_A). Otherwise, the chain goes into state (R, H'_A) and a retransmission is performed. If successful, the chain moves into state (S, H''_A), otherwise into (S', H''_A). At this point, a new frame is transmitted and the chain then evolves into states (S, H'''_A) or (R, H'''_A), depending on whether the packet was correctly transferred. Flat, slow frequency fading is assumed, thus the transition probability is equal to the probability of correct/failed packet reception (if the state goes into states S or R/S', respectively) multiplied by the channel transition probability from H_A into H'_A. In order to compute the throughput, a suitable gain matrix must be defined. In this protocol it is enough to have a gain of 1 packet at any transition into state (S, H_A) and 0 otherwise.

For CCSMA and Phoenix, some modifications must be adopted (see Fig. 2.18). Now both A's and B's SNRs must be tracked, because they are both needed during a retransmission. The state is composed by four elements: $X \in \{S, R, S'\}$, the node that is transmitting (A or B) and the SNRs of the channels A-T and B-T, H_A and H_B. Let us start with a packet that is transmitted for the first time by, say, A. Then the chain is in state (S, A, H_A, H_B). If the packet is successfully received, then the chain transitions into state (S, B, H'_A, H'_B). Otherwise, a retransmission is performed, moving into (R, B, H_A, H'_B). Note that the state of channel A-T has not evolved, since memory of the first transmission must be kept. In this case, CCSMA retransmits A's frame. Instead, in Phoenix B will send a network coded combination of A's and one of B's packets. Node B will transmit an altogether new packet after this phase, but if the retransmission is successful, the chain moves into

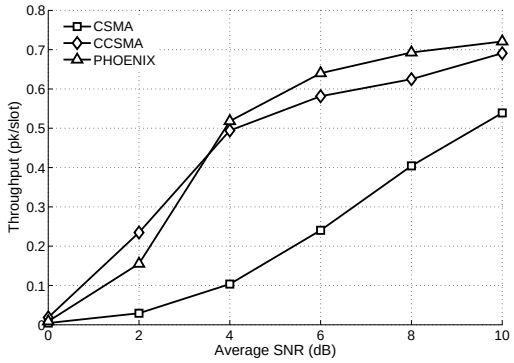

Figure 2.19. *Throughput vs average SNR (Markov analysis)*

state (S, B, H''_A, H''_B). If this is not the case, state (S', B, H''_A, H''_B) will be the new destination. Note that channel H_A has evolved by two steps into H''_A. If the arrival state is S' no reward is earned, while a reward of 1 (2) is won by CCSMA (Phoenix) if the retransmission gets across. If instead the model goes into state S from S' or S, the reward is just 1, since a simple, not NC-combined packet is received. In Phoenix, the transition probability from R into S is the probability that the packets are jointly decoded, multiplied by the channel transition probabilities.

The results obtained from the analysis of these Markov models are reported in Fig. 2.19, where the throughput is shown. First of all, Phoenix outperforms all protocols for sufficiently high SNRs. This is reasonable, since for very low mean SNR, two combined packets can seldom be decoded with just two transmissions. However, if the protocol is sufficiently well designed (for instance, relays with high SNR are picked), these situations can be avoided most of the time. On the other hand, Phoenix can achieve important improvements over CSMA and also has some advantage over the simple cooperative protocol. In an actual network deployment, the first transmitter always contacts one of its neighbors. Therefore, the average SNR will be in the high region, where Phoenix and CCSMA can achieve gains of 30% and 25% over CSMA. Finally, the cooperative protocols can be expected to increase the transmission range, since they achieve satisfactory performance at lower SNR.

2.4.3 Performance Evaluation

In this Section, we investigate the performance of the proposed protocols in more realistic network scenarios. The standard set of parameters reported in Table 2.2 is used, unless otherwise stated. For all results, the 95% confidence interval never exceeds 3% of the estimated value. Two scenarios were analyzed. In the former, Phoenix has been tested in networks with single hop flows and hidden terminals. In this case, the network is deployed in a $300\text{m}\times300\text{m}$ square, and there is enough spatial separation to allow multiple communications at any given time. This configuration tests the performance of the protocol when external interference and hidden terminals, representative of an actual network deployment, are present. Such a scenario stresses the impact of the different medium access policies.

The protocols have been tested also in a true multihop case, where packets must be forwarded. We have chosen a tree topology, where all nodes transmit their packets through multihop routes to a single sink. This scenario is well known in sensor networks, where data must be gathered by a sink. This setting is interesting because of the presence of several bottlenecks. In these places, packet losses are more frequent and they are particularly damaging since they affect all upstream nodes. In this case, MIMO_NC can be particularly useful because its throughput efficiency can help reduce the impact of these losses. In all cases, the wireless environment is subject to correlated Rayleigh fading. The Doppler frequency is equal to 70 Hz, corresponding to a speed of 5 m/s at 2.4 GHz. The initial maximum contention window is 128, because if a smaller window is selected, the collision probability in the 35 node network becomes so high that the protocols show very poor performance, and collisions simply hide all the phenomena we are interested in.

Finally, we also report some results on clustered networks, while a more extensive elavuation can be found in [9]. By clustered network we mean a wireless system where nodes have single hop connectivity to a gateway. Each gateway and the terminals connected to it (*cell members*) form a *cell*. This scenario finds important practical examples in cellular networks, Wireless LANs, some military networks (where low complexity nodes, like soldier radios, are directly connected to more sophisticated terminals, like Command & Control Centers, airplanes and so on), last mile connectivity or mesh networks (where mesh node communication may be supported by means of a different wireless technology than that of cell members). We have considered networks composed by N_c non-overlapping cells of radius 75 m; each of them is made up by a gateway (GW) located at the center and n additional nodes that generate single hop traffic flows addressed to the GW. The transmitters are randomly distributed within an annulus centered at the GW of inner and outer radii equal to 25 m and 75 m, respectively. Such a configuration has been chosen to reduce the number of nodes very close to the GW. CSMA with Binary Exponential Backoff is known to lead to severe unfairness, where nodes in favorable positions get the vast majority of the bandwidth. This situation is not interesting, because that would basically turn off peripherals terminals and leave only the central nodes active. Furthermore, only the uplink is studied, because NC retransmissions offer no clear advantage for downlink communications. In order to understand why, consider the case of a cell member A not successfully receiving a packet from the GW. Another cell member B that acted as relay would have no packet in queue for A, as all its traffic is addressed to the GW. Hence, in a clustered scenario, users within a cell are not able to exploit the coded retransmission mechanism of Phoenix. Moreover, even if the GW were allowed to act as a relay for itself and to perform a NC retransmission, the two coded packets would be sent by the same node (the GW), thus offering no spatial diversity. For these reasons, we believe that the downlink would benefit in no special way from NC retransmissions.

In our simulations, all the nodes in the network share the same bandwidth (i.e., universal frequency reuse).

The Short Retry Limit has been set to 3, a suitable value for delay constrained applications. In these conditions the network is heavy loaded, and transmissions tend to be affected by a high level of both intra-cell and inter-cell interference (as no frequency division multiplexing has been considered). This setting is critical for CSMA-based medium access policies, and represents a good

Table 2.2. *Parameters used in our simulations*

Transmission power	10 dBm
Noise Floor	-102 dBm
CS threshold	-100 dBm
Detection threshold	-96 dBm
Path loss exponent	3.5
Maximum Doppler shift	70 Hz (5 m/s)
Slot, DIFS, SIFS duration	20, 128, 28 μs
Data Rate	1 Mbit/s
Initial maximum contention window	128 slots
Short Retry Limit - CSMA, CCSMA, Phoenix	3, 2, 2
Number of slots used for relay contention, CW_{rel}	32
Simulation Time	30 s
Simulation Transient (metrics not collected)	10 s
DATA payload	2000 bits
DATA header CSMA - CCSMA	272 bits
DATA header Phoenix	280 bits
ACK/NACK/CTS	112 bits
RTS	160 bits

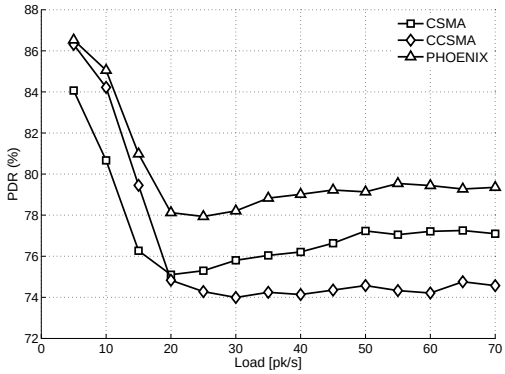

Figure 2.20. *Packet Delivery Ratio*

test for CCSMA and Phoenix, as the high number of packet losses is likely to often trigger cooperative procedures.

Single hop flows

In order to have a fair comparison among the systems, we want them to offer similar reliability. To this aim, we have tuned the SRL parameters of each protocol until a comparable Packet Delivery

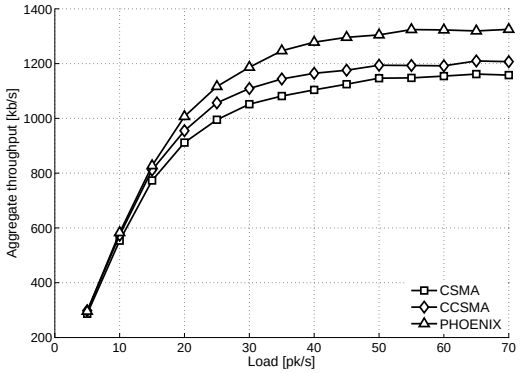

Figure 2.21. *Aggregate Throughput*

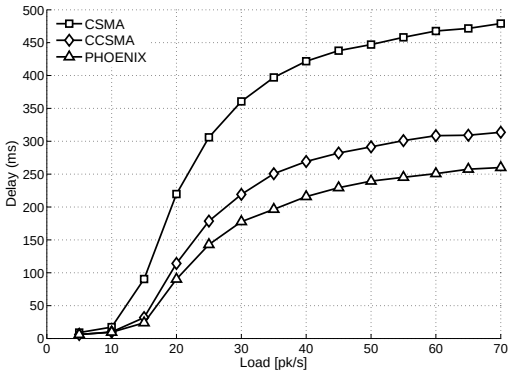

Figure 2.22. *System delay per acknowledged packet*

Ratio (PDR) was obtained. It turns out that CCSMA and Phoenix share the same SRL, while CSMA may require this limit to be larger. The PDR is shown in Fig. 2.20. It can be pointed out that Phoenix has a higher PDR than CCSMA. The reason is that the NC enhanced retransmission enables to safely deliver more packets by means of reliable relays. When a cooperative phase succeeds, CCSMA can increase the numerator of the PDR (i.e., number of acknowledged packets) by one unit. Instead, Phoenix increases it by two units and the denominator (i.e., the number of frames which are transmitted at least once) by one. Since the contention system often leads to a good choice of the relay, the retransmission is often successful, and in the end this leads to a higher PDR. The fact that the curves mildly rise for very high loads is due to the presence of nodes whose neighbors are very close; therefore their success rate is always very high. As the load increases, these terminals deliver most of the acknowledged traffic while the other nodes transmit fewer frames because their PDR is rather small and then they are stuck in backoff cycles. For these reasons, the weight of the first nodes in the PDR computation is larger: this induces the observed PDR rise. In perfectly symmetrical topologies (i.e., circular networks) such trend completely disappears.

Aggregate throughput is reported in Fig. 2.21. It is apparent that Phoenix yields a 15% improvement over plain CSMA and 10% over CCSMA. This plot shows that Phoenix is effective in recovering lost packets in heavy interference scenarios. On the other hand, in the previous Section, the model

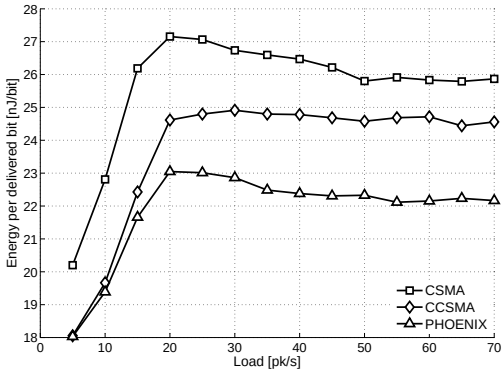

Figure 2.23. *Transmit energy consumption per information bit*

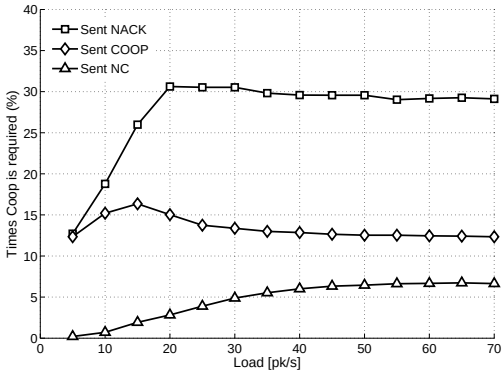

Figure 2.24. *Percentage of times that cooperative behavior is shown*

made some idealized assumptions about medium access and relay coordination, and therefore the quantitative results cannot be the same. However, the model predicts the qualitative behavior of the curves that can be found again in the simulation results; for instance the ranking of the three protocols is correctly foreseen. In conclusion, the model supports the choice of MIMO_NC as a suitable physical layer for high performance.

Another metric of interest is the average MAC delay (Fig. 2.22). Phoenix reduces delay at high loads by about 16% over CCSMA and 40% over CSMA. CSMA has a high delay because of the three retransmission phases, but the higher SRL is needed in order to have a comparable reliability with the other protocols. The key reason that enables a significant delay reduction by Phoenix over CCSMA and CSMA is the transmission of additional data packets by means of NC. These frames sent in an opportunistic fashion have much lower delay than the others. The reason is straightforward: they basically have no latency due to MAC contention procedures. Therefore, they are subject only to queueing delay. Since MAC contention delay can be rather larger, even a few of these frames can significantly improve the mean delay.

The last end-user metric we report is transmit energy consumption. This is computed as the total energy spent for packet transmissions divided by the total number of information bits successfully acknowledged. While CCSMA reduces this metric at high loads by about 5% with respect to CSMA,

Table 2.3. *Performance dependence on node density*

	nodes (AD)	CSMA	CCSMA	Phoenix
PDR (%)	25 (3.5)	76.2	73.2	75.1
	35 (5.5)	77.1	74.5	79.4
	56 (7.5)	75.5	72.5	74.9
Throughput (kb/s)	25	1010	1040	1105
	35	1160	1210	1330
	56	1308	1375	1465
Delay (ms/pk)	25	368	242	212
	35	480	310	260
	56	673	434	367
Energy (nJ/bit)	25	26.3	24.9	23.9
	35	25.8	24.6	22.2
	56	27.1	25.8	24.5
NACK/Coop/ /NC (%)	25			32.1/9.7/5.5
	35			29/12/6.6
	56			34/12.5/6

Phoenix leads to as much as 13%. This is due to the transmission of additional data packets during some retransmission phases at no cost in terms of bandwidth and transmission time.

Finally, Fig. 2.24 shows the percentage of data packets that require a cooperative retransmission (Sent NACK), how many contention phases are carried out by a relay (i.e., a relay has retransmitted the corrupted data packet, Sent COOP) and how many times MIMO_NC is actually used (Sent NC). Several conclusions can be inferred. Firstly, in 44% of the requested cooperative phases, some node actually performs the retransmission. The trace files have shown that in a large number of cases (40% of all requested retransmissions) there are no relays. This is partly due to topological reasons (there may be no relay close to source and destination) but in most cases the data packet or the NACK have not been received, because the relays may be decoding another packet in the meantime, or fading caused the packet loss. In the remaining 16%, there were relays, but carrier sense prevented them from transmitting. Two main conclusions can be drawn: first of all, interference does affect the proposed contention phase, especially preventing nodes from correctly decoding the lost data packet. On the other hand, further improvements could be attained by means of a more effective relay election or a system that may prevent the contention phases from being deserted (i.e., so that at least one node can actually contend for the role of cooperator). It is remarkable that the gains shown so far have been achieved by acting on only the cooperative packets, that is to say 16% of all the traffic. Hence, even if MIMO_NC is actually used only a few times, it delivers an interesting performance bonus.

The protocols behavior has been also studied varying the network density. Besides the reference case (35 nodes, $300\text{m}\times300\text{m}$), two more settings have been analyzed, with 25 or 56 nodes, in the

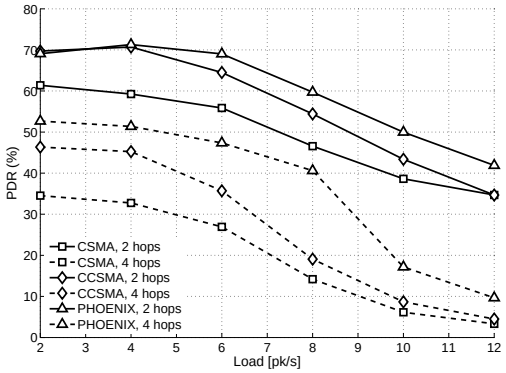

Figure 2.25. *Packet Delivery Ratio according to hop distance from the sink*

same area. The average number of neighbors (AD) is 3.5, 5.5 and 7.5 respectively. Table 2.3 reports the results for the metrics of interest. First of all, the relative performance is approximately the same for all configurations, showing that the system is robust to node density. However, this is a non trivial result, because node density affects two competing factors in Phoenix. More nodes imply higher interference and more collisions, which beset any protocol. On the other hand, a denser network offers more potential relays. By and large, it turns out that these factors balance themselves out, and thus Phoenix is a suitable choice for a wide range of scenarios.

Multihop scenario

We have also tested the protocols in multihop scenarios where 25 nodes are arranged in a tree topology. All terminals send their load towards the sink, and the network can be as deep as six hops. This setting is interesting because of the presence of bottlenecks, especially around the sink. In this case, any packet loss delays all the flows that converge into the bottleneck. Therefore, MIMO_NC can be very helpful in ”masking” these failures: if the packet is recovered together with the transmission of a new frame, the delay induced by the retransmission will play out less severely on the waiting flows. We show results up to a load of 12 pk/s, for which saturation arises. This remarks that severe congestion is present in this scenario, and a fast failure recovery procedure is essential.

In multihop networks, it is well known that the longer the hop count to the destination, the less satisfactory the performance. Therefore, we have analyzed PDR and throughput for nodes at different distances from the sink. Fig. 2.25 shows the PDR for nodes at 2 and 4 hops from the destination. It is clear that Phoenix delivers higher performance in all cases, especially for nodes farther from the sink, whose PDR can be twice as large for Phoenix as for CCSMA or CSMA. This is confirmed by Fig. 2.26, which shows that Phoenix outperforms all other protocols by as much as 30% in terms of throughput.

Fig. 2.27 analyzes the aggregate throughput for all the network. In this case the advantage of Phoenix is about 14% over CCSMA, lower than the 20-30% shown in Fig. 2.26. This is because that one hop neighbors of the sink have always a high throughput no matter what protocol is used (although the cooperative protocols do attain higher performance), and deliver the majority of the

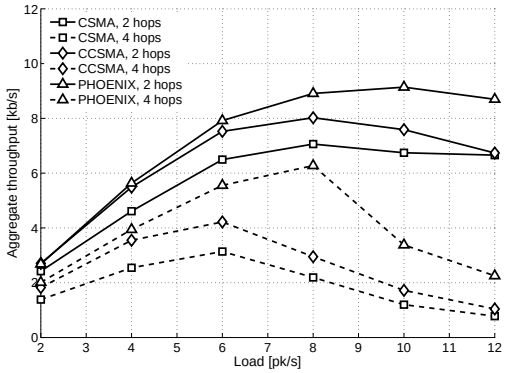

Figure 2.26. *Aggregate Throughput according to hop distance from the sink*

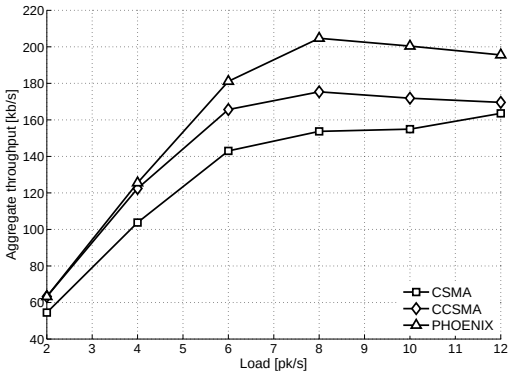

Figure 2.27. *Total aggregate Throughput, multihop scenario*

traffic. Therefore, CSMA and, to a lesser extent, CCSMA are rather unfair. Phoenix can offer more throughput to the network periphery than CSMA and CCSMA, and thus its gains over the other protocols are more significant than in the single-hop flow scenario. Phoenix is thus more fair, and therefore one can expect more throughput also at higher loads because the outer regions of the network deliver more traffic than with CCSMA. Fig. 2.28 reports Jain's fairness index and confirms exactly this idea: at higher loads, Phoenix's index is 10-12 percentage points higher than that of CCSMA, let alone CSMA, whose notorious unfairness shows up very early. We point out that our protocol has not been explicitly designed to be fair, and thus this important result is a desirable byproduct that comes for free.

Clustered Networks

Phoenix and the other protocols have been evaluated for many different configurations of clustered networks [9]. We report here the reference scenario, with with $N_c = 4$ cells and $n = 9$ nodes per cell (plus the Gateway GW). The Short Retry Limit has been set to 3, a suitable value for delay constrained applications. In these conditions the network is heavy loaded, and transmissions tend to be affected by a high level of both intra-cell and inter-cell interference (as no frequency division multiplexing has

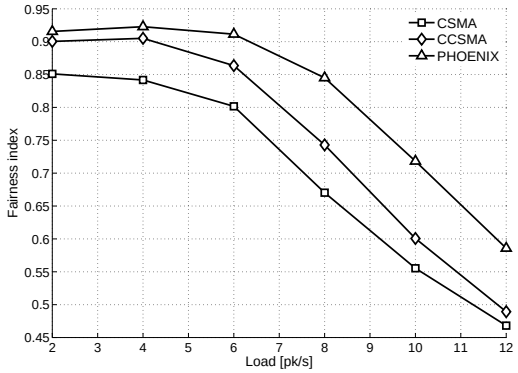

Figure 2.28. *Jain's fairness index, multihop scenario*

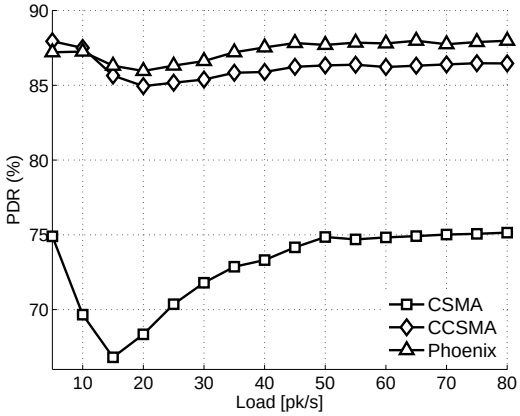

Figure 2.29. *Packet Delivery Ratio*

been considered). This setting is critical for CSMA-based medium access policies, and represents a good test for CCSMA and Phoenix, as the high number of packet losses is likely to often trigger cooperative procedures.

The first metric that we consider is the average Packet Delivery Ratio (PDR), defined as the ratio of the number of packets successfully received at the GWs to the number of packets injected in the network (i.e., transmitted at least once, either by means of a direct transmission or by means of an NC phase). The PDR is depicted against the nominal average generated load λ, in pk/s at every node, in Fig. 2.29. First of all, we notice that both CCSMA and Phoenix are able to significantly improve reliability with respect to CSMA when operated at the same SRL. The reason for this is that when a GW is not able to successfully receive a packet, CSMA can only rely on the temporal diversity provided by successive retransmissions in order to recover the failure. On the contrary, protocols that implement cooperative relaying are able to take advantage of spatial diversity as well, which strongly enhances the probability of successfully performing ARQ phases and thus improves the overall PDR. All the curves in Fig. 2.29 mildly rise for high loads. This effect is a result of the well known unfairness that affects CSMA-based protocols. In every cell, nodes that are closer to the GW experience a success ratio higher than the one of terminals that are farther away. As the load

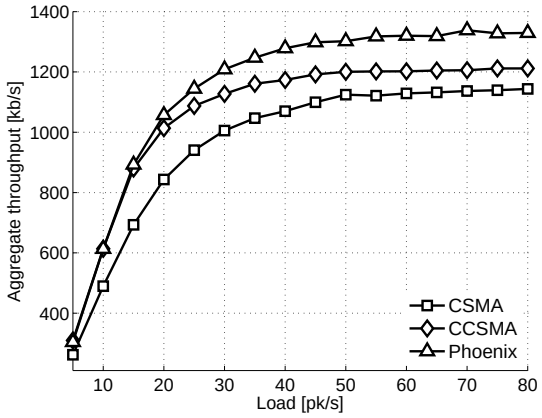

Figure 2.30. *Aggregate Throughput*

increases, the former nodes tend to access the channel more often, while the latter get stuck for longer periods in backoff cycles due to their lower success probability and deliver fewer packets. For these reasons, terminals closer to the GW have a larger weight on the overall metric computation, thereby inducing the observed PDR rise. It is interesting to remark that this effect is far less pronounced in CCSMA and Phoenix. Cooperative relaying helps in particular nodes that incur frequent packet losses, enhancing their success probability and thus shortening the number of backoff phases they have to undergo. Therefore, this technique is extremely beneficial to reduce unfairness. Finally, we notice that Phoenix yields a slight improvement over CCSMA (around 3% at high loads). This effect stems from the higher reliability of cooperative-NC phases.

Aggregate network throughput is depicted in Fig. 2.30 against λ. The plot shows that Phoenix outperforms CSMA by 16% and CCSMA by 10%. The improvement offered by our protocol is twofold. On the one hand Phoenix takes advantage of cooperative relaying in order to both increase the number of successfully delivered packets and to reduce the time required to perform a communication. On the other hand, cooperative transmissions can be exploited by nodes that act as relays to opportunistically deliver their own traffic without negotiating the channel. The combination of these two factors significantly increases the number of served and delivered packets.

In order to get further insight on the performance of CCSMA and Phoenix, it is interesting to consider the metrics depicted in Fig. 2.31: impact of cooperative phases and impact of NC phases. The former is computed as the ratio of the number of packets sent by relay nodes (either as pure cooperation or as NC combinations) to the number of NACKs sent out by the GWs asking for a retransmission, while the latter is the ratio of the times MIMO_NC is actually used to the same denominator. At low loads, cooperative phases take place with high probability when a retransmission has to be performed, proving the effectiveness of the contention scheme implemented by CCSMA and Phoenix. On the other hand, MIMO_NC is very rarely used, as nodes that act as relays are unlikely to have own packets to send to the GW. These two remarks explain both why at low loads the cooperative protocols are able to obtain interesting throughput gains with respect to CSMA and why Phoenix does not improve over CCSMA. As the load increases, the impact of cooperative phases

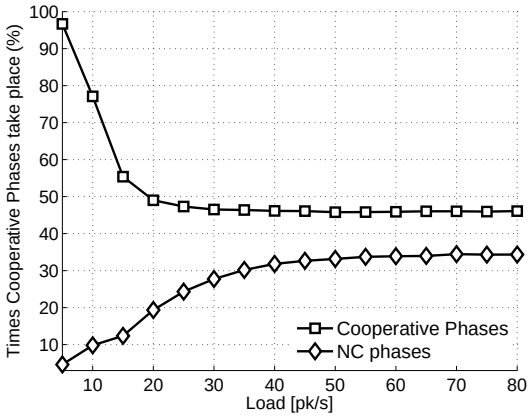

Figure 2.31. *Impact of Cooperation and Network Coding*

tends to decrease, stabilizing to 50%. This is due to the higher level of interference, that may on the one hand prevent some nodes from decoding the original packet or the NACK sent by the GW (thus reducing the number of potential cooperators) and on the other hand may induce relay candidates to erroneously leave the contention phase (see Section 2.4.1). The impact of NC phases, instead, increases with load as expected, because of the larger queue sizes. In saturation, a cooperative transmission involves a MIMO_NC encoded packet in the vast majority (75%) of cases. This behavior is reflected once again in Fig. 2.30: for high traffic rates Phoenix significantly outperforms CCSMA as the benefits of MIMO_NC become more and more important.

Another metric of interest is transmit energy consumption, presented in Fig. 2.32. The metric is computed as the total energy spent for packet transmissions divided by the number of successfully delivered information bits. The plot shows that Phoenix is more energy efficient than its competitors: at high loads, CSMA is outperformed by 20% while the improvement over CCSMA is as high as 10%. This stems from the capability of our protocol to exploit retransmission phases in order to deliver information packets at no additional cost in terms of energy and bandwidth. Two more observations can be made: first of all, the curves in Fig. 2.32 tend to decrease for higher loads, as an effect of the unfairness of CSMA-based protocols discussed earlier. Secondly, the gain of Phoenix over CCSMA is strengthened as traffic rate increases. This is due to the higher impact of NC phases on the cooperative mechanism that characterizes high load conditions (Fig. 2.31).

In our studies we have also investigated the behavior of the protocols with respect to network capacity. We define $\bar{\lambda}$ as the minimum value of λ that saturates the bandwidth B, i.e., $\bar{\lambda} n T_p = B$, where T_P is the minimum time to complete a data exchange (including overhead). Moreover, we call L the payload size and define $\bar{\tau} = \bar{\lambda} L$, the target throughput per node corresponding to this condition. In the reference scenario, $\bar{\lambda} \simeq 40$ pk/s per node and $\bar{\tau} = 80$ kb/s per node. We identify four classes of terminals with respect to QoS; the highest class contains nodes that achieve an average throughput τ higher than or equal to $\bar{\tau}$; the second class is for cell members that satisfy the constraint $2\bar{\tau}/3 \leq \tau < \bar{\tau}$; the third class groups terminals that obtain a throughput $\bar{\tau}/3 \leq \tau < 2\bar{\tau}/3$ and finally the lowest class includes nodes that do not reach a minimum target throughput equal to $\bar{\tau}/3$. The

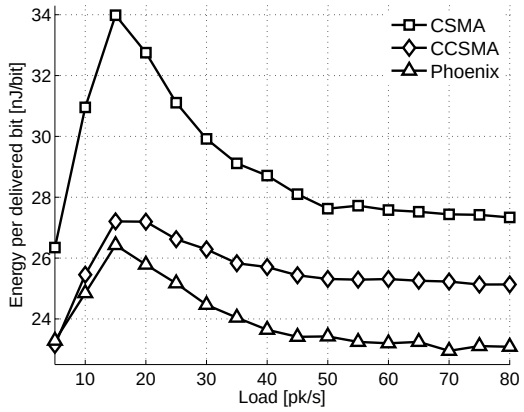

Figure 2.32. *Energy Consumption*

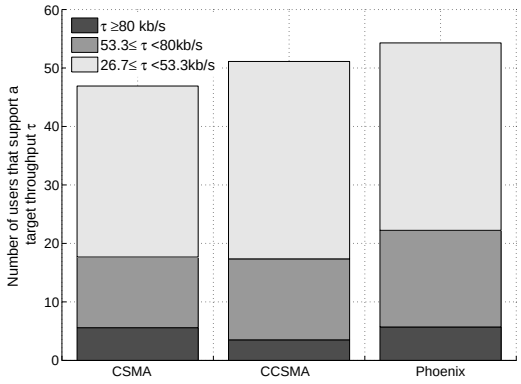

Figure 2.33. *Network Capacity in a 10 cells scenario*

results obtained at high load (60 pk/s) for the three highest classes are reported in Fig. 2.33. The improvement offered by Phoenix over its competitors is twofold. On the one hand, the number of cell members that support the minimum target throughput increases by 9% with respect to CCSMA and by 20% with respect to CSMA. On the other hand, our protocol boosts the number of nodes with medium and high QoS by as much as 35% if compared to the other medium access policies. These results show that the combination of cooperative relaying and network coding is able to guarantee a minimum service to a much higher share of cell members thanks to a more fair distribution of the resources. Moreover, MIMO_NC may be extremely beneficial for applications characterized by high QoS constraints, as networks that rely on Phoenix can support many more cell members with such requirements even in harsh interference conditions.

Let us now focus on the trends for CCSMA and CSMA in Fig. 2.33. Two behaviors can be observed: i) CCSMA increases the number of nodes that support the minimum throughput service, i.e., $\tau \geq \bar{\tau}/3$; ii) the cardinality of the two highest QoS classes for CCSMA is slightly lower than the one that characterizes CSMA. This offers an interesting insight on the impact of cooperative relaying techniques. In a decode-and-forward approach, relay nodes spend some of their resources in order to

help other terminals. In this way, not only do cooperators reduce their performance, but also nodes that benefit from relaying become more aggressive, as the enhancement of their success rate leads them to contend for the channel more often. Both these factors have a detrimental effect on cell members that are more likely to cooperate, i.e., those that are closer to their GW and that would normally enjoy high performance. We can then infer that cooperation redistributes the resources in the network at the expense of users with high QoS. This effect, on the contrary, does not affect Phoenix: the MIMO_NC scheme that we propose does not penalize terminals that decide to act as relays but rather it boosts their performance by letting them exploit cooperative phases to serve their own traffic. Thus, our approach overcomes an important limit of cooperative relaying, improving the performance of both high and low QoS classes.

2.4.4 Discussion

In this work we have applied for the first time the new MIMO_NC physical layer to a MAC protocol. In particular, we have enhanced cooperative techniques so as to deliver new packets together with retransmissions. This system encourages relays to cooperate, because they can help their neighbors without delaying their own traffic. It turns out that significant performance gains in terms of throughput, delay and energy consumption are possible, according to both analytical modelling and network simulations in a variety of contexts. On the other hand, interesting research directions are ahead. For instance, the possibility of transmitting multiple packets at the same time ushers new perspectives for routing. This includes the analysis of the impact of MIMO_NC in generic multi-hop networks as well as the development of a routing protocol specifically tailored to Phoenix. Finally, the performance analysis on clustered networks shown that interesting performance improvements in terms of throughput and especially capacity can be achieved for a wide range of design parameters. We have also found that Phoenix is particularly suited for high density networks with tight delay requirements.

2.5 Chapter Conclusions

This Section has explored how MIMO signal processing can be useful in several different, apparently unrelated areas. It is of course essential to properly design the physical layer of a MIMO ad hoc network, especially to balance the range of unicast and broadcast transmissions; it is also very helpful to make Network Coding more resilient to channel errors; finally its application to cooperative networks can relieve problems at the network level like congestion and spur nodes to cooperate.

2.6 Acknowledgments

The second part of this work (foundations of MIMO_NC) has been developed with Elena Fasolo, University of Padova. The last part (the Phoenix protocol) has been the result of a partnership with Elena Fasolo and Andrea Munari, both at the University of Padova.

References

[1] I. E. Telatar, "Capacity of multi–antenna Gaussian channels," *Eur. Trans. Tel.*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[2] G. J. Foschini and M. J. Gans, "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, Jun. 1998.

[3] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space–Time Wireless Communications*. Cambridge, UK: Cambridge University Press, 2003.

[4] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block coding from orthogonal designs," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1456–1467, Jul. 1999.

[5] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Trans. Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[6] D. Gesbert, M. Shafi, D. Shiu, S. P. J., and A. Naguib, "From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 281–302, Apr. 2003.

[7] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[8] S. N. Diggavi, A. S. N. Al-Dhahir, and A. R. Calderbank, "Great Expectations: The Value of Spatial Diversity in Wireless Networks," *Proc. IEEE*, vol. 92, no. 2, pp. 219–270, Feb. 2004.

[9] F. R. Farrokhi, A. Lozano, G. J. Foschini, and R. A. Valenzuela, "Spectral Efficiency of FDMA/TDMA Wireless Systems With Transmit and Receive Antenna Arrays," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 591–599, Oct. 2002.

[10] D. Gesbert, M. Kountouris, R. W. Heath Jr., C.-B. Chae, and T. Saelzer, "Shifting the MIMO Paradigm," *IEEE Signal Process. Mag.*, vol. 36, no. 9, pp. 36–46, Sep. 2007.

[11] S. Kittipiyakul and T. Javidi, "Optimal Operating Point for MIMO Multiple Access Channel with Bursty Traffic," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4464–4474, 2007.

[12] M. Levorato, P. Casari, S. Tomasin, and M. Zorzi, "Physical Layer Approximations for Cross-Layer Performance Analysis in MIMO-BLAST Ad Hoc Networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4390–4400, Dec. 2007.

[13] K. Sundaresan, R. Sivakumar, M. Ingram, and T.-Y. Chang, "Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 350–365, Oct. 2004.

[14] K. Sundaresan and R. Sivakumar, "Routing in Ad-Hoc Networks with MIMO Links," in *IEEE ICNP*, Boston (MA, USA), Nov. 2005.

[15] IEEE Standards Department, *ANSI / IEEE Standard 802.11*. IEEE Press, 1999.

[16] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," in *ACM MOBIHOC*, Long Beach (CA, USA), Oct. 2001.

[17] S. Bandyopadhyay, K. Hausike, S. Horisawa, and S. Tawara, "An adaptive MAC and directional routing protocol for ad hoc wireless networks using ESPAR antenna," in *ACM MOBIHOC*, Oct. 2001, pp. 243–246.

[18] T. Korakis, G. Jakllari, and L. Tassiulas, "A MAC protocol for full exploitation of directional antennas in ad hoc wireless networks," in *ACM MOBIHOC*, Annapolis, MD, Jun. 2003.

[19] T. ElBatt, T. Anderson, and B. Ryu, "Performance evaluation of multiple access protocols for ad hoc networks using directional antennas," in *IEEE WCNC 2003*, New Orleans (LA, USA), Mar. 2003.

[20] R. R. Choudhury and N. H. Vaidya, "Deafness: a MAC problem in ad hoc networks when using directional antennas," in *IEEE ICNP*, Oct. 2004.

[21] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "On designing MAC protocols for wireless networks using directional antennas," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 477–491, May 2006.

[22] M. Zorzi, J. Zeidler, A. Anderson, B. Rao, J. Proakis, A. L. Swindlehurst, M. Jensen, and S. Krishnamurthy, "Cross–layer issues in MAC protocol design for MIMO ad hoc networks," *IEEE Wireless Commun. Mag.*, vol. 13, no. 4, pp. 62–76, Aug. 2006.

[23] A. Lopez-Toledo and X. Wang, "TCP performance over Wireless MIMO Channels with ARQ and packet Combining," *IEEE Trans. Mobile Comput.*, vol. 5, no. 3, pp. 208–223, Mar. 2006.

[24] S. Y. Oh and M. Gerla, "MIMO and TCP: a case for cross layer design," in *IEEE MILCOM*, Orlando (FL, USA), Oct. 2007.

[25] C. A. Balanis, *Antenna Theory: Analysis and Design*, 2nd ed. New York: John Wiley and Sons, Inc., 1996.

[26] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *ACM MOBIHOC*, Long Beach, CA, Oct. 2001, pp. 87–94.

[27] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information Flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[28] H. L. van Trees, *Optimum Array Processing, Part IV*. New York: John Wiley and Sons, Inc., 2002.

[29] P. Mohapatra and S. Krishnamurthy (editors), *Ad Hoc Networks: Technologies and Protocols*. Cambridge (UK): Springer Verlag, New York (NY, USA), 2005.

[30] G. Jakllari, I. Broustis, T. Korakis, S. V. Krishnamurthy, and L. Tassiulas, "Handling Asymmetry in Gain in Directional Antenna Equipped Ad Hoc Networks," in *IEEE PIMRC 2005*, Berlin, (Germany), Sep. 2005.

[31] R. R. Choudhury and N. H. Vaidya, "Impact of Directional Antennas on Ad Hoc Routing," in *IFIP Personal and Wireless Communications (PWC)*, Venice (Italy), Sep. 2003.

[32] S. Roy, Y. C. Hu, D. Peroulis, and X. Y. Li, "Minimum-energy Broadcast Using Practical Directional Antennas in All-Wireless Networks," in *IEEE INFOCOM*, Barcelona (Spain), Apr. 2006.

[33] Y. Wang and J. Garcia-Luna-Aceves, "Broadcast Traffic in Ad Hoc Networks with Directional Antennas," in *IEEE GLOBECOM*, San Francisco (CA, USA), Dec. 2003.

[34] M. Levorato, S. Tomasin, P. Casari, and M. Zorzi, "Analysis of spatial multiplexing for cross–layer design of MIMO ad hoc networks," in *IEEE VTC-Spring*, Melbourne, Australia, May 2006.

[35] M. Hu and J. Zhang, "MIMO ad hoc networks: medium access control, saturation throughput, and optimal hop distance," *Journ. of Commun. and Networks, Special Issue on Mobile Ad Hoc Networks*, pp. 317–330, Dec. 2004.

[36] F. Rossetto and M. Zorzi, "On gain asymmetry and broadcast efficiency in MIMO ad hoc networks," in *IEEE ICC*, Istanbul (Turkey), Jun. 2006.

[37] O. Tirkkonnen, A. Boariu, and A. Hottinen, "Minimal Non-Orthogonality Rate 1 Space-Time Block Code for 3+ Tx Antennas," in *IEEE 6th ISSSTA*, Parsippany (NJ, USA), Sep. 2000.

[38] N. Sharma and C. B. Papadias, "Improved Quasi-Orthogonal Codes Through Constellation Rotation," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 332–335, Mar. 2003.

[39] M. Park, R. W. Heath Jr., and S. M. Nettles, "Improving Throughput and Fairness for MIMO Ad Hoc Networks Using Antenna Selection Diversity," in *IEEE GLOBECOM*, Dallas (TX, USA), Nov. 2004.

[40] T. Lo, "Adaptive Space-Time Transmission With Side Information," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1496–1504, Sep. 2004.

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

[42] J. Widmer and J.-Y. L. Boudec, "Network Coding for Efficient Communication in Extreme Networks," in *ACM SIGCOMM*, Philadelphia (PA, USA), Aug. 2005.

[43] J. Widmer, C. Fragouli, and J.-Y. L. Boudec, "Low–complexity energy–efficient broadcasting in wireless ad–hoc networks usign network coding," in *IEEE Information Theory Workshop*, San Antonio (TX, USA), Oct. 2004.

[44] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 49, no. 11, pp. 782–795, Nov. 2003.

[45] Y. E. Sagduyu and A. Ephremides, "Joint Scheduling and Wireless Network Coding," in *NetCod 2005*, Riva del Garda (Italy), Apr. 2005.

[46] 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 *INTERSENSE*, Nice (France), Apr. 2006.

[47] T. Ho, B. Leong, R. Koetter, and M. Medard, "Distributed Asynchronous Algorithms for Multicast Network Coding," in *Netcod*, Riva Del Garda, Italy, Apr. 2005.

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

[49] 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.

[50] P. A. Chou, T. Wu, and K. Jain, "Practical network coding," in *41st Allerton Conf. Communication, Control and Computing*, Monticello (IL, USA), Oct. 2003.

[51] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," in *IEEE INFOCOM*, Calcutta, India, Dec. 2005.

[52] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge (UK): Cambridge Univ. Press, 2005.

[53] E. Fasolo, F. Rossetto, and M. Zorzi, "Network Coding meets MIMO," in *NetCod 2008*, Hong Kong (China), Jan. 3-4 2008.

[54] S. Zhang, S. C. Liew, and P. P. Lam, "Physical-Layer Network Coding," in *ACM MOBICOM*, Los Angeles (CA, USA), Sep. 2006.

[55] P. Popovski and H. Yomo, "Physical Network Coding in Two–Way Wireless Relay Channels," in *IEEE ICC*, Glasgow (UK), June 2007.

[56] S. Yang and R. Koetter, "Network Coding over a Noisy Relay: a Belief Propagation Approach," in *IEEE ISIT*, Nice (France), 9 Jan. 2007.

[57] P. Larsson, N. Johansson, and K.-E. Sunell, "Coded Bi-directional Relaying," in *VTC Spring*, Melbourne (Australia), May 2006.

[58] S. Katti, I. Maric, A. Goldsmith, D. Katabi, and M. Medard, "Joint relaying and network coding in wireless networks," in *IEEE ISIT*, Nice (France), 24-28 June 2007.

[59] C. Hausl and P. Dupraz, "Joint Network-Channel Coding for the Multiple Access Relay Channel," in *IEEE IWWAN*, New York (NY, USA), June 2006.

[60] S. Zhang, Y. Zhu, S. Liew, and K. Ben Letaief, "Joint Design of Network-Coding and Channel Decoding for Wireless Networks," in *IEEE WCNC*, Hong Kong (China), 11-15 Mar. 2007.

[61] J.-H. Kim, S.-Y. Park, J. Y. Kim, Y.-J. Kim, and H.-Y. Song, "Joint ldpc codes for multi-user relay channel," in *NetCod 2008*, Hong Kong (China), Jan. 3-4 2008.

[62] L. Lv, H. Yu, and J. Yang, "Opportunistic cooperative network-coding based on space-time coding for bi-directional traffic flows," in *NetCod 2008*, Hong Kong (China), Jan. 3-4 2008.

[63] A. Montanari, V. Rathi, and R. Urbanke, "Iterative coding for network coding," in *ITA workshop 2008*, San Diego (CA, USA), Jan. 28-Feb 1 2008.

[64] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[65] H. Vikalo and B. Hassibi, "On Joint Detection and Decoding of Linear Block Codes on Gaussian Vector Channels," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3330–3342, Sep. 2006.

[66] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[67] E. Fasolo, F. Rossetto, and M. Zorzi, "The MIMO_NC project," University of Padova, Tech. Rep. 110, 2007.

[68] R. Blahut, *Theory and practice of error control codes*. Reading (MA, USA): Addison Wesley, 1983.

[69] J. Laneman, *Cooperation in Wireless Networks: Principles and Applications*. Springer, 2006, ch. Cooperative Diversity: Models, Algorithms, and Architectures, pp. 163–188.

[70] S. Moh, C. Yu, S.-M. Park, H.-N. Kim, and J. Park, "CD-MAC: Cooperative Diversity MAC for Robust Communication in Wireless Ad Hoc Networks," in *IEEE ICC*, Glasgow (UK), June 2007.

[71] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar, "CoopMAC: A Cooperative MAC for Wireless LANs," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 340–354, Feb. 2007.

[72] B. R. Hamilton and X. Ma, "Noncooperative Routing with Cooperative Diversity," in *IEEE ICC*, Glasgow (UK), June 2007.

[73] F. Librino, M. Levorato, and M. Zorzi, "Distributed Cooperative Routing and Hybrid ARQ in MIMO-BLAST Ad Hoc Networks," in *IEEE GLOBECOM*, Washington (DC, USA), Nov. 2007.

[74] H. S. Wang and N. Moayeri, "Finite state Markov channel - a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[75] C. Tan and N. Beaulieu, "On First-Order Markov Modeling for the Rayleigh Fading Channel," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2032–2040, Dec. 2000.

[76] A. Munari, F. Rossetto, and M. Zorzi, "On the viability of a Cooperative-Network Coding Protocol in Clustered Networks," in *IEEE MILCOM*, San Diego (CA, USA), Nov. 2008 (accepted).

Chapter 3

Interference models for Carrier Sense and Scheduling in ad hoc networks

Contents

3.1 Introduction 60
3.2 Related Work 61
3.3 A Robust approach to Carrier Sense for MIMO ad hoc networks 62
3.3.1 System Model and Protocol Description 63
3.3.2 The Carrier Sense System 65
3.3.3 Performance Evaluation 69
3.3.4 Discussion 71
3.4 A low complexity Model for Interference in Wireless Networks 72
3.4.1 Model Description 72
3.4.2 Performance Evaluation 79
3.4.3 Discussion 80
3.5 Dynamic Carrier Sense Adaptation 82
3.5.1 Dependence of the CS threshold on the node density 82
3.5.2 A Distributed Algorithm for Threshold Optimization 84
3.5.3 Performance Evaluation 86
3.5.4 Discussion 91
3.6 A Physical Model Scheduler for Multi-Hop Wireless Networks Based on Local Information 93
3.6.1 Scheduler Design 96
3.6.2 Implementation Issues 97
3.6.3 Performance Evaluation 100
3.6.4 Discussion 106
3.7 Chapter Conclusions 106
3.8 Acknowledgments 107
References 107

3.1 Introduction

One of the main approaches used to regulate medium access control in wireless networks relies is carrier sense (CS). In the wired world this idea coupled with Collision Detection (CD) proved to be so efficient to reduce collisions almost to zero, especially when the propagation delay is negligible with respect to the packet duration. On the other hand, in the wireless world CD would require full duplex radios on the same frequency. Unfortunately, these radios are not yet technically feasible, and so far IEEE 802.11 falls back on either a pure CSMA access control or a more sophisticated CSMA/CA (Collision Avoidance), which includes RTS/CTS handshake packets. CS tries to avoid simultaneous transmissions of interfering packets. This is particularly important in a scenario with a single access point (AP) and other nodes which exchange data with it. In this case, only a single station should be allowed to transmit, because a collision would occur otherwise.

However, if several APs are deployed in the same area, the question of whether multiple transmissions can occur simultaneously naturally arises. The key problem is to keep under control the level of mutual interference. According to this point of view, CS provides a tool to achieve this goal, because a terminal can assess if there is room for an additional transmission. The multiple AP scenario is representative of two types of networks which are receiving a great deal of attention: ad hoc networks (where every node is both a router and a traffic generator) and the Internet ”hot-spot” (the distribution of the Internet throughout an extended area). These systems must support the communications of multiple users at the same time and have to face the challenge of doing that in spite of scarce bandwidth. Spatial re-use is indeed a prime concern in this case, but without the technology of multiple antennas (which increases the capacity of the system by means of the separation of the signal spatial signatures) admission control is one of the main ways to control mutual interference. The Carrier Sense principle works towards this direction (at the MAC layer), because it avoids broadcasting new packets if the medium is already sensed to be busy.

A fundamental question is how to choose the power threshold that discriminates whether a node is allowed to send a packet or not. This threshold determines a tradeoff between spatial reuse and interference. Usually, this value is decided by means of simulation and real-world testing. There is no analytical method that may suggest a good value (not even roughly). The upshot is that in most cases this threshold is set to a low value, thus conservatively reducing bandwidth re-use and medium access. This eventually lowers the data rate and increases the delay, that affect the performance of the upper layers. For instance, TCP depends on the timely receptions of ACKs that may be inhibited in a wireless scenario because of an excessively conservative CS threshold.

The choice of the CS threshold may be either static or dynamic. In the former case, the network topology is known beforehand and the designer needs to find this optimal value. Such a scenario is important for static wireless networks whose environment (e.g., number of nodes and deployment area) changes very slowly in time, and hence an infrequent centralized optimization can be carried out off-line. Instead, if the environment is unknown beforehand or changes too often, the carrier sense threshold has to be found at run-time, possibly by the nodes themselves and in a distributed fashion.

An important contribution of this research branch is the development of an interference model for

these networks that can quickly predict the value of the carrier sense threshold that maximizes the aggregate throughput. This model is also very useful for the design at a low complexity scheduler in wireless multihop mesh networks, which will be studied in Section 3.6.

Moreover, all previous discussions implicitly referred to single antenna nodes. We have also explored special carrier sense systems for MIMO ad hoc networks which strike a balance between the spatial reuse allowed by multiple antennas and the necessity to keep interference under a tolerable level. This study works as an ideal bridge between Chapter 2 and Chapter 3.

3.2 Related Work

There has been some interest in the past for the performance optimization in CSMA ad hoc networks. In [1] the authors find, by simulation, the optimal value of the carrier sense threshold. However, a very simple interference carrier sense model was adopted: two nodes were assumed to interfere with each other only if they were closer than a certain range. While this is true for LOS scenarios and if only two nodes are present, this no longer holds in a more general propagation environment or in a situation with many nodes. In [2] and [3] the authors study the dependence of the optimal CS threshold of CSMA networks with respect to several parameters (e.g., protocol overhead or route length). [4] studied CS in a CDMA/CSMA network, and concluded that for these networks the carrier sense threshold could be lowered quite significantly with respect to a non-CDMA system, because the interference suppression capabilities of CDMA may reduce the minimum distance between two competing nodes and thus the value of the power threshold. Finally, [5] studied the problems of carrier sense on a real-world testbed. The conclusion was that the threshold is often very conservative and, if increased, may result in significant throughput improvements.

In addition, there have been some approaches in the past that have analytically modelled carrier sense in the IEEE 802.11 protocol. One of the first works has been [6], whose authors assume a disk model for carrier sensing (i.e., nodes whose distance is smaller than a certain sensing range R_{CS} cannot transmit simultaneously). In [7], the carrier sense model is still a disk, but the estimation of the fundamental parameters in carrier sensing (duration of a busy channel period and probability that at a generic instant of time the channel can be perceived busy) is based on accurate (though complex) graph models. The theoretical models well predict the simulation results. However, in neither of those two cases has the dependence of network performance on carrier sense been really studied. In [8], a CSMA protocol without binary backoff was studied. The importance of this study also lies in the interference model adopted: the aggregate interference is modelled by a Gaussian random variable, whose parameters depend on the network activity.

On the other hand, some work has analyzed the adaptation of the carrier sense threshold for ad hoc networks. For instance [1, 9] noted that the CS threshold is usually kept to a low value (close to the noise floor), so as to suppress neighboring interferers when a new transmission starts. This choice guarantees a high success probability, but reduces spatial reuse. They also proved that a more aggressive approach can bring significant performance improvements. [2] and [3] established that this study cannot ignore the impact of MAC layer overhead or adopted data rate. Our research has also

found that the optimal CS threshold (which enhances aggregate throughput) does depend on various parameters including in particular the density of nodes in the network. Moreover, [9] has proposed an interesting CS adaptation algorithm, and we will later compare our CS adaptation system with it.

3.3 A Robust approach to Carrier Sense for MIMO ad hoc networks

As the previous Chapter widely discussed, the introduction of multiple antennas and MIMO technology has spurred a significant deal of activity in the wireless ad hoc community, especially at the PHY/MAC layers [10]. In MIMO Ad Hoc networks, the additional spatial degrees of freedom can be used to improve performance, for instance increasing throughput. However, there is not yet general consensus on which PHY/MAC scheme can optimally exploit these additional resources. At least two approaches can be discerned in the literature. First, all the spatial degrees may be assigned to a single user; but this case would be a simple extension of the SISO case and does not require much redesign. However, this would not be an optimal use of the spatial degrees of freedom. Firstly, if a single user is assigned all degrees of freedom, not all the spatial channels may be reliable. In addition, the receiver needs to untangle the spatially multiplexed signals by non trivial signal processing [11] unless rather complete channel state information is available at the transmitter [12], which is however costly. This can be avoided by exploiting multi-user diversity along with the spatial degrees of freedom. Then the users can share the channels and all these bit pipes are likely to be of good quality by virtue of multi-user diversity. How to best accommodate multiple simultaneous transmissions is still an open question and several proposals exist [12–15].

A particularly interesting approach is the Opportunistic MAC (OMAC) protocol proposed in [15]. OMAC exploits multi-user diversity to ensure reliable channel quality and also full diversity for every user by antenna selection. An important issue is the correct estimation of the number of simultaneous transmissions in the neighborhood, by a generalization of carrier sense to MIMO. The performance evaluation in [15] assumed that this mechanism was error-free (i.e., it would always correctly estimate the number of concurrent communications). In this research branch, we develop a robust method to estimate this quantity, based on a rank estimation algorithm resilient to noise and interference [16]. Our key contribution is the design and performance evaluation of a novel carrier scheme tailored to MIMO that exploits the spatial structure of MIMO waveforms. Incidentally, the proposed carrier sense system used here is by no means specific to OMAC, and could also be adopted by other MIMO ad hoc protocols. The other MIMO MAC protocols based on carrier sense may use the conventional approach (i.e., compare the average received power across the antennas against a certain threshold) that does not exploit this structure [**?**, 13, 14, 17, 18], or may propose ad hoc techniques that work only when transmissions are slot synchronous [12], while our scheme works also for asynchronous communications. We will show that this carrier sense mechanism can untap the communication parallelism inherent in MIMO at a limited cost in computational complexity.

3.3.1 System Model and Protocol Description

System Model

We investigate a single hop wireless ad hoc network without hidden terminals composed of N nodes with M antennas each. Consider a typical transmission between a transmitter node t and a receiver node r in the presence of N_I interferes. The channel matrix between t and r is H, where $H \in \mathcal{C}^{M\times M}$ and its elements are independent and identically distributed (i.i.d.), zero-mean, complex Gaussian random variables $\sim \mathcal{CN}(0,1)$. Let H_{yx} denote the channel coefficient between the transmit antenna x and the receive antenna y and let $\mathbf{h}_x$ be the x-th column of H. Note that $\mathbf{h}_x$ is the set of channel coefficients between the x-th transmit antenna and the receiver array. All transmitters use selection diversity. In selection diversity, the transmitter node sends its packet using the antenna a_t which corresponds to the column with the maximum Frobenius norm:

$$a_t = \arg\max_x ||\mathbf{h}_x||_F^2 = \arg\max_x \sum_{y=1}^{M} |H_{yx}|^2 \tag{3.1}$$

Let the vector corresponding to the maximum norm be denoted by $\mathbf{h}_{a_t}$, let a transmitter i send a symbol s_i using transmit antenna a_i and the average received power from i be p_i. Assuming that the fading is constant over the observed time window (block fading model), the aggregate signal at the receiver r at time T is:

$$\begin{aligned} \tilde{\mathbf{Y}}_{r,T} &= \mathbf{h}_{a_t}\sqrt{p_t}\, s_{t,T} + \sum_{i=1}^{N_I} \mathbf{h}_{a_i}\sqrt{p_i}\, s_{i,T} + \mathbf{W}_{r,T} = \\ &= \mathbf{h}_{a_t}\sqrt{p_t}\, s_{t,T} + \mathcal{I}_{r,T} \end{aligned} \tag{3.2}$$

where $\mathbf{W}_{r,T}$ is a $M \times 1$ vector of complex, zero mean, circularly symmetric Gaussian random variables with variance N_0, which represent the thermal noise at r. It stems that the interference covariance matrix $\tilde{R}_I$ is:

$$\tilde{R}_I = \mathbf{E}\left[\mathcal{I}_{r,T}\mathcal{I}_{r,T}^H\right] = \sum_{i=1}^{N_I} p_i \mathbf{h}_{a_i}\mathbf{h}_{a_i}^H + N_0\mathbf{I} \tag{3.3}$$

At the intended receiver r, the received signal $\tilde{\mathbf{Y}}_{r,T}$, is pre-multiplied by the optimum MMSE combining weights, given by [19]:

$$\tilde{R}_I^{-1}\mathbf{h}_{a_t} \tag{3.4}$$

Note that the above equations assume the signals to be symbol-synchronous. This is by no means restrictive, as a simple example can show. Let us assume with no loss in generality that just two users were transmitting, but their waveforms were not symbol-synchronous at a certain receiver and the relative delay were τ. Let us also assume that the receiver had acquired timing synchronization with the first one. The aggregate signal would be:

$$\begin{aligned}\tilde{\mathbf{Y}}_{r,T} &= h_{a_2} r_{ss}(\tau)\sqrt{p_2}\, s_{2,T} + h_{a_2} r_{ss}(\mathcal{T}-\tau)\sqrt{p_2}\, s_{2,T+1} \\ &+ h_{a_1}\sqrt{p_1}\, s_{1,T} + \mathbf{W}_{r,T}\end{aligned}$$

where $\mathcal{T}$ is the symbol duration and $r_{ss}(\tau)$ is the autocorrelation function of the shaping pulse. After some algebra, the autocorrelation matrix is:

$$\begin{aligned}\tilde{R}_I &= p_2 h_{a_2} h_{a_2}^H (r_{ss}^2(\tau) + r_{ss}^2(\mathcal{T}-\tau)) + \\ &+ p_1 h_{a_1} h_{a_1}^H + N_0\mathbf{I}\end{aligned} \tag{3.5}$$

Eq. (3.5) proves that the presence of asynchronous users is equivalent to a change in the received power, but the rank is preserved.

Protocol Description

This section briefly describes the OMAC protocol proposed in [15]. It is based on the IEEE 802.11 Distributed Coordination Function (DCF) with MIMO-adapted carrier sensing to enable asynchronous simultaneous transmissions. OMAC ensures that with high probability the maximum number of concurrent transmissions is not bigger than the number of antennas on each node.

If the received samples $\tilde{\mathbf{Y}}_{r,T}$ were noiseless, the interference correlation matrix R_I would be:

$$R_I = \sum_{i=1}^{N_I} p_i \mathbf{h}_{a_i} \mathbf{h}_{a_i}^H \tag{3.6}$$

Its rank is the number of transmitters currently active in the network. In Section 3.3.2 we will describe a noise robust method to estimate the rank of R_I from $\tilde{\mathbf{Y}}_{r,T}$ and its effects on the protocol. One of the design principles of OMAC is to allow up to k_M concurrent transmissions in the network, that is to say, up to k_M spatial degrees of freedom should be used. Since these degrees are M, it stems that $k_M \leq M$. A large k_M enhances the spatial reuse, but also reduces the margin to withstand unusual levels of interference (generated, for instance, by terminals outside the network or incorrect carrier sense decisions).

The protocol consists of the following three important stages: rank determination, transmission decision, antenna selection. In the following, it is assumed (as in [15]) that the sensing node can correctly estimate the rank of R_I. The next Section will show how to perform an actual, effective estimation.

1: **Rank Determination:**
The transmitter calculates the rank of R_I, $Rank(R_I)$. The rank depends on the number of interferers active in the network.

2: **Transmission Decision:**

1. If $Rank(R_I) = X$ where $0 \leq X < k_M$, the sender can transmits an RTS packet if the channel state satisfies $Rank(R_I) < k_M$ for a period of Distributed Inter Frame Space (DIFS) duration.

2. $Rank(R_I) = k_M$ represents full rank of R_I which, in OMAC, is considered as the busy state of the channel and the sender must now enter the binary exponential back-off stage.

3. During the back-off stage, if the channel state changes from $Rank(R_I) < k_M$ to $Rank(R_I) = k_M$, then the backoff counter is frozen, as in CSMA, until $Rank(R_I)$ becomes less than k_M. Once $Rank(R_I) < k_M$ is satisfied, the backoff counter decrementing process is continued. When the backoff counter reaches zero, the sender begins its transmission based on the antenna that was used to perform the last communication with the designated receiver.

4. If the condition $Rank(R_I) < k_M$ is satisfied at the receiver, this destination replies with a CTS.

3: **Antenna selection:**
For the receiver-transmitter pair (r, t), the index of the best antenna according to the selection diversity criterion (3.1) is fed back in the CTS packet by the receiver r.

The RTS/CTS packets are sent using Space Time Block Codes. The details of the employed codes are given in [15].

3.3.2 The Carrier Sense System

Estimation of the number of active users

The carrier sense mechanism starts from the analysis of Eqs. (3.3) and (3.6). Eq. (3.6) suggests that the rank of the correlation matrix of the interference is the number of transmitting terminals. However, the correlation matrix $\tilde{R}_I$ (3.3) estimated from the received samples (3.2) would always have full rank because of the white noise. Without further elaboration, noise would thwart any computation of the number of users based on $\tilde{R}_I$.

The MIMO carrier sense mechanism that we propose employs a robust rank estimation method based on the Singular Value Decomposition (SVD) [16]. Let us consider a set of n consecutive time slots. Without loss of generality, let the first time slot be $T = 0$. The terminal collects $M \times n$ samples, $n \geq M$, out of its M antennas:

$$\widetilde{\mathcal{Y}} = \left[\tilde{\mathbf{Y}}_{r,0}\, \tilde{\mathbf{Y}}_{r,1},\, ...\, \tilde{\mathbf{Y}}_{r,n-1}\right] \tag{3.7}$$

The estimation of the number of users is based on $\widetilde{\mathcal{Y}}$ according to the following two important logical steps:

1. The singular values of $\mathcal{Y}$ and $\tilde{R}_I$ are tightly linked;

2. The singular values of $\tilde{R}_I$ can indicate the rank of R_I

As far as the first point is concerned, note that the spectrum of $\tilde{R}_I$ is $\{\widetilde{\sigma}_1^2, \cdots, \widetilde{\sigma}_M^2\}$, while the singular values of $\widetilde{\mathcal{Y}}$ converge to $\{\widetilde{\sigma}_1, \cdots, \widetilde{\sigma}_M\}$ as $n \to +\infty$. For finite n, the second part of this section will prove that the approximation is very accurate for $n \geq 20$.

Let us define $\mathcal{W}$ as the noise that corrupts $\widetilde{\mathcal{Y}}$, $w_{i,j}$ the element of $\mathcal{W}$ in the i-th row and j-th column and $\mathcal{Y}$ as the observation $\widetilde{\mathcal{Y}}$ devoid of noise: $\mathcal{Y} = \widetilde{\mathcal{Y}} - \mathcal{W}$. The SVD decomposition of $\widetilde{\mathcal{Y}}$ is:

$$\widetilde{\mathcal{Y}} = \widetilde{\mathcal{U}} \left(\widetilde{\Sigma}\, 0_{n-M,\,M} \right) \widetilde{\mathcal{V}}^H \tag{3.8}$$

$$\widetilde{\Sigma} = \text{diag}\,(\widetilde{\sigma}_1, \cdots, \widetilde{\sigma}_M) \tag{3.9}$$

where $\widetilde{\mathcal{U}}$ and $\widetilde{\Sigma}$ are $M \times M$, $0_{n-M,\,M}$ is the $n - M \times M$ null matrix and $\widetilde{\mathcal{V}}$ is $n \times n$. Let us also call $\{\sigma_1^2, \cdots, \sigma_M^2\}$ the singular values of R_I. Note that the number of non zero $\{\sigma_1^2, \cdots, \sigma_M^2\}$ is the number of active users (Eq. (3.6)). [1]

A precise statement about the second point is given by the Schmidt theorem [16]:

Theorem 1 (Schmidt)**.** *Let $\mathcal{Y}$ and $\widetilde{\mathcal{Y}}$ have the same meaning as before. Let us also define the thresholds $\widetilde{\tau}_k^2$, $0 \leq k \leq M$ as the sum of the square of the $M - k$ smallest singular values of $\widetilde{\mathcal{Y}}$:*

$$\widetilde{\tau}_k^2 = \widetilde{\sigma}_{k+1}^2 + \cdots + \widetilde{\sigma}_M^2 \tag{3.10}$$

Then the rank of $\mathcal{Y}$ is k if and only if:

$$\widetilde{\tau}_k^2 \leq ||\mathcal{W}||_F^2 < \widetilde{\tau}_{k-1}^2 \tag{3.11}$$

where $\|\mathcal{W}\|_F^2$ is the square Frobenius norm of $\mathcal{W}$.

Note that Eq. (3.11) makes a statement on $\mathcal{Y}$ by means of $\widetilde{\mathcal{Y}}$'s singular values. However, this theorem is still impractical since it involves the Frobenius norm of the noise, which is unknown. The final step is to notice that the statistics of $||\mathcal{W}||_F^2$ is known: it is a chi-square random variable with $2nM$ degrees of freedom, because the real and imaginary parts of $w_{i,j}$ are normal random variables with mean 0 and variance $N_0/2$ [19], and the CDF $F(x; 2nM)$ is:

$$F(x) = 1 - e^{-x/N_0} \sum_{i=0}^{nM-1} \frac{(x/N_0)^i}{i!} \tag{3.12}$$

Hence, two approaches are possible. In the first one, $||\mathcal{W}||_F^2$ is approximated by its expected (and deterministic) value nMN_0. Hence, the rank k is the only value of k such that:

$$\widetilde{\tau}_k^2 \leq nMN_0 < \widetilde{\tau}_{k-1}^2 \tag{3.13}$$

In the second strategy, $||\mathcal{W}||_F^2$ is a random variable with CDF given in (3.12). To each event $[\widetilde{\tau}_k^2 \leq ||\mathcal{W}||_F^2 < \widetilde{\tau}_{k-1}^2]$ is associated a probability equal to:

$$\begin{aligned} \Pr[\text{users} = k] &= \Pr[\text{rank} = k] = \Pr\left[\widetilde{\tau}_k^2 \leq \|\mathcal{W}\|_F^2 < \widetilde{\tau}_{k-1}^2\right] \\ &= F\left(\widetilde{\tau}_{k-1}^2\right) - F\left(\widetilde{\tau}_k^2\right) \end{aligned} \tag{3.14}$$

[1] Because of its conceptual simplicity and insight, here we deal with the SVD based technique. However, [16] proposes other algorithms based on computationally less expensive decompositions, like the QR. We discuss complexity issues for the SVD method later in this section.

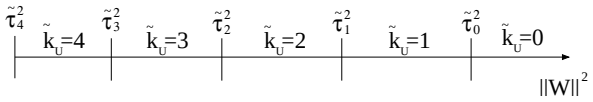

Figure 3.1. *Pictorial representation of the estimation process as a function of the thresholds and the Frobenius norm of the noise*

where $\widetilde{\tau}_{-1}^2 = +\infty$. We have used the latter method because it exploits a richer information than just the expected value of the noise and the involved computations can still be carried out with relatively low complexity. Last but not least, it provides some gain in terms of performance.

The MIMO carrier sense system proposed in [15] integrates this mechanism as follows: the node computes the number of active nodes $\tilde{k}_U$ as the rank of R_I estimated from $\widetilde{\mathcal{Y}}$. The node transmits only if $\tilde{k}_U$ is strictly smaller than a maximum number of allowed ongoing communications k_M. Note that $\tilde{k}_U$ cannot be larger than M, while the number of ongoing communications can be larger than that. However, from the point of view of medium access control, this is not important, because in both cases all spatial degrees of freedom have been used and there is no room for more simultaneous transmissions. In addition, the proposed carrier sense approach is robust to noise not only because the algorithm in [16] is so, but also because the node need not to correctly estimate the number of active users, but just whether k_U is smaller or larger than k_M. For instance, if $k_U = 3$ and $k_M = 3$, both $\tilde{k}_U = 3$ and $\tilde{k}_U = 4$ lead to the proper MAC decision of deferring the transmission.

Let us consider as an important example a system with $M = 4$ antennas. The thresholds τ_i^2 are:

$$\begin{aligned}
\widetilde{\tau}_0^2 &= \widetilde{\sigma}_1^2 + \widetilde{\sigma}_2^2 + \widetilde{\sigma}_3^2 + \widetilde{\sigma}_4^2 \\
\widetilde{\tau}_1^2 &= \widetilde{\sigma}_2^2 + \widetilde{\sigma}_3^2 + \widetilde{\sigma}_4^2 \\
\widetilde{\tau}_2^2 &= \widetilde{\sigma}_3^2 + \widetilde{\sigma}_4^2 \\
\widetilde{\tau}_3^2 &= \widetilde{\sigma}_4^2 \\
\widetilde{\tau}_4^2 &= 0
\end{aligned}$$

The rank is the number of the elements in $\left[\widetilde{\tau}_3^2\ \widetilde{\tau}_2^2\ \widetilde{\tau}_1^2\ \widetilde{\tau}_0^2\right]$ that are bigger than the square Frobenius norm of the error matrix $\mathcal{W}$. A pictorial representation is given in Fig. 3.1.

We point out that the singular values $\sigma_i^2, 1 \leq i \leq M$ can be estimated either from $\widetilde{\mathcal{Y}}$ or from an estimate of the $M \times M$ matrix $\tilde{R}_I$, which can be computed as $\widetilde{\mathcal{Y}}\widetilde{\mathcal{Y}}^H/n$, where $\widetilde{\mathcal{Y}}^H$ is the hermitian of $\widetilde{\mathcal{Y}}$. We prefer the former for two reasons. First of all, the ratio of the largest and smallest singular values of $\widetilde{\mathcal{Y}}\widetilde{\mathcal{Y}}^H/n$ is the square of the same quantity computed for $\widetilde{\mathcal{Y}}$ [19], which implies that the results are less accurate in finite precision arithmetic. Then, it is not convenient even from a computational point of view. The cost to evaluate an SVD for an $M \times n$ matrix is $O(n^2M + nM^2)$ [19], but computing $\widetilde{\mathcal{Y}}\widetilde{\mathcal{Y}}^H/n$ requires M^2n^2 operations. Hence, the computational complexity to operate on $\widetilde{\mathcal{Y}}$ is $O(Mn^2 + M^2n)$, while $\widetilde{\mathcal{Y}}\widetilde{\mathcal{Y}}^H/n$ entails $O(M^3) + O(n^2M^2)$. A quick comparison shows that it is less expensive to figure out the rank from $\widetilde{\mathcal{Y}}$ than from $\widetilde{\mathcal{Y}}\widetilde{\mathcal{Y}}^H/n$.

Another important consideration about the SVD-based rank estimation method is that it works very well if the Signal to Noise Ratio (SNR) is strong enough. We will show how to tune the carrier

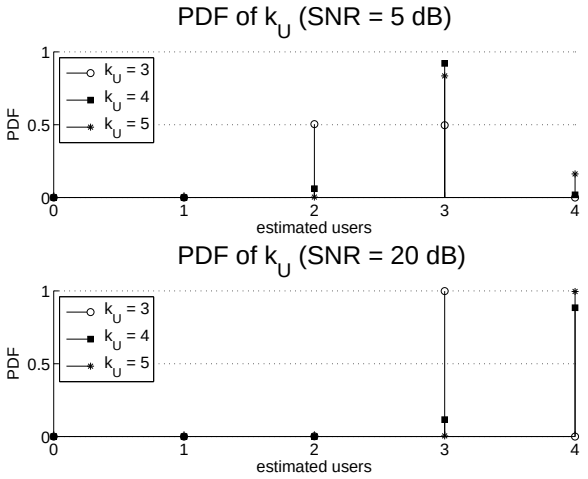

Figure 3.2. *Mass Distribution function of the number of estimated users*

sense parameters according to the SNR so as to avoid noticeable performance degradation in the rank estimation.

Performance evaluation

The rank estimation is the core of our MIMO Carrier Sense method, and valuable insight can be gained by studying how its performance depends on different parameters. The independent variables that have been considered are the SNR, the number of samples n used for the estimation and the number of active users k_U. It is reasonable to expect that the larger n, the more reliable $\tilde{k}_U$. Moreover, the probability of incorrectly evaluating k_U should approach an asymptotic value for $n \to +\infty$.

In this setting, the sensing node has to estimate k_U and is equipped with $M = 4$ antennas. The average SNR between any transmitter and the sensing node is the same and each sender uses one antenna. The channel model is the one described in Section 3.3.1. Each scenario was run 10000 times so as to achieve the necessary statistical confidence.

Fig. 3.2 shows the mass distribution function of $\tilde{k}_U$ when the SNR is 5 or 20 dB, $n = 100$ and $k_U \in \{1, 2, 3, 4, 5\}$. At low SNR the estimation is reliable only with few transmitters (1 or 2, not shown in the figure), and if more of them are present, the method underestimates k_U. With $k_M = 4$ (the system is allowed to saturate the spatial degrees of freedom) and 5 active users, the method yields $\tilde{k}_U = 3$ in most (80%) of the cases. At high SNR the estimation is dependable.

Setting $k_M = 4$, a missed detection is defined as the event that $\tilde{k}_U < 4$ given that $k_U \geq 4$. Fig. 3.3 shows the probability of missed detection versus the length of the data sample, for different SNR values. For each SNR, as k_U increases, it is more likely to detect that the system is saturated. Moreover, the missed detection probability decreases with n, approaching an asymptotic value, as predicted. Note that usually $n = 40$ is enough to get very close to this minimum error probability. Hence n does not have a major influence on the system performance unless it is very small (less than 20 samples). Finally, at low and medium SNR the method is unable to reliably detect the saturation condition, while at high SNR (20 dB) it correctly detects channel saturation.

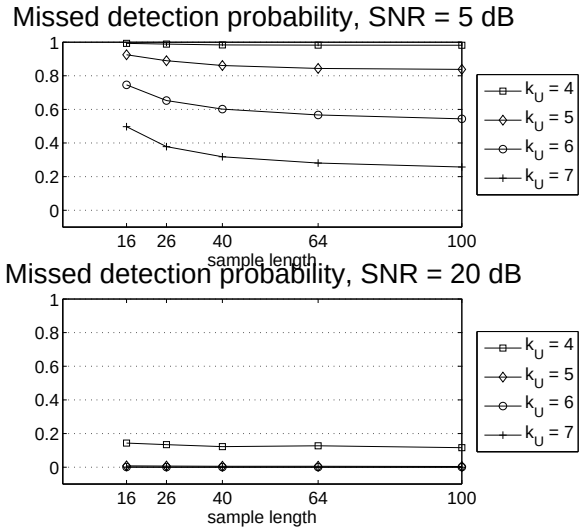

Figure 3.3. *Probability of missed detection for* $k_M = 4$

Due to the poor performance at low and medium SNR for $k_M = 4$, a more conservative approach with $k_M = 3$ has been explored. This choice of k_M creates a margin of safety against estimation errors in $\tilde{k}_U$. In addition, Fig. 3.4 points out that fewer missed detections occur for $k_M = 3$ than for $k_M = 4$. The two different choices are compared in Fig. 3.5.

In conclusion, we note that $k_M = 3$ is a safe choice for this parameter, since it effectively suppresses decision errors at the MAC layer and still untaps the majority of the spatial degrees of freedom. For low SNR, $k_M = 4$ is unsuitable for two reasons. First of all it misleads the MAC layer decision process quite often. Moreover, at low SNR the degrees of freedom are fewer than $M = 4$ because the system is noise limited rather than interference limited. Hence, lower spatial reuse should be sought, since the SINR would be unacceptably low if too many concurrent communications were allowed. Finally, while we analyze protocols with a fixed value of k_M, it is easy to envision an adaptive protocol that would tune it according to the system parameters, chiefly the SNR.

3.3.3 Performance Evaluation

The protocol performance was tested in a single-hop network, with 20 nodes uniformly distributed in a $20\,\text{m} \times 20\,\text{m}$ area, half of them labeled as sources and the other half as destinations. Nodes are paired in couples, and all couples are disjoint (i.e., they share neither the source nor the destination). The simulation parameters are listed in Table 3.1 and the transmit power was varied to achieve different average SNRs (10, 20, 30 and 40 dB). The sources generate Poisson traffic with rate 2, 4, 8, 16, 32 or 64 packets per second. A sufficient number of simulations has been run to achieve the desired statistical confidence.

Unlike [15], the rank of the interference autocorrelation is not assumed to be known and needs to be estimated from the received samples. Fig. 3.6 shows the throughput when up to 4 simultaneous transmissions are allowed ($k_M = M = 4$, i.e., all degrees of freedom can be used), for different values of the SNR. In addition, the ideal situation of perfect rank estimation (as in [15]) is reported

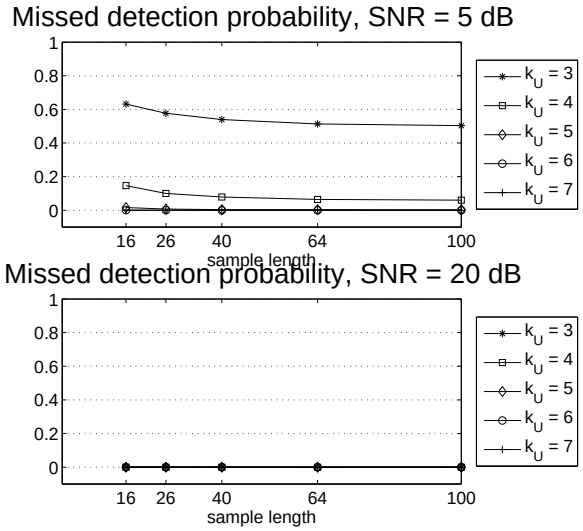

Figure 3.4. *Probability of missed detection for* $k_M = 3$

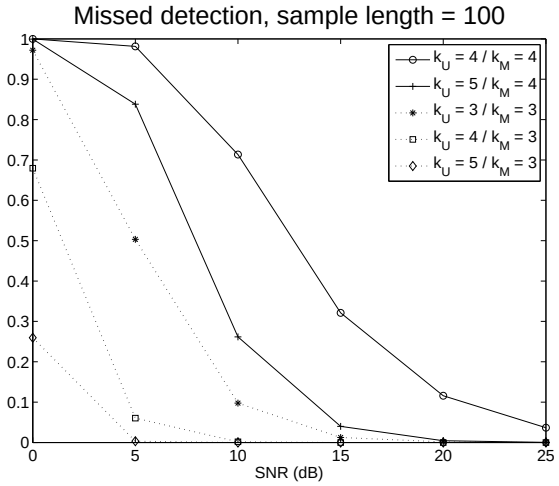

Figure 3.5. *Comparison of the missed detections for* $k_M = 3$ *and* $k_M = 4$*, for different values of the SNR*

for comparison. At high SNR (40 dB) the CS is highly reliable and thereby ideal with good approximation. As the load grows, the throughput reaches a maximum and then decreases to an asymptotic value, both of which increase with the SNR. At low SNR, this decrease can be explained by the fact that the CS tends to underestimate the number of concurrent transmissions and the higher the load, the more the users that try to simultaneously access the channel. It stems that it is more likely that at least one node will fail to detect the channel to be busy and will therefore overload the medium, with the consequent drop in throughput.

As suggested in Section 3.3.2, limiting k_M to 3 may yield better performance at low SNR, albeit at the price of a reduced usage of the spatial degrees of freedom. This strategy decreases the probability that more users than k_M may access the channel, as Fig. 3.5 shows. In addition, even if 4 nodes were simultaneously transmitting, this would not yet overload the channel. Moreover, the missed detection probability is negligible and hence collisions are very unlikely. The results are shown in Fig. 3.7. In

Table 3.1. *Simulation Parameters*

Parameter	Value
Data Rate	1 Mb/s
Tslot, $SIFS$	20, 10 μs
# of pairs	10
Packet size	10 kb
T_{ACK}	0.304 ms
M	4

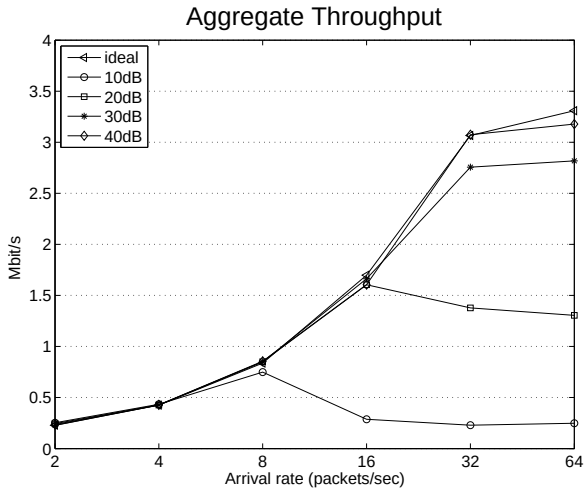

Figure 3.6. *Throughput of the original protocol ($k_M = 4$), for different values of the SNR,* 10 *pairs*

the ideal case $k_M = 3$ and $k_M = 4$ yield a throughput of 2.6 and 3.3 Mbit/s, respectively. Given the 1 MHz bandwidth and $M = 4$, the maximum achieveable throughput is 4 Mbit/s and thus 65% and 82% of the spatial degrees of freedom are used, respectively. A virtue of choosing $k_M = 3$ is that the system performance is not significantly sensitive to the SNR, while for $k_M = 4$ there is a dramatic impact of carrier sense inaccuracies. For instance, in saturation and for SNR = 10 dB, $k_M = 3$ has a throughput of about 2.6 Mb/s, while under the same conditions $k_M = 4$ offers just 0.25 Mb/s. Interestingly, setting the SNR to 10 dB the saturated throughput outperforms the ideal case because the missed detections of $k_U = 3$ users leads to some extra transmissions which do not overload the channel.

3.3.4 Discussion

We improved the protocol presented in [15] integrating a real CS mechanism based on the estimation of the rank of the interference correlation matrix. From a first study, we showed that the performance of these techniques are strongly dependent on the SNR. Moreover, we considered setting $k_M = 3$, which limits the number of concurrent transmissions but provides a more robust protocol behavior. The CS mechanism was tested in a network of 20 nodes that implement the modified OMAC

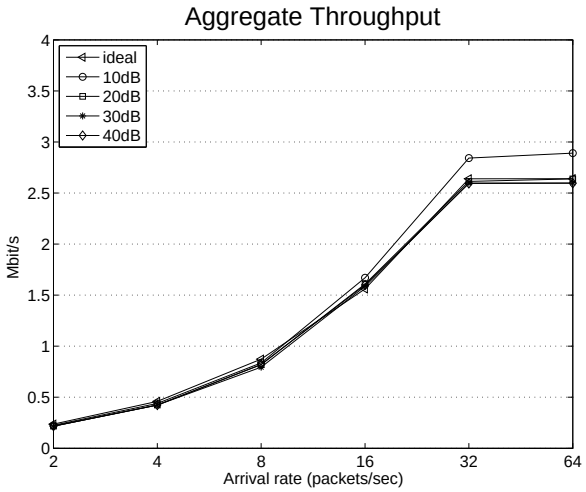

Figure 3.7. *Throughput of the modified protocol ($k_M = 3$), for different values of the SNR,* 10 *pairs*

protocol. For high SNR (30, 40 dB) $k_M = 4$ uses a large fraction of the spatial degrees of freedom. Instead, at low SNR the safer choice of $k_M = 3$ performs better. At very low SNR (10 dB), $k_M = 3$ yields an impressive performance gain of 1100% with respect to $k_M = 4$. Extensions of this work include multihop routing and adapting k_M to environment parameters (like the SNR).

3.4 A low complexity Model for Interference in Wireless Networks

3.4.1 Model Description

The goal of our work is to model a basic CSMA protocol with binary backoff (only data and ACK packets are included) by means of a semi-Markov process (Fig. 3.8). This is similar to Bianchi's model [20], although it uses a semi-Markov, rather than Markov, model and because it also includes carrier sense. The nodes are assumed to be in saturation conditions. The channel contention starts with a backoff (according to IEEE 802.11), which corresponds to state B1 in Fig. 3.8 and then the node sends its data frame. If the attempt is successful, then the terminal returns to B1 for the next packet. Otherwise, it undergoes a new backoff stage until either the maximum number of transmissions is reached or the communication is successful.

Since the binary backoff mechanism repeatedly tries to send the same packet, the chain structure is made up by a set of states (backoff, packet transmission and ACK waiting) that is replicated as many times as the short retry limit. Therefore the process has the following states:

- B1-B2-..-B6: the node undergoes the i-th backoff stage since it started contending for the channel for a tagged packet;
- TX 1-..-TX 6: the node sends the tagged packet for the i-th time.
- Wait ACK 1-..-Wait ACK 6: the node waits for the ACK after having sent the data packet for the i-th time.

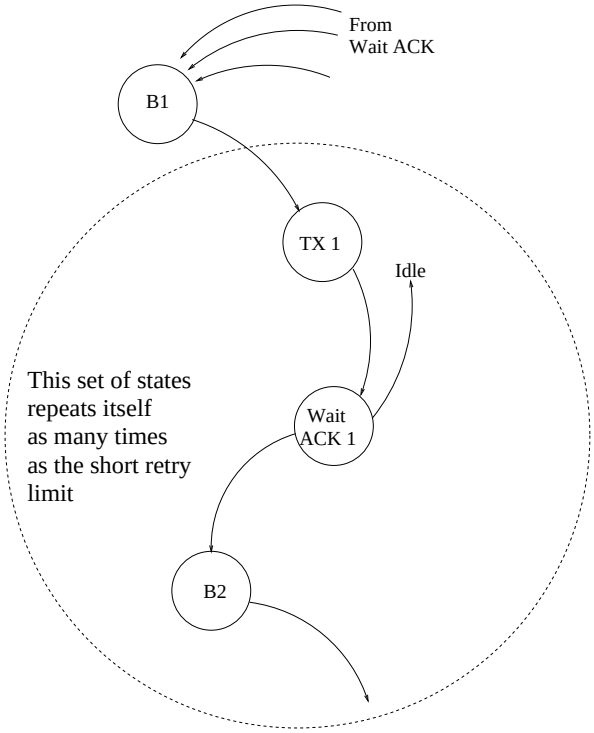

Figure 3.8. *Semi-Markov model describing the protocol*

This semi-Markov model is described by the matrices of transition probabilities and average transition times. All transitions depicted in Fig. 3.8 have probability one except for the transitions from the i-th Wait ACK state back to B1, which is equal to the probability that a data packet and the corresponding ACK are correctly received. This probability depends on the local interference and its computation will be considered later.

The transition times matrix elements are the following:

- from the i-th backoff state into the next transmission time: this time depends on the local interference and is covered in 3.4.1;
- in state TX i the chain transmits a data packet, thus T_{DATA} seconds elapse in this state where T_{DATA} is the time to send a data packet;
- in state Wait ACK i the node waits for the acknowledgment. Thus SIFS seconds plus the time to send an ACK are spent in this state.

The following parameters cannot be known *a priori* and must be estimated:

- the statistics of the interference;
- the probability of correct packet reception;
- the probability that at any given instant the interference may cause the carrier sense mechanism to prevent a node from transmission;

- the duration of a backoff stage.

These quantities are iteratively evaluated, since knowledge of the first item influences the other ones, while the others affect the node activity and communication attempts, which in turn have an impact on the interference statistics.

Interference model and parameter estimation

The choice of the interference model is a key point for the overall prediction accuracy. In this Section we evaluate a Gaussian mixture approach and compare it with the Gaussian model proposed in [8].

In [8], the aggregate interference Ξ is assumed to be Gaussian. The model is appealing because of the central limit theorem: the total interference is the sum of many (approximately incoherent) interferers. We suppose that every node transmits with a given fixed power P_T. Let us call $P_R(i,j)$ the power received at the j-th node when the i-th node is sending a packet, let $1[i](t)$ be the function that is equal to 1 if the i-th node is transmitting at time t or 0 otherwise, and let π_i be the long-run fraction of time that the i-th node spends transmitting a packet. Hence, $E[1[i](t)] = \pi_j$. Therefore, the power received by node j because of the i-th terminal is $P_R(i,j)1[i](t)$, and the aggregate interference $\Xi_j(t)$ is just the sum of these contributions:

$$\Xi_j(t) = \sum_{i \neq j} P_R(i,j)1[i](t) \tag{3.15}$$

and its mean is:

$$\overline{\Xi_j} = \sum_{i \neq j} P_R(i,j)\pi_i \tag{3.16}$$

The probability π_i is equal to the steady state probability that the node is in a state where it is sending a packet plus the probability that its backoff counter is 0 at a generic instant of time [2]. This addition is needed because such an event would lead the node to transmit within the next slot, and this event cannot be prevented by the carrier sense mechanism; in the IEEE 802.11 protocol, this would cause a collision. This probability is equal to $2P_{Bi}/(W_i+1)$ (see [20]), where P_{Bi} is the steady state probability of being in the i-th backoff stage and W_i is the contention window length of this backoff stage. Finally, the interference variance is $P_R^2(i,j)\pi_i(1-\pi_i)$, and the global interference variance is just the sum:

$$\sigma_j^2 = \sum_{i \neq j} P_R^2(i,j)\pi_i(1-\pi_i) \tag{3.17}$$

A single mode Gaussian model for the interference [8] is completely specified by Eqs. (3.16) and (3.17).

[2] For simplicity, we ignore the low-probability event that a node that is active in a slot will not continue its transmission in the next slot.

While this is a very intuitive and mathematically simple model, it suffers from a few drawbacks. The first one is that the central limit theorem may not necessarily apply. Let us focus on a specific terminal: in a dense network, some nodes may be very close to the tagged terminal, and their interference power may be orders of magnitude above that of the other nodes. Hence, their interference is very large and it may be even larger than all the background interference combined, thus making the final interference statistics very asymmetrical, and definitely not Gaussian.

The distinction between two "levels" of interferers (some dominant ones and the background interference) can be justified for two reasons. The first one is that other models in the literature (for instance in cellular networks models) assume that all the interference is generated by the first tier of neighbors. While we believe that this may be too simplistic a model for an ad hoc network, it proves that past research upholds the idea of dominant interferers.

Our second argument states that even the application of the central limit theorem may be partly questionable. Consider a network deployed inside a circle and a node at its center. All nodes generate some interference at the tagged node, and if their positions are randomly drawn inside the circle, their interference is a random variable as well. Since the received powers can be widely different (even by several orders of magnitude), the speed of convergence of the central limit theorem can be rather slow [21].

Therefore we have opted for a Gaussian mixture model. At each node we select the K closest nodes (with K in the range of 2-6), which generate the largest amount of interference. Given that we know which of these K nodes are transmitting, then they generate a certain (constant) level of interference. On top of this, the other (far) interferers are assumed to create a Gaussian background interference whose statistics can be computed as in (3.16) and (3.17). It is clear that the interference is a Gaussian mixture random variable, where 2^K modes are present, each corresponding to a possible combination of the dominant interferers states (active or waiting).

Let us call $\rho(i)$ the probability of the i-th configuration of dominant interferers. Then the final pdf of the interference at the j-th node is the following:

$$f_{\Xi_j}(x) = \sum_{i=0}^{2^K-1} \rho(i)\phi\left(x, \overline{\Xi_{j,i}}, \sigma_j\right) \tag{3.18}$$

where $\overline{\Xi_{j,i}}$ is average interference power in the i-th mode, σ_j is the standard deviation of the background model and $\phi\left(x, \overline{\Xi_{j,i}}, \sigma_j\right)$ is the pdf of a Gaussian random variable evaluated in x with mean $\overline{\Xi_{j,i}}$ and standard deviation σ_j. Note that σ_j depends only on the background interferers, and thus it is not a function of i.

The interference primarily affects two aspects of a CSMA protocol: error probability and busy channel probability. A closer look at these two situations suggests that two different types of interference are at work: in the first case, the tagged node is transmitting and its power can inhibit the activity of its neighbors. We shall call the interference perceived in this scenario as "inhibited". Instead in the second case the tagged node is not transmitting and thus the nodes in the neighborhood are not inhibited. This is called the "unconditioned" interference.

In the first case, we compute the received power due to the transmitting node at each node in the

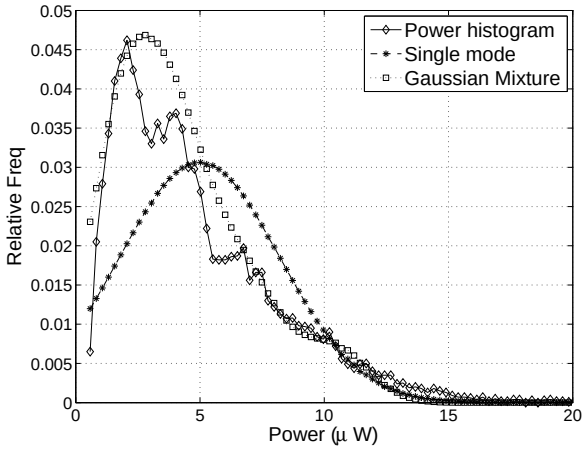

Figure 3.9. *Comparison of power histogram vs model PDFs*

network. If this power is above the carrier sense threshold, then these nodes can cause interference only if their backoff counter is 0. Thus their π_is include only the probabilities $2P_{Bi}/(W_i + 1)$. Terminals which are not inhibited are actually hidden nodes.

In the second case, all other nodes are not inhibited, thus the parameters as given in Eqs. (3.16) and (3.17) can include all the terminals in the network. In our model, the difference between these two types of interference is simply in the node set which is used to compute the overall interference.

Fig. 3.9 reports an insightful example: in a 40 node network inside a 40 $\times$ 40 m^2 square, we have compared the histogram of received power during a packet reception (computed by means of ns2 simulations) and the pdfs of the single mode and Gaussian mixture model for a randomly chosen node. For this example, a simple 4-mode mixture is applied ($K = 2$). All the models' parameters have been computed according to Eqs. (3.16)-(3.18). It is apparent that in the single mode case the standard deviation can be very close to the average interference value, thus there is a non-negligible probability (in this case about 15%) that the interference may be negative, while in the mixture case this is not the case. In addition, the histogram is asymmetrical (making a single mode Gaussian fitting quite hard), while the Gaussian mixture has enough degrees of freedom to fit the actual received power distribution. Incidentally, the mixture already works well with just 4 modes ($K = 2$ dominant interferers).

Finally, we would like to remark that some aspects of the wireless channels can be quite easily included in the model. Let us consider for instance fading (which is assumed to be constant at the packet level). Then Eq. (3.15) must include a fading term $F(i, j)$ that represents the fading coefficient from node i to node j. Since fading is unrelated to any node activity, then the mean value is just multiplied by the fading average value. Since long-term attenuation is incorporated in the path loss, $E[F(i, j)] = 1$ and therefore Eq. (3.16) goes unchanged. As far as the variance is concerned, Eq. (3.17) now becomes:

$$\begin{aligned}\sigma_j^2 &= \sum_{i\neq j} E[(F(i,j)P_R(i,j))^2 1[i](t)^2] + \\ &\quad - \sum_{i\neq j} E[F(i,j)P_R(i,j)1[i](t)]^2 = \\ &= \sum_{i\neq j} P_R(i,j)^2 \pi_i \left(E[F(i,j)^2] - \pi_i\right) \end{aligned} \tag{3.19}$$

Busy Channel Probability

Given that the interference statistics has been estimated, the probability that node j may perceive an interference above the Carrier Sense Threshold is equal to the complementary cumulative distribution function of the inhibited interference computed at ThCS, the Carrier Sense Threshold, i.e.: $P_{busy}(j) = 1 - F_{\Xi_j}[ThCS]$ ($F(x)$ is the cumulative distribution function evaluated at x). P_{busy} is related to the fraction of time the backoff counter is frozen.

Backoff Duration

Every average duration in the Semi-Markov chain is known a priori, except for the duration of the backoff stages. The length T_{Bi} of the i-th stage depends on the initial value of the backoff counter w_i and the length of every slot T_{SLOTk} (one slot being the time between two consecutive backoff counter decrements), which in turn depends on the carrier sense mechanism and the level of interference:

$$E[T_{Bi}] = E\left[\sum_{k=0}^{w_i-1} T_{SLOTk}\right] = E[w_i]E[T_{SLOTk}] \tag{3.20}$$

Our first assumption is that the random variables w_i and T_{SLOT_k} are independent of each other. The first term ($E[w_i]$) is simply $(W_i - 1)/2$ (since w_i is uniformly distributed in $\{0, 1, ...W_i - 1\}$). The random variables T_{SLOTk} are functions only of the interference level, and the computation of $E[T_{SLOTk}]$ must distinguish two cases, depending on whether the interference at the beginning of the slot is above or below the carrier sense threshold. While at the end of the previous slot the interference was clearly below this level (otherwise the counter would not have been decreased), we assume that the interference level at the beginning of the k-th slot does not depend on its value during the previous $(k-1)$ slots. Thus, the random variables T_{SLOTk} are postulated to be iid, because of the independence of the unconditioned interference power. This power may be above the ThCS with probability P_{busy}, and the node is assumed to freeze its counter for a duration equal to a data packet transmission time plus a DIFS. This duration is chosen because the most likely cause that may trigger the CS mechanism is that a node has started a new packet. Therefore the interference power will stay above the ThCS as long as the packet is being transmitted. While this is not strictly true, we make this approximation because of its simplicity. At the end of this packet transmission, the interference falls again below the ThCS for DIFS seconds and the backoff counter is eventually decreased. Therefore in this case the slot duration is equal to $T_{DATA} + DIFS + T_{SLOT}$. If instead the channel is perceived to be free (and this happens with probability $1 - P_{busy}$), then the total slot duration is just T_{SLOT}.

Given that the contention window for the i-th backoff stage is w_i, the average backoff duration is:

$$\begin{aligned} T_{Bi} &= E[w_i]E[T_{SLOTk}] = \\ &= [(W_i-1)/2][P_{busy}(T_{DATA}+DIFS+ \\ &+ T_{SLOT})+(1-P_{busy})T_{SLOTk}] \end{aligned} \tag{3.21}$$

All other transitions in the matrix have duration known *a priori*, because they correspond to packet transmissions/receptions.

Probability of correct packet reception

We assume that the interference statistics has been computed, and that $f_{\Xi_j}(I)$ is the value of its PDF evaluated at a given interference level I. Let us also suppose that the PER vs. SNR curve of the employed modulation/coding scheme is known. Then, the Packet Error Probability (PER) for node j can be computed as:

$$PER_j = \int_0^{+\infty} PER_j(I) f_{\Xi_j}(I) dI \tag{3.22}$$

This expression can be simplified if a capacity achieving code is employed. For these codes, the PER is 0 if the received SNR is above a certain threshold, and is 1 otherwise. In this case, if the received signal power is P and the threshold SNR is Λ_0, then $PER(I \leq P/\Lambda_0) = 0$ and $PER(I > P/\Lambda_0) = 1$. Thus Eq. (3.22) reduces to $1 - F(P/\Lambda_0)$. If we assume that the interference is a Gaussian mixture, each Gaussian mode implies a PER equal to $Q[(P/\Lambda_0 - \overline{\Xi_{j,i}})/\sigma_j]$, where $\overline{\Xi_{j,i}}$ is the average value of the interference for the i-th mode and σ_j its standard deviation. Finally, the overall PER is:

$$PER_j = \sum_{i=0}^{2^K-1} \rho(i) Q[(P/\Lambda_0 - \overline{\Xi_{j,i}})/\sigma_j] \tag{3.23}$$

where $\rho(i)$ is the probability associated to the i-th mode.

The simplicity of this expression is indeed appealing, since it only involves complementary Gaussian functions. Therefore it is adopted here to model the error correction capabilities of IEEE 802.11, even though this standard uses non capacity achieving convolutional codes. Simulations reveal that this approximation does not lead to significant discrepancies. Given the PER/SNR curves for IEEE 802.11 codes, we have set to 8 dB the correct detection threshold for the 6 Mbps data rate.

Model summary

We give a pseudocode description of the model:

```
for (counterIterations =
      1:maxNumberOfIterations) {
   for (nodeIndex = 1:allNodes) {
```

```
      1) compute the interference statistics
      2) evaluate
         a) the busy channel probability
         b) the average backoff duration
         c) the packet error rate
      3) solve this node's chain.
   }
}
```

The algorithm solves every node's chain several times (usually 5-6 iterations are enough for convergence). Every time a specific node is considered, it employs the other nodes' chains average statistics, which are implicitly assumed to be in steady state.

The algorithm complexity is linear with the number of iterations and quadratic with the number of nodes. Most of the complexity is due to the computation of the interference statistics: each node must collect the powers coming from every other node in the network, whose complexity is a linear function of the network size. Thus the overall complexity is quadratic with the number of nodes.

3.4.2 Performance Evaluation

We have compared the two models (single mode and Gaussian mixture) against ns2 simulations [3]. The network encompasses 40 nodes that adopt the IEEE 802.11g protocol, used at base rate (6 Mbps), and have been randomly deployed in squares of increasing size: $40 \times 40m^2$, $60 \times 60m^2$, $80 \times 80m^2$ and $100 \times 100m^2$. All nodes always have available packets to transmit.

Fig. 3.10 and 3.11 depict the average throughput per node achieved by ns2 simulations and the Gaussian mixture model. The optimal CS threshold changes with the network size, and it turns out that a larger area implies a smaller threshold, which is reasonable, because in this case the interference power is reduced (node density is lower), and so is the mutual interference. The model reasonably predicts the optimal CS threshold for all the network sizes. Several other network configurations (varying the number of nodes and the network size) have been tested, giving similar results. The difference between the simulated and predicted values is due to the inaccuracies of the model (for instance, the channel is sensed to be busy only for a single packet exchange or channel coding is assumed to be capacity achieving). Nonetheless, we remark that the shape of the curves is the same and our goal is to optimize the CS threshold by means of a simple model, not to replicate the exact performance of the network and its complex protocol.

In addition, we have reported the model predictions when the single mode interference model is adopted (Fig. 3.12). According to the previous discussion, this model does not have enough degrees of freedom to correctly represent the interference, and the results show that the optimal CS threshold is missed by 2-3 dB.

[3] We have modified ns2 interference models so as to model it more accurately, for instance by taking into account the capture effect and the temporal distribution of the interferers inside a packet.

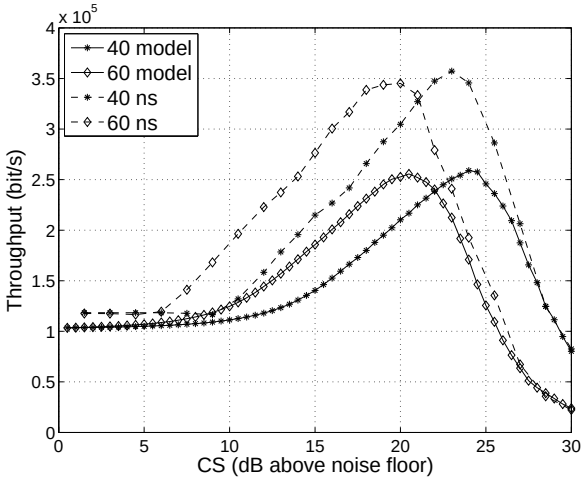

Figure 3.10. *Throughput for the mixture model vs ns2 simulations*

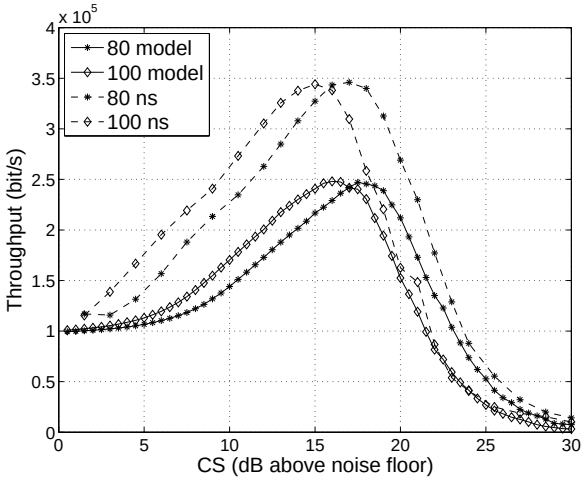

Figure 3.11. *Throughput for the mixture model vs ns2 simulations*

Finally, Table 3.4.2 summarizes the model predictions against the simulations when 60 nodes are deployed in the same area. The last column reports the average per node throughput computed by ns2 simulations at the optimal CS thresholds predicted by the models, as well as the loss that this model-based choice implies with respect to the true maximum. The CS threshold for the Gaussian mixture model is within 1 dB from the real one, and the estimation error entails a small throughput loss (about 15%), while for the single mode model this loss can be as much as 69%.

3.4.3 Discussion

We have analyzed how a simple Gaussian model can be used to estimate the interference behavior in a wireless ad hoc network for carrier sense threshold optimization. We have shown that a Gaussian mixture model can correctly predict the network behavior and enable a quick optimization of the network parameters.

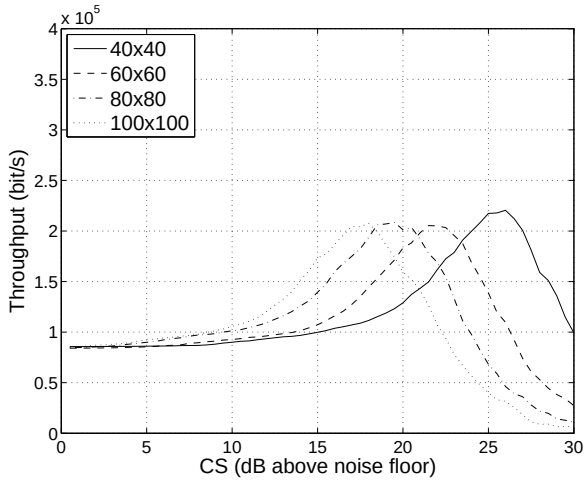

Figure 3.12. *Throughput for the single mode interference model*

Table 3.2. *Results for a 60 node network, 6 Mbps*

	Side (m)	Opt Th (dB)	Throughput (kb/s)
NS	40	25	312
Single mode	40	27	286 (-8%)
Gauss. mixture	40	25	312 (-0%)
NS	60	20	134
Single mode	60	25.5	45 (-66%)
Gauss. mixture	60	22	110 (-18%)
NS	80	17	86
Single mode	80	22	27 (-69%)
Gauss. mixture	80	18.5	80 (-7%)

3.5 Dynamic Carrier Sense Adaptation

3.5.1 Dependence of the CS threshold on the node density

In cellular networks it is well known that a basic way to deploy more capacity is to create smaller cells. The fundamental reason is that all signal strengths increase and thus the SIR stays constant so that in an interference limited scenario the PHY performance is largely independent of the cell size, while on the other hand being inversely related to the system capacity. However, in multihop CSMA networks also the absolute value of the interference (and not only its ratio with the intended signal) plays an important role. If the internode distance reduces, the SIR remains constant but the interference power increases and thus the probability that the channel may be perceived as idle would significantly drop. If the CS threshold is always kept at a very low level the channel would be perceived busy even when a single packet is being transmitted, and the network behavior would not be heavily affected. However, if the CS threshold is regarded as an optimization parameter this is no longer the case and interesting tradeoffs arise. We argue that the threshold value should scale with the intended signal strength.

Our work has focused on the optimal choice of the carrier sense threshold in the IEEE 802.11g MAC protocol. The two main contributions are:

1. we show that the optimal choice of the CS threshold critically depends on the node density according to a simple mathematical law; we give a qualitative proof of this fact and we empirically validate it by simulation;

2. we propose a novel, adaptive and fully distributed algorithm for local CS threshold optimization. Simulations show that it approaches the capacity offered by a centralized choice of the threshold based on *a priori* knowledge of the node density.

First, we briefly explain why the optimal CS threshold should scale with the node density. It has been proved that in a dense network performance is maximized when routes have short hops [22]. Therefore, the denser the network, the shorter its hops, since neighbors would be closer on average. Let us focus on a specific node T and let us assume that all its one hop neighbors are inside a circle of radius ρ, while the entire network is within R meters from T. All nodes whose distance is between ρ and R can not directly communicate with T and their transmissions only generate interference to it. In addition, let us express the received power $P(r)$ at distance r given a certain transmission power P_t (assumed to be the same for all nodes) and a certain propagation constant G as

$$P(r) = \frac{P_t G}{r^\alpha} \tag{3.24}$$

where α is the path loss exponent. A key assumption is adopted in the following analysis[4]. We assume that when the aggregate throughput is maximized, the average fraction of time that a node is transmitting ($\pi(i)$, with i the node index) does not significantly depend on the CS threshold, nor

[4]This assumption is made here only for analytical convenience, and is removed in the simulation results

on the node density. That is to say, in an optimized system nodes transmit for a certain fraction of time that is affected by the specific topology but not (as a first order approximation) by the optimal CS threshold itself. Since nodes transmit for a constant fraction of time in optimal condition, it can be inferred that the *average* interference perceived at T increases roughly linearly with the number of nodes N:

$$\overline{I} = \sum_{i=1}^{N} \overline{P(r_i)}\pi(i) \tag{3.25}$$

where $\overline{P(r_i)}$ is the average received power from the i-th node.

According to these hypotheses, the received power from a single node obeys (3.24). Since we assume that the interferer position is uniformly distributed in the circular annulus of radii ρ and R and that the ratio $R/\rho = K$ is fixed, the average value of $P(r)$ can be computed as the integral from 0 to $+\infty$ of the complementary cumulative distribution function of $P(r)$ itself, which is:

$$\begin{aligned} \mathcal{P}[P(r) \geq A] &= \mathcal{P}\left[\frac{P_t G}{r^\alpha} \geq A\right] = \mathcal{P}\left[r \leq \sqrt[\alpha]{\frac{P_t G}{A}}\right] \\ &= \frac{1}{R^2 - \rho^2}\left[\left(\frac{P_t G}{A}\right)^{2/\alpha} - \rho^2\right]; \\ &\quad \frac{P_t G}{R^\alpha} \leq A \leq \frac{P_t G}{\rho^\alpha} \end{aligned} \tag{3.26}$$

Since $\rho \leq r \leq R$, then $P(r) \geq P_t G/R^\alpha$ with probability 1. On the other hand, $P(r)$ cannot exceed $P_t G/\rho^\alpha$. Therefore, the average value of $P(r)$ is:

$$\begin{aligned} \overline{P(r)} &= P(R) + \int_{P(R)}^{P(\rho)} \mathcal{P}[P_R \geq A] dA = \\ &= P(R) + \frac{(P_t G)^{\frac{2}{\alpha}}}{R^2 - \rho^2} \int_{P(R)}^{P(\rho)} \frac{1}{A^{2/\alpha}} dA + \\ &- \frac{\rho^2}{R^2 - \rho^2}\left(\frac{P_t G}{\rho^\alpha} - \frac{P_t G}{R^\alpha}\right) \end{aligned} \tag{3.27}$$

If $\alpha = 2$ (like in Friis propagation model), then:

$$\overline{P(r)} = \frac{P_t G}{R^2} \frac{K^2}{K^2 - 1} 2\ln(K) \tag{3.28}$$

If, instead, $\alpha > 2$, then:

$$\overline{P(r)} = \frac{P_t G}{R^\alpha}\left[1 + \frac{K^{\alpha-2} - 1}{1 - 2/\alpha} - \left(\frac{K^\alpha - 1}{K^2 - 1}\right)\right] \tag{3.29}$$

for given α and R.

According to Eqs. (3.28) and (3.29), $\overline{P(r)}$ decays as $1/R^{\alpha}$. Therefore the average interference is proportional to N/R^{α} (see (3.25)), which we shall call *generalized node density* (and is equal to the true node density for $\alpha = 2$). The network performance can scale with the network size if the average fraction of time in transmission π is kept constant with respect to the node density and carrier sense threshold. Then the CS threshold must be increased because the absolute value of the interference rises. Otherwise, carrier sense would prevent node transmissions because of the higher received power, effectively reducing π as the node density rises. This analysis, even though based on strong approximations, suggests that the optimal network configuration can heavily depend on its size, which is not the case for cellular networks.

We have analyzed pure CSMA (rather than CSMA/CA with RTS/CTS handshake) so as to study carrier sense in isolation. While the inclusion of RTS/CTS is a topic for future research, we expect that the given linear dependence of the optimal CS threshold on the generalized node density should not change. The reason is the following: in the above proof, all the nodes inside the inner circle are not allowed to transmit. However, when the RTS/CTS mechanism is activated, all the terminals which can decode one of these two packets will be inhibited. Thus, nodes inside the union of the coverage areas of the transmitter and receiver will not be allowed to transmit and the above discussion goes basically unchanged. As a result, although the actual numbers of the linear dependence may change, the qualitative dependence of the optimal CS threshold on the generalized node density is expected to hold.

In order to verify this idea we have found by means of ns-2 simulations[5] the CS threshold that maximizes the aggregate throughput for the Friis propagation model (α = 2) without fading. The MAC protocol is IEEE 802.11g at base rate (6 Mbps) in the pure CSMA mode (no RTS/CTS packets are used). We have tested networks with 30, 40, 100, 150 and 200 nodes. The terminals are located in square areas, whose size ranges between 30 and 1400 meters, thus the node density varies between 0.75 and 450 nodes/hectare, sweeping over two orders of magnitude. All terminals are in saturation conditions. The results are reported in Fig. 3.13, which shows that the optimal CS threshold linearly increases with respect to the network density. This empirical relation also holds for different packet sizes and data rates, even though the coefficients may depend on these variables. Finally, we note that the points in Fig. 3.13 never go below the noise floor (set to -101 dBW in our simulations) and the optimal CS threshold can never be smaller than this value.

Moreover, we have also analyzed the value of the optimal CS threshold when a two-ray propagation model ($\alpha = 4$) is adopted (Fig. 3.14): now the CS threshold scales linearly with the generalized node density N/R^4.

3.5.2 A Distributed Algorithm for Threshold Optimization

It has been proved that if all the nodes share the same CS threshold, the optimal value of this parameter is strongly affected by the network density. There are at least two ways to exploit this

[5]We have modified ns2 interference models so as to compute it more accurately, for instance by taking into account the cumulative character of interference, the capture effect, and the temporal distribution of the interferers inside a packet transmission.

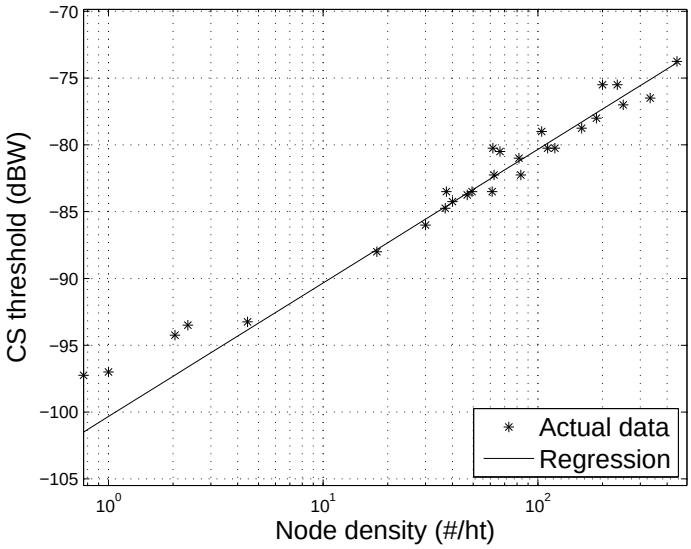

Figure 3.13. *Optimal carrier sense density with respect to the node density, free space path loss model*

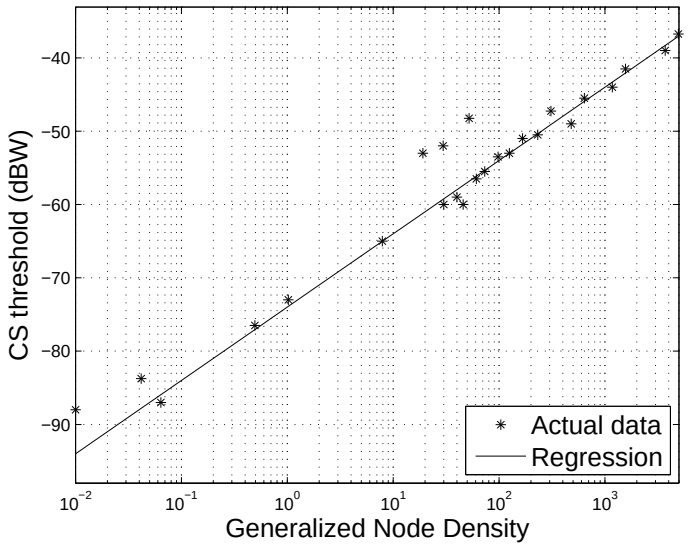

Figure 3.14. *Optimal carrier sense density with respect to the node density, two rays path loss model*

behavior. In the first case, if the node density is known, all nodes can have their CS threshold set to the optimal value. However, this is a centralized approach that requires *a priori* knowledge of the (generalized) node density. On the other hand, it is possible to create a distributed scheme where each node estimates the local node density, e.g., according to the algorithm in [23], and then sets its own CS threshold in agreement with the relationship previously found. That is to say, given a certain estimated generalized node density $\hat{n}$, the CS threshold Th (expressed in W) is equal to $C\hat{n}$, where C is a scaling constant that depends on the data rate. We have found C by regression. For instance, at a data rate of 6 Mbps and in free space, C can be computed from Fig. 3.13, and is equal to 93 pW/(nodes/hectare).

This is a distributed algorithm that performs a local optimization of the CS threshold. We would like to outline the difference with the algorithm in [9]: in our case, we base the choice of the CS threshold on the estimated local node density, and the CS threshold for the different nodes need not be equal. Instead, in [9] the authors propose to adapt the threshold according to the estimated SNIR. This estimate may be larger than a certain upper threshold T_{high}, smaller than a low threshold T_{low}, or in between. When the next ACK is sent, if the "Duration ID" field is unused, this field will include an indication to increase, decrease or keep constant the CS threshold, according to which of the three previous cases has happened. The nodes will then collect these ACK packets and at the end of a suitable period they will update their estimate by using the most conservative approach according to this information. For instance, the CS threshold will be decreased if even a single ACK states so. This mechanism has been introduced in order to keep the threshold variations under control. Moreover, since the most conservative CS value is adopted, nodes will usually have very similar CS threshold values. In conclusion, the scheme by Zhu et al. employs an SNIR estimate and in addition it requires some information exchange among the nodes, while we do not require it. However, we need to know the path loss exponent of the propagation environment, while [9] is not affected by this physical layer parameter.

3.5.3 Performance Evaluation

The performance of our approach has been compared to a conventional IEEE 802.11 network where the CS threshold was either 5 dB above the noise floor (thus set to a very low value) or optimized according to the centralized approach (but still constrained to be the same for all nodes). The network is composed of 30 or 50 nodes (all in saturation), uniformly distributed in a square whose side is 40, 80 or 120 meters. Every value is computed by averaging 80 simulations, each lasting 100 seconds. The routing protocol used is AODV, the transmission power is 10 dBm and the propagation environment is free space. Throughput results are reported in Figs. 3.15-3.18 for data rates of 6 and 12 Mbps. First of all, both algorithms achieve on average a 40% improvement over a conservative choice of the CS threshold. In addition, the centralized (where all nodes must have the same CS threshold value) and distributed (where each node can have a different CS threshold level) approaches have comparable performance. Our simple and practical distributed algorithm yields a 40% improvement over the IEEE 802.11 benchmark. Moreover, the optimized fixed-threshold case and

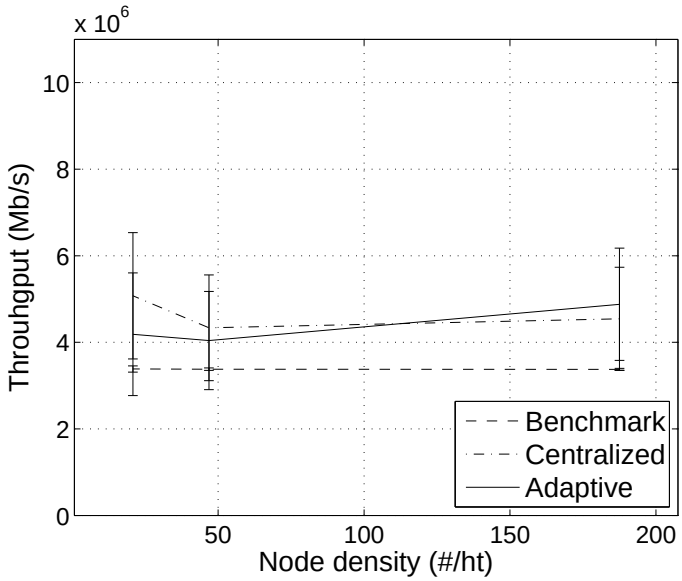

Figure 3.15. *Aggregate throughput, 30 nodes. Data Rate is 6 Mbps.*

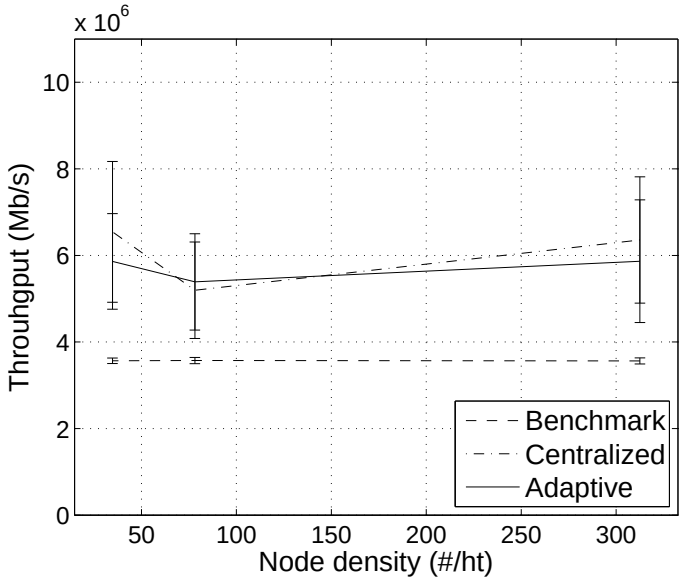

Figure 3.16. *Aggregate throughput, 50 nodes. Data Rate is 6 Mbps.*

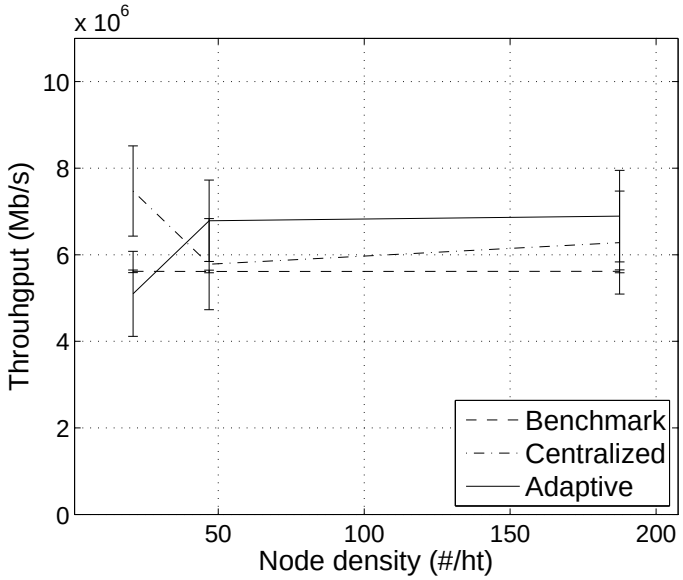

Figure 3.17. *Aggregate throughput, 30 nodes. Data Rate is 12 Mbps.*

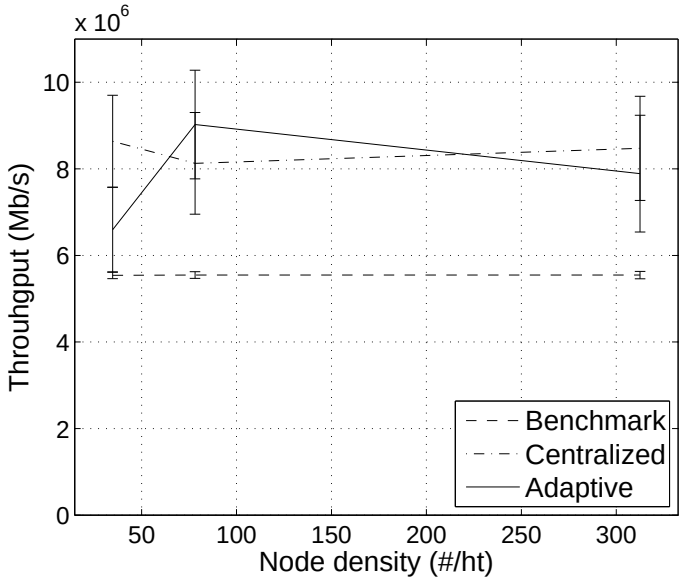

Figure 3.18. *Aggregate throughput, 50 nodes. Data Rate is 12 Mbps.*

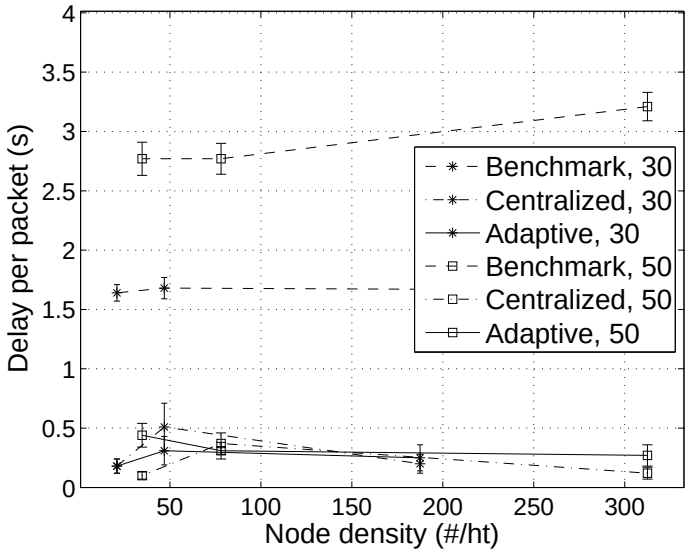

Figure 3.19. *MAC delay. Data Rate is 6 Mbps*

the adaptive algorithm show large standard deviations. The reason is as follows: with a very low CS threshold, only one transmission is allowed at any given time (except during a collision) and thus the throughput does not depend on the topology (note that the throughput is constant with respect to the node density). Instead, when a higher threshold is employed, the specific node arrangement critically affects the number of possible simultaneous transmissions.

Figs. 3.19-3.20 compare the MAC delay for the three approaches (data rates again of 6 and 12 Mbps, respectively) and our simulations show that this metric can be significantly reduced (by about 50%), because nodes are far less restrained while the success probability is still high. Incidentally, Figs. 3.19 and 3.20 show that the delay of conventional IEEE 802.11 heavily depends on the number of nodes. The delay curve for the 30 node network is lower than that in the 50 node case. Since only one transmission is allowed at any time, the more the nodes in the network, the longer the average delay to gain channel access. On the other hand, our algorithm adapts itself to traffic intensity, and therefore it is more robust to the network configuration precisely because of its adaptation.

Fig. 3.21 shows the time evolution of the average route hop count length vs time in the 80 $\times$ 80m, 50 node network. It is evident that the hop count significantly increases at the beginning of the network history. The reason lies in the relatively high CS threshold that is set by our algorithm. Such a choice increases the node activity and then the interference. The lower SNIR disconnects nodes that would have been otherwise linked in conventional IEEE 802.11 with lower interference. Thus, on average the routes require more hops because the effective communication range shrinks. However, the reduced hop length preserves an adequate SNIR. These two elements (higher interference and smaller hop length) work against each other, but it turns out that the latter one paired with the high CS threshold has a bigger impact. The upshot is that the delay on each hop is relatively low, because the high CS threshold prevents transmissions only in a small number of cases. Therefore communications are faster and still reliable, leading to the overall performance improvement (Figs. 3.15-3.24).

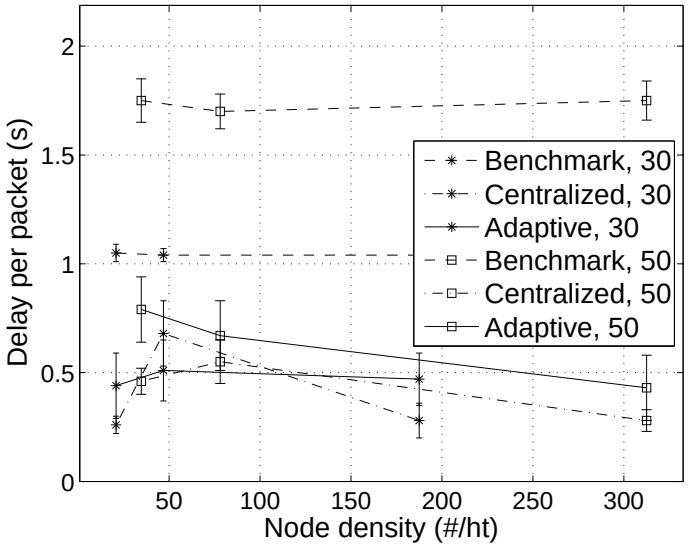

Figure 3.20. *MAC delay. Data Rate is 12 Mbps*

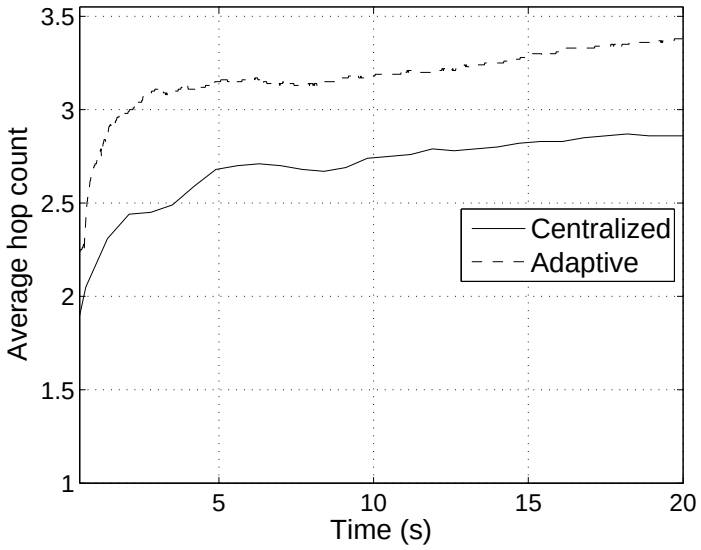

Figure 3.21. *Average Hop Count Length vs time in the optimized case*

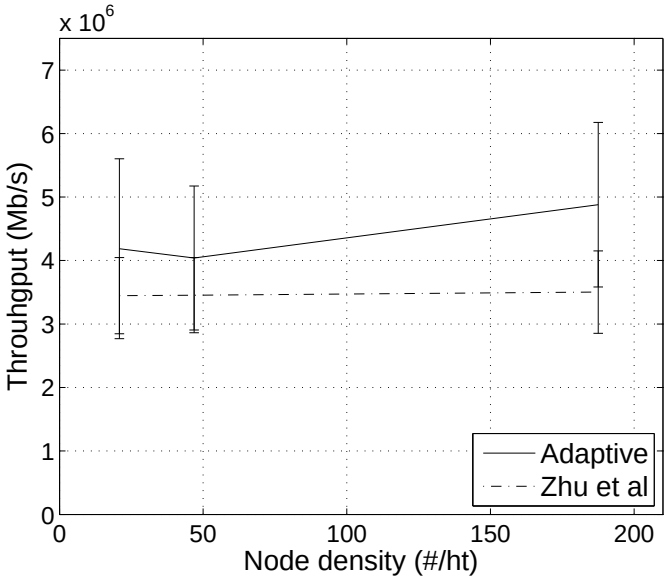

Figure 3.22. *Aggregate throughput, 30 node network. Data Rate is 6 Mbps.*

Finally, we compare the performance of our algorithm with that of Zhu et al. [9]. The basic difference between our algorithm and Zhu's is that in our case the CS thresholds are allowed to have different values at different nodes, so as to match the local interference and congestion level, while [9] forces all nodes to have the same threshold. The flexibility of our approach yields an overall performance improvement as shown by Figs. 3.22-3.23 for the aggregate throughput and by Fig. 3.24 for the delay. The latter graph is particularly important, because it proves that a local, rather than global, CS threshold optimization lets nodes in highly congested areas to have significantly faster data transfer.

3.5.4 Discussion

We have shown (by analysis and simulation) that the optimal choice of the CS threshold for throughput maximization depends linearly on the generalized node density defined as N/R^{α}, where N is the number of nodes in the network, α is the path loss coefficient and R is the network radius. We exploit this relationship by building a simple, fully distributed and effective algorithm for CS adaptation, that offers significant advantages over conventional IEEE 802.11 and is competitive with the state-of-the-art CS adaptation algorithm in [9]. Our future work will study the interaction between transmission power control and CS tuning and the performance evaluation of our system when important parameters such as the path loss coefficient are incorrectly estimated.

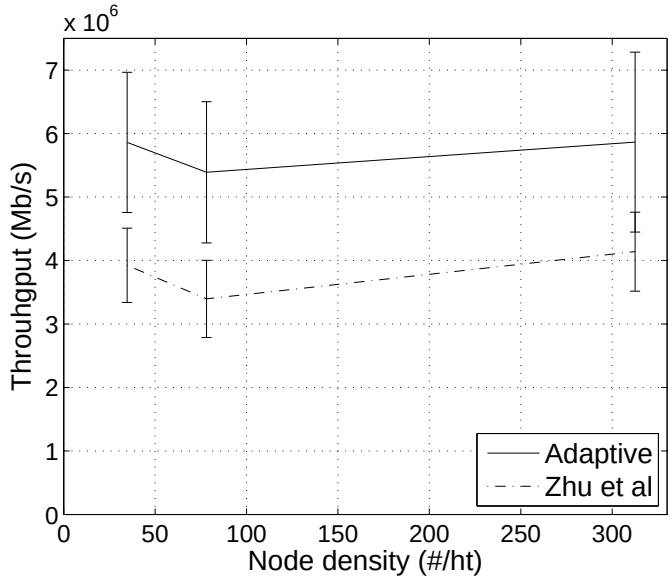

Figure 3.23. *Aggregate throughput, 50 node network. Data Rate is 6 Mbps.*

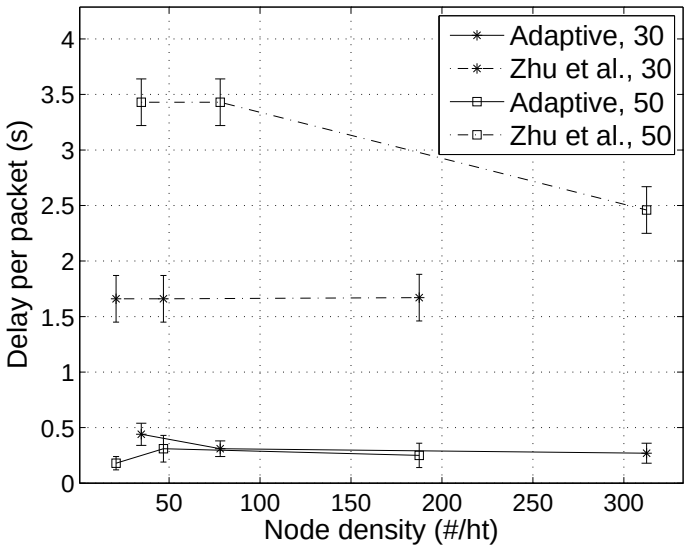

Figure 3.24. *MAC delay for the adaptive algorithms. Data Rate is 6 Mbps*

3.6 A Physical Model Scheduler for Multi-Hop Wireless Networks Based on Local Information

In multi-hop wireless networks, routing strategies are used to identify the best paths to deliver information from source to destination terminals. Moreover, since the radio channel is shared, access to it must be managed through proper scheduling algorithms. These two problems can be combined in order to realize a joint routing and scheduling framework, whose theoretical basis has been posed by the pioneering work reported in [24, 25].

In this context, the network is represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where nodes in $\mathcal{N}$ are the network terminals and edges in $\mathcal{E} \subseteq \mathcal{N}^2$ are the communication links. In the following these terms (nodes and terminals, or edges and links, respectively) will be used interchangeably. Scheduling and routing are addressed by considering a transmission over a given link $e \in \mathcal{E}$ as corresponding to *activating*, i.e., "turning on," e, which is conversely inactive / turned off, when the link is not used for transmission. By looking at the activations of links in a sequential manner over time, one can determine scheduling for a Time Division Multiple Access (TDMA). The subsequent activation of links from a source node to a destination also implies routing. This framework can address, e.g., the minimization of the time required for the information delivery; this corresponds to an efficient utilization of the network capacity, i.e., allowing the simultaneous activation of a large number of transmissions, while checking at the same time that the active links bring information toward the desired destination.

Even though these theoretical principles are well studied, the network descriptions commonly used often abstract from a detailed characterization of the wireless medium. The only requirements used to determine the admissibility of a scheduling pattern relate to flow conservation and to avoiding the simultaneous utilization of a node for transmission when it is receiving a packet, or viceversa, a condition which in most of the literature [26, 27] is referred to as *primary interference constraint*. Note that this terminology is somehow improper, since this condition does not really depend on wireless interference, but rather on the half-duplex transceiver capability, which limits the number of simultaneous operations which can be performed at a time [28].

In fact, multi-hop radio networks may suffer a severe capacity limitation due to *wireless interference* phenomena. The majority of the approaches to model wireless interference beyond the primary constraint, e.g., involving at least two disjoint pairs of nodes, follow the classification made in [22], which distinguishes between the so-called *protocol* and *physical interference models*. Other variations have been proposed [29], e.g., to take into account additional aspects such as the capture effect, but these proposals can also be related to the aforementioned distinction.

The protocol model describes wireless interference by means of *conflict sets* (sometimes this representation is also translated into graphs called *conflict graphs*). For every edge $e \in \mathcal{E}$ the conflict set $\mathcal{I}(e) \subseteq \mathcal{E} \setminus \{e\}$ is defined as containing all links whose simultaneous activation with e is forbidden due to interference. This is simply modeled as a binary relationship: given a link $f \in \mathcal{E} \setminus \{e\}$, it can either interfere with e, thus is put into $\mathcal{I}(e)$, or not. However, this representation fails to capture that interference has a cumulative effect, since the simultaneous activation of multiple links may cause

too high an interfering power for link e, even though none of these links alone is to be considered as interfering with e, and thus does not belong to $\mathcal{I}(e)$.

In spite of this problem, such a model is adopted in most of the literature which deals with routing and scheduling issues to capture wireless interference [30–33]. However, the more realistic physical interference model should be preferred, as pointed out in [34]. The main problem in applying the physical interference model is instead on the complexity side, because it requires to check the Signal-to-Interference Ratio (SIR) at the receiver's side of all active communication links and to evaluate if it is above a given threshold. This requires a large amount of information, namely the link gains between any pair in $\mathcal{N}^2$.

To overcome this problem, in the present research effort we propose to exploit the interference model previously proposed to reduce the complexity of the physical model and enable its use when certain interference terms are difficult to quantify exactly, especially because they consist of many small contributions to the overall interfering power. The rationale behind this model is to define, for each node, a number K of *dominant interferers*, which are, typically, the K closest neighbors. For each node, the channel gains towards its dominant interferers must be known precisely, whereas the rest of the network is simply described in statistical terms. If K is properly chosen, the approximation introduced by this representation is almost negligible, while the complexity of the description can be significantly reduced.

To validate our model, we evaluate the scheduling time on graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ having different kinds of topologies with single-path routing and where the destination set contains only one node. All edges in $\mathcal{E}$ are also directed toward the root, possibly via multi-hop relaying, without multi-path. This choice can be justified by several reasons. On the one hand, this allows us to focus on scheduling only, without being involved in considerations about routing optimality, since in such a topology there is only one possible route from any node to the root. This also allows a simpler implementation of the scheduler, since it is reasonable to use, as will be argued in the following, greedy scheduling strategies which maximize the number of packets forwarded toward the sink. On the other hand, this kind of topology is still realistic, and can actually be envisioned in many implementations of wireless multi-hop networks, such as the IEEE 802.16 Mesh mode operating with centralized scheduling [27].

We will compare our interference model with alternative techniques, using either the protocol model or a different implementation of the physical model [34] and we also use, as a benchmark, the optimal scheduling computed through exhaustive search. We perform several evaluations with the Network Simulator 2 (ns-2) [35]. Numerical results show a very good agreement between the performance limit and our proposed strategy. At the same time, our interference model obtains a significantly better scheduling with respect to the protocol model and we are also able to improve the results obtained in [34], where a physical model is used. This justifies our model as a practical strategy to use the physical interference model in wireless multi-hop networks.

The general approach used in the literature to describe TDMA scheduling in multi-hop wireless networks can be found in [24, 25]. In [24] the scheduling problem is studied through linear programming, and a polynomial complexity algorithm which solves the pure scheduling problem is given. In [25], and all the extensions to this framework proposed in other works by the same authors, the

problems of routing and scheduling for wireless packet networks are framed in the more general context of identifying a suitable *link activation pattern* which satisfies certain optimality criteria and is subject to certain constraints, so that a linear programming framework can be derived. The main goal of this approach is to minimize the delivery time from all sources to all destinations; to this end, the network capacity must be efficiently exploited.

The contribution we give in the present Section can be viewed as an extension of this approach to a more realistic wireless environment. In fact, in [25] a more accurate characterization of wireless interference is left open; in most of the developments derived from this framework, simplified approaches such as the protocol interference model are used. For example, different theoretical aspects of scheduling in multi-hop networks are investigated in [30–33] by means of link activation schemes which rely on the protocol interference model only.

In more detail, in [32] the problem of delay guarantees in wireless multi-hop networks is studied. Differently from our approach, which closely follows [25], in this paper the per-flow delay is considered instead of the overall time to deliver all the packets to their respective destinations. In [30], the authors outline and investigate from a high level perspective certain bottleneck problems which arise in joint routing and scheduling scenarios. In this way, related performance bounds are highlighted. However, the analysis heavily relies on the protocol interference model, so it is unclear which conclusions can be extended to general wireless scenarios where a different interference model is to be adopted. The contribution of [31] is an analysis of optimal scheduling conditions, again based on the protocol interference model. With this background, a fair scheduling mechanism is proposed and discussed to activate wireless links, based on the maximal clique search over a graph. Finally, the contribution of [26] is to discuss a linear programming approach in order to solve routing and scheduling and to introduce practical algorithms based on efficient heuristics. Similarly to the analysis reported in [25], the only limitation imposed by wireless interference is that nodes can not be active in more than one operation (which can be either a packet transmission or a packet reception). This rationale is extended by the authors in [33] to a case where wireless interference is considered in a broader sense, but again this involves the protocol model only. Instead we consider the problem of obtaining efficient scheduling heuristics when more realistic wireless interference models are considered. Note also that this poses an issue in terms of computational complexity and information exchange, which we will address in the following.

In this sense, our approach is similar to [34], which also focuses on the physical interference model, even though it introduces a simplification to prioritize the links in the scheduler. However, in that paper the challenge of multi-hop transmission is mitigated since intermediate nodes have a backlog equal to the aggregate backlog of all previous nodes. This simplification makes explicit relaying unnecessary (i.e., a packet can be forwarded even if it has not been received by the relay, because the relay backlog has been suitably increased to take these forwarded packets into account). Thus, the algorithm was run on a set of single-hop flows, whose traffic demands were sized so as to take routing into account. In our analysis, we consider multi-hop more explicitly, as nodes are also required to relay packets and therefore the status of the information delivery has to be constantly monitored also at intermediate nodes.

Finally, paper [27] also presents some similarity with our analysis, mostly in the fact that a similar application scenario is considered. In fact, in that paper the specific case of IEEE 802.16 Mesh mode operating with centralized scheduling is addressed, which might result in node placements similar to the topologies considered in our work. However, we remark that in [27] again only primary interference constraints are taken into account. For this reason, our investigations can be seen as an extension of this work to a more realistic interference description.

3.6.1 Scheduler Design

In designing the scheduler, we have considered different kinds of node placements, both deterministic and random, in an assigned area. A realistic propagation model (e.g., also including fading effects) has been considered, both in the analytical model presented in Section 3.4.1 and in the numerical evaluations in the following. Thanks to this accurate radio propagation description, even when the node deployment is regular, channel impairments occur in an unpredictable manner. For what concerns the random placement, the obtained topologies are even more variable as they are rather different in terms of node degree and length of the links. We also remark that we have kept a quite general approach for what concerns lower layers than the scheduler.

About higher layers instead, and more specifically routing, we focused here on graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with only one destination node and where a single path is available from any node to the destination. The main reason to consider single-path topologies is in order to abstract the evaluation of the scheduling performance from routing. In fact, as shown in [36], there are many considerations which would advise for a cross-layer approach where routing and scheduling are performed jointly. If TDMA scheduling is coupled with a sub-optimal routing, it can achieve very low efficiency; even worse, a comparison of scheduling strategies obtained in this case may not reflect reality, since it is biased by routing inefficiencies. In our scenario this problem does not occur, as the routes are uniquely determined, and every non-root node can only transmit over a single edge, i.e., the one toward its parent node. This allows us to decouple the scheduling from the routing problem and to investigate in a more direct manner the application to a scheduling problem of the interference evaluation framework presented in Section 3.4.1 In our opinion, studying single-path networks is a first necessary step in order to gain valuable insight on the problems related to scheduling, but our future work will extend these findings to multi-path networks where a routing algorithm is also included.

This scenario corresponds to having a tree topology (either inherently derived from the radio propagation or superimposed by a Minimum-Spanning-Tree algorithm as is done by many routing algorithms [37]), where the root can be seen as a gateway node, reachable via multi-hop by all nodes, and in charge of collecting the information from them. Exchange of packets is allowed only from a node to its parent node in the tree hierarchy; however, all nodes can generate some interference at a receiver node, depending on their physical placement and not on the logical position in the tree. This actually happens in many scenarios, such as Wireless Mesh Networks operating with centralized scheduling in IEEE 802.16 Mesh mode [27], or Wireless Sensor Networks for distributed measurements [38]. In the former case, the gateway (i.e., the root node) is the access point of the

wireless network backbone to a cabled connection to the Internet, in the latter it is the central data collecting unit; in general, it is sensible to think of it as the location where the centralized scheduling algorithm is run.

As a consequence of focusing on single-path topologies, we can simplify the computational complexity of the search for the minimal time schedule. In particular, even simple greedy scheduling strategies can be utilized as efficient schedulers, as justified by the following discussion. If $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_N)$ describes the queue lengths at all non-root nodes in $\mathcal{N}$, we denote with $\tau(\boldsymbol{\ell})$ the minimum time to deliver all the packets of $\boldsymbol{\ell}$ to the tree root. Also, we call $\mathbf{e}_k$ the canonical base vector equal to 1 at the kth entry and 0 otherwise. It is easy to prove that, if $i \in \mathcal{N}$ is the parent node of $j \in \mathcal{N}$, for any fixed vector $\boldsymbol{\ell}$ the value of $\tau(\boldsymbol{\ell}+\mathbf{e}_i)$ is not greater than $\tau(\boldsymbol{\ell}+\mathbf{e}_j)$. In fact, the transmission of $\boldsymbol{\ell}+\mathbf{e}_i$ can be achieved with the same optimal activation pattern for $\boldsymbol{\ell}+\mathbf{e}_j$ by turning off link $j \rightarrow i$ in one of the slots where it is active.

In order to reduce the computational complexity of the scheduling, we might want to define a slot-wise activation criterion. Of course different criteria are possible for the specific choice of which links to activate. However, the reasoning above implies that, within an already existing activation set, turning on another link which is compatible with interference conditions is always beneficial on single-path routing topologies. We can generalize this, when determining the link activation pattern for a given time-slot, stating a practical rule of thumb that the more active links, the better. Therefore, greedy schedulers appear to be appropriate for the scenario under study, since they offer very good performance and also seem to be better implementable in practical scenarios. Also, they allow comparisons to test the goodness of our proposed interference model against existing solutions present in the literature, which also rely on heuristic schedulers often with a greedy rationale. However, we stress that the choice of focusing on heuristic greedy scheduling does not give any advantage to our proposed model, and we may reasonably infer that a similar comparison would also hold for more detailed theoretical investigations performed within an optimization framework, which are however out of the scope of this work, being far more difficult to implement and compare.

Finally, we remark that our scheduler is a centralized one. However, nodes employ accurate local information (about the dominant interferers) and the rest of the network is modeled in statistical terms. Thus, our scheduler makes some important steps toward the goal of a distributed system based on the physical model, rather than the oversimplified protocol model.

3.6.2 Implementation Issues

In this section, we describe a low-complexity, centralized scheduler for transmitting data in the uplink of the tree topology (i.e., from all nodes to the root). In our setting, time is slotted and in each slot a centralized controller, e.g., located at the tree root, activates some links to transmit data. The key concept in our scheduler is that the nodes are selected in a greedy manner according to their chances of successfully transmitting a packet. This probability is estimated by means of the interference model reported in Section 3.4.1.

The scheduler knows the queue status at all nodes. We remark that this is the only information

for which a shared knowledge is required: this can be important if a distributed implementation is sought. We also point out that it is possible to achieve good performance also with rather coarse information about queue status; it is often enough to know if the queue is empty, or its length is below or above a certain congestion threshold. Simulation traces show that such a strategy often leads to satisfactory results. Each terminal i is associated with a weight ψ_i, which is the sum of two factors: the probability of successful packet transmission and a function of the queue status. The success probability at a node is computed by means of (3.23) and multiplied by a constant suppression factor $\alpha \in [0, 1]$ if a node transmitted a packet in the previous slot, in order to improve fairness for nodes who are in disadvantaged positions, which otherwise would have fewer transmission opportunities. The queue status obviously influences the scheduler in the sense that nodes with empty queues are not eligible for transmission. Additionally, in order to keep the queue lengths under control, we assign a bonus b to the nodes with long queues. The weight ψ_i is equal to this bonus b plus the success probability, multiplied (if necessary) by α. In the following numerical evaluations, for a given queue length ℓ, this bonus b follows a linear piecewise function:

$$b = \begin{cases} 0 & \ell < \ell_A \\ \frac{\ell - \ell_A}{\ell_B - \ell_A} & \ell_A \leq \ell \leq \ell_B \\ 1 & \ell > \ell_B \end{cases} \tag{3.30}$$

The thresholds ℓ_A, ℓ_B have been empirically set to, respectively, 150% and 250% of the value of the initial node backlog. Note that this additional term is particularly useful in a tree topology since the nodes closer to the root have a higher traffic to deliver, since they also act as relays, but the leaves generally have a greater chance of being scheduled, because they have fewer neighbors and thus less interference.

This is the main loop the scheduler performs in each slot:

1. **select** the candidates
2. **compute** the queue size bonuses
3. **while** (the candidate queue is not empty)
   for (nodeIndex = 1:allNodes)
   (a) update the interference statistics and the packet success probability: for every node j, compute (3.23) by estimating the probability that each dominant interferers w of j transmits, as equal to its weight ψ_w. Add to all weights the queue bonus.
   (b) pick the best node
   (c) is this link compatible with the existing communications?
   yes) include it in the list of scheduled nodes. For the rest of the time-slot, its activity factor is 1 (i will surely transmit) and the activity factor and weight of every node k which is a neighbor of i will be set to 0 (k will not transmit). All the neighbors are removed from the candidate list.
   no) remove it from the candidate queue. Set its activity factor and weight to 0.

end_for
end_while

In phase 1, the scheduler selects which nodes may be eligible to be scheduled. In the default implementation of the scheduler, only nodes whose weight ψ_i is larger than 0 and whose queue is not empty continue in the following steps.

Step 3 (the *while* loop) can be fully understood realizing that the interference model is completely specified when the number of dominant interferers K and all the received powers and activity factors are defined. The first term is a constant and the second does not change within a slot. However, the activity factors must be given suitable values. If a node cannot be scheduled in this slot (e.g., it has an empty queue or there is a direct link with a node which has been scheduled to transmit), then it is assigned the value 0. Otherwise, the activity factor can be equal to 1 if the node has been scheduled to transmit. In all other cases, when it is not yet defined whether the node may or may not transmit, the activity factor is set to 1/2 if the node has not sent a packet in the previous slot or $\alpha/2$ if it has. Please note the inclusion of the suppression factor α. This reduces, in the interference model, the transmission probability (and therefore, the weight ψ_i) of the nodes which transmitted in the previous slot. Thus, the other terminals will predict a higher SIR and will compute a higher success probability. Finally, if $\alpha = 0$, then weight ψ_i for certain nodes will be 0 and they will not have a chance to transmit in the next slot. We shall discuss the impact of α in Section 3.6.3.

In step 3a, all the nodes update their interference model, which means to set the activity factors of the dominant interferers. In step 3c, if the new link does not decrease the SIR of the other nodes below the target level, it will be scheduled in the incoming slot. However, its father and all its children in the tree must have their weight ψ_k set to 0 because they will not transmit due to the half-duplex constraint. If instead the candidate link is incompatible with the links already scheduled, it is discarded and it will not be considered for the rest of the slot as a possible candidate. Thus its weight ψ_i is set to 0.

We point out that step 3c is carried out taking into account all ongoing transmissions in the network. This step is not distributed and requires global knowledge. However, [34] is subject to the same requirement, while in the other steps our scheduler requires less knowledge than the algorithm in [34].

The computational complexity of the scheduler with respect to the network size N depends on the interference statistics model, which affects step 3a. Its complexity is $O(K)$, since the weights of the dominant interferers are updated. Steps 3b and 3c have a constant complexity with respect to N. These operations must be repeated until the queue becomes empty, thus the outer loop is run $O(N)$ times, leading to a total of $O(NK)$. We state that K weakly depends on N, and this statement will be proved in the results section. Thus we approximate in this analysis K as a constant factor, independent of N. Finally, the scheduler must compute the queue size bonus at the beginning of the slot, and this is an $O(N)$ operation. Therefore the computational complexity is linear in the network size N.

Since we assume that the scheduler has perfect channel state information, scheduling errors (that is to say, some links turn out to have an insufficient SIR) cannot happen in our settings.

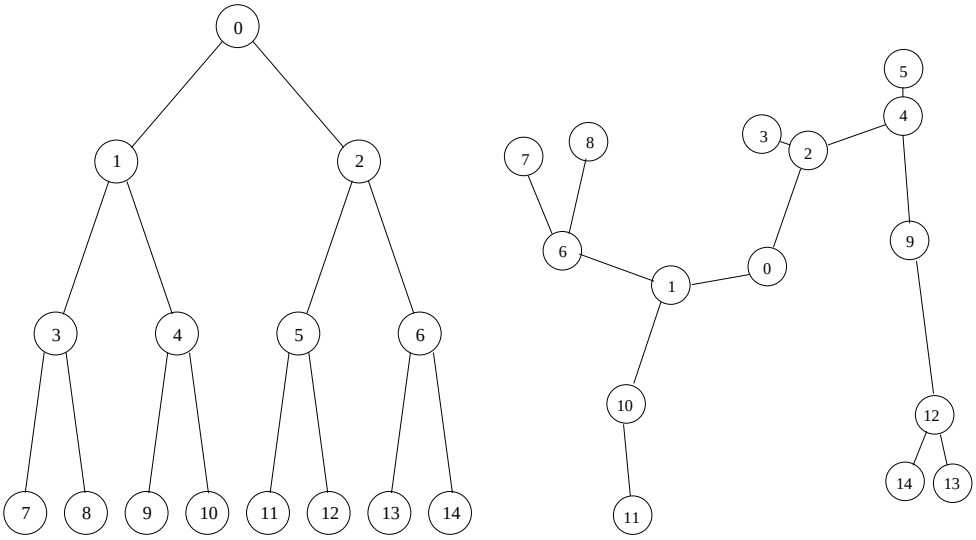

Figure 3.25. *A sample regular (a) and random (b) tree topology*

3.6.3 Performance Evaluation

We have quantified the performance of our scheduler in a number of different situations. Our goal was to compare the absolute performance of our proposed scheduler against some recognized benchmarks and to explore which factors impact its performance.

Scenario Description — All our tests have been run on tree topologies composed by a number of nodes ranging from 16 to 31. For all links, we assume a path loss proportional to $d^{-3.5}$, where d is the distance between transmitter and receiver. Additionally, we superimpose a correlated shadowing term modeled as in [39], with a variance of 5 dB and a correlation at 100 m equal to 0.6. Moreover, two classes of topologies have been created. The first type corresponds to the regular node deployment depicted in Fig. 3.25a. In such a case, each node is 300 m away from its next hop, and the tree is binary and balanced (the difference between the depth of any two leaves is at most one). Observe that, even though the nodes' positions are fixed and regular, the presence of the correlated shadowing, which is introduced in all the investigated topologies, allows us to obtain different values of the path gain for each topology instance. In the second class of topologies, nodes are randomly placed in a 1000 m $\times$ 1000 m square. A tree topology is generated by means of a spanning tree algorithm which chooses the closest node to the center of the square as the root, and allocates children nodes to the already built tree, with a limit on the node degrees set to 3. An example is shown in Fig. 3.25b. Differently from the previous scenario, the tree is no longer binary and balanced. In spite of these differences, most of the conclusions we derive for these two scenarios are quite similar, so we infer that they are likely to hold true for other cases as well.

Given an N node topology, an $N-1$ node topology is created by removing a leaf picked at random. According to this procedure, given a base tree consisting of 31 nodes (i.e., a full tree where all leaf nodes have depth 4), we generate a sequence of smaller topologies by successively removing

one leaf, until as few as 16 nodes are left (i.e., exactly one leaf node has depth 4). All the curves reported in the numerical results are averaged over 30 different samples, which ensures adequate statistical confidence. Where meaningful, 95% confidence intervals are reported.

The performance of our link scheduling algorithm is assessed by means of the following indices:

- *schedule length*: the duration of the schedule produced by the algorithm in number of slots.
- *end-to-end system throughput* (or simply *throughput*): the overall amount of net user data delivered by the system per unit time.
- *fairness index*: $(\sum_{i=0}^{n} x_i)^2 / \left(n \sum_{i=0}^{n} x_i^2\right)$ [40], where n denotes the number of data flows to the gateway, and x_i the throughput of the *i-th* flow. By definition, the fairness index is bounded in [0, 1] and for equal partitioning of bandwidth is equal to 1.

To investigate the performance of our algorithm in terms of schedule length, we assume that nodes have an integer number of packets in their queues (whose initial sizes need not be equal) and the link rates are normalized to 1. All nodes transmit at a fixed power of 10 dBm. The goal of the scheduler is to transfer all data as quickly as possible from the nodes to the tree root/gateway. If not stated otherwise, the interference model employs $K = 2$ dominant interferers, the suppression factor α is 0 and each node initially has 8 packets in its buffer. The number of dominant interferers K was chosen to be 2 because for higher K performance improvement was found to be negligible. Such a low K strikes a good balance between computational complexity and performance. Moreover, especially in random topologies, it is hard to find many strong interferers which generate comparable interference so that they should all be regarded as dominant. Hence a higher K does not lead to significant performance improvement in this setting. The set of results will explore the scheduler performance as a function of network size, SIR threshold and K.

In addition, the system throughput is analyzed under realistic traffic conditions. Two types of data traffic are used in the simulations, namely Web and Constant Bit Rate (CBR). In the former case, traffic is modeled as a Web source generating variable size packets at variable inter-arrival times. The packet size is distributed as a truncated Pareto random variable with location 10.3 kB, shape 1.1, and cut off 1500 kB. Packet inter-arrival time is exponentially distributed. In the latter case, the source produces packets with length equal to 1000 B at a constant average rate of 50 kB/s. The analysis was carried out by means of Network Simulator 2 (ns-2) [35]. Note that, since we deal with physical realism of the interference models, we utilized, within ns-2 simulation, a more detailed implementation of the physical level, including in particular the additive behavior of the interference. This means that, according to Section 3.4.1, the packets need an SIR above the threshold Λ_0 at the receiver's side to be correctly decoded.

Numerical Results — The first test compared the time to empty the node queues for four different systems: our proposed scheduler, the optimal link activation, the protocol model and the physical-model based scheduler by Brar *et al.* [34]. The scheduler by Brar *et al.* is the present benchmark for scheduling based on the physical model. The protocol model is implemented as in [34]; that is to say, whenever a tagged link is activated, it silences all other links whose transmitter or receiver lie inside

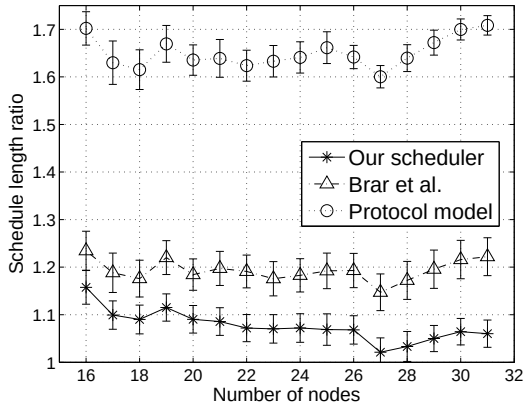

Figure 3.26. *Performance comparison for the regular topologies,* $\Lambda_0 = 2.5$ *dB,* $K = 2$

the interference radius to the tagged link receiver. Observe that, whereas our proposed scheduler and also the one by Brar *et al.* mandatorily verify the feasibility condition for the SIR being above Λ_0 for each activated link, the protocol model, which performs just an approximate computation of the interference, may instead obtain infeasible link activation patterns. When this happens, we assume that an ideal ARQ recovery mechanism is available, which means that the erroneous packets are always detected and immediately (i.e., without delay) notified at the transmitter, which can retransmit them already in the next time slot. This is clearly an optimistic assumption, so the behavior of the protocol model is overestimated; actually, in practical environments, a realistic error recovery mechanism would imply an even worse performance.

The optimal link activation sequence is found by means of an exhaustive search over all possible schedules that are feasible under the physical model.

Figs. 3.26 and 3.27 report the results for the regular and random topologies, respectively. They show the ratio between the lengths of the schedules computed by the different approaches and the optimal schedule length. Note that all of the approaches achieve an approximately linearly increasing schedule length in the number of nodes, but with different slopes, that is to say the heights of the curves in Figs. 3.26-3.27 (which is the most interesting aspect, as it tells us also how the scheduling algorithms scale with the network size). First of all, the curve relative to our scheduler with queue bonus is usually within 1.1. This means that our schedule is about 10% longer than the optimal one (and often less than that). This is a non-trivial result, since there is no easy way to predict the performance of our scheduler, which could have been anywhere between the lower bound and the protocol model. This fact points out that our algorithm can harness the potential spatial reuse and achieve results which are very close to the optimal scheduling. Moreover, the performance of the algorithm by Brar *et al.* is 20% worse than the optimal schedule. Therefore, we are able to halve the gap with the lower bound, and we can often do better. Incidentally, Brar *et al.* proved that the performance of their algorithm was within a constant multiplicative factor from the optimal schedule. This is confirmed by our graphs. Finally, the protocol model performs rather poorly, because of the low degree of spatial reuse.

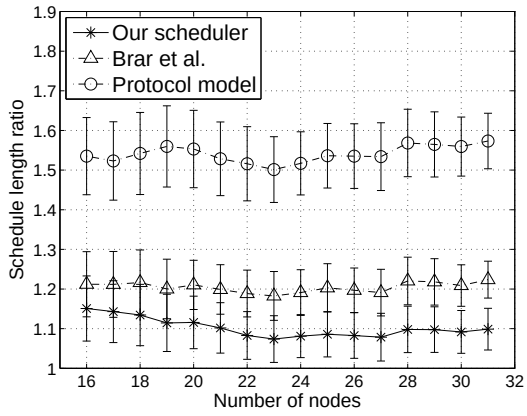

Figure 3.27. *Performance comparison for the random topologies,* $\Lambda_0 = 2.5$ *dB,* $K = 2$*,* $\alpha = 0$

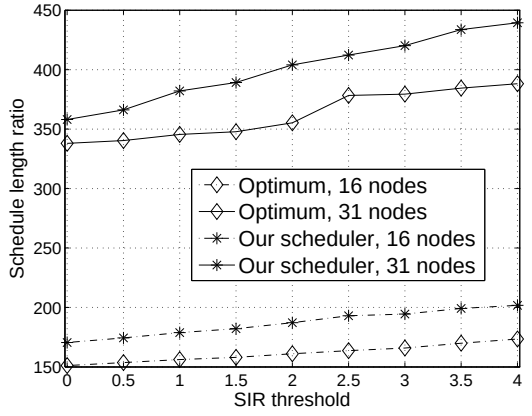

Figure 3.28. *Performance dependence on the decoding threshold,* $K = 2$*,* $\alpha = 0$

Similar reasoning can be applied to Fig. 3.27, where the random topology case is considered. Again, our scheduler's curve is very close to the optimal policy and thus confirms the adaptability of our method to realistic topologies. In this situation, the algorithm by Brar *et al.* does not significantly improve its performance over the previous case, while we approach the lower bound more tightly.

Fig. 3.28 compares the performance of the optimum link activation sequence and our scheduler when the target SIR is changed, in the regular topology case. We note that the curve corresponding to our scheme remains close to the optimal one for all the SIR values. This shows that our approach is robust to the SIR choice. The schedule length increases with a higher SIR because the lower tolerable level of interference decreases the spatial reuse.

A key issue for our scheduler is the determination of the minimum number of dominant interferers used by the interference model necessary for satisfactory performance. It is reasonable to expect that the more the dominant interferers, the better the scheduler performance because the interference model becomes more accurate. However, the computational complexity increases. Fig. 3.29 explores this tradeoff for the regular topology when the queue size bonus is set to zero. This bonus has been removed in this context because we want to study the influence of the interference model accuracy in

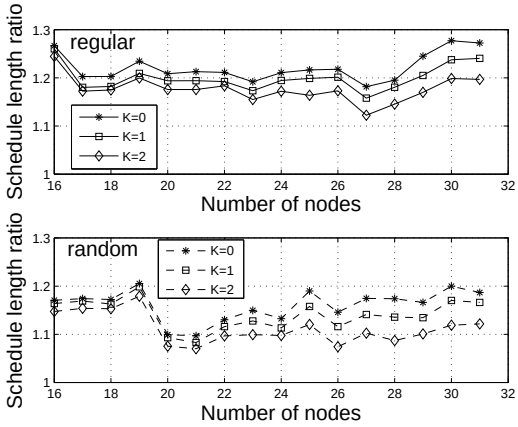

Figure 3.29. *Dependence on the number of dominant interferers, for regular (above) and random (below) topologies,* $\Lambda = 2.5$ *dB,* $\alpha = 0$

isolation. The queue size bonus can mask the different impacts and so it has been turned off for these experiments. Therefore, the curves here do not extend those in the previous graphs (e.g., Fig. 3.26 or 3.27). First of all, the mixture model ($K > 0$) yields non negligible improvement over the single mode model ($K = 0$) only for rather large networks (at least 20 nodes). The reason is the following: as pointed out in [13], the Gaussian mixture model works well when there are a few nodes whose power received by a certain terminal is larger than all the rest of the combined interference. When the nodes are few (less than 20) they are usually confined in a small area, and the range of powers received at any point in the tree from all the nodes is within one order of magnitude. Thus, the final interference is not multimodal, but can be already predicted fairly well by a simpler single-mode Gaussian estimator. However, for larger networks dispersed in a wider region, the ratio of the powers between two interferers may become significantly large and result in some noticeable performance difference. This is only partially captured by the average schedule length. In fact, due to the choice of a tree topology, the main bottleneck of the delivery is the tree root, which is independent of the interference evaluation. Thus, also other quantities such as the second order moments should be considered. In any case, the reported difference is about 5–10%. Also note that K is quite low, because no significant performance improvements can be achieved with higher K. Actually, the curves for $K \geq 3$ are not plotted because they are almost indistinguishable from the case $K = 2$. The graphs suggest that the dependence of K on the network size is weak. We conjecture that it is in fact sublinear, but further investigation is still needed in the area. Finally, observe also that the slope of the curves changes, thus we infer that for larger networks the gap would increase, which is also confirmed by preliminary results.

We have also explored how the scheduler performance changes when the queue size is modified. In particular, all our previous simulations considered all nodes to be equally backlogged. We have tested two more scenarios: in the former case, the nodes closer to the root have a longer queue than the leaves, and vice versa for the latter. In the first case the scheduler length is on average shorter (the packets are closer to the root) and the opposite happens in the latter case. But no matter what the load

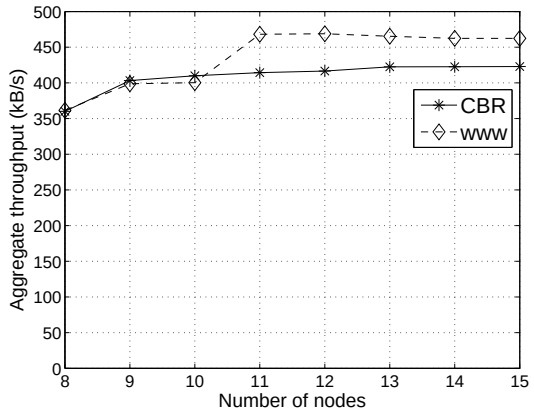

Figure 3.30. *Performance of our scheduler with web-browsing and CBR flows,* $\alpha = 0.5$.

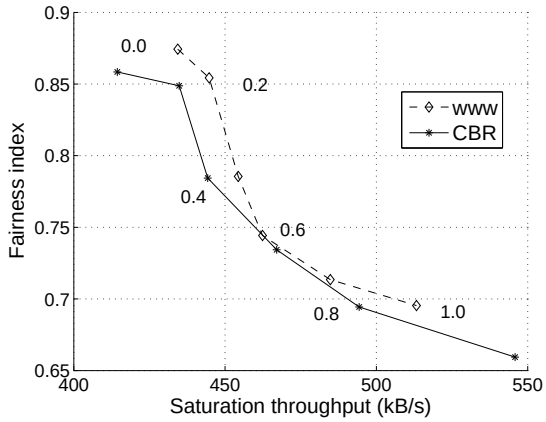

Figure 3.31. *Performance of our scheduler for different values of the suppression factor* α.

distribution could be, the gap between the optimal scheduling and our system is always in the order of 10% as in the previous cases. Therefore our scheduler is robust to the backlog location.

So far we have proved that our system offers excellent performance compared to the optimal scheduler and [34]. We complete our study by an analysis of our scheduler performance with realistic traffic sources, CBR and Web by means of ns-2.[6] In addition, further insight about the dependence of the scheduler behavior with respect to its fairness parameter α has been sought. In Fig. 3.30 we have studied the system throughput as the tree size is increased from 8 to 15 nodes. We note that the throughput already saturates at 15 nodes, so we have not analyzed larger networks. Each link has a data rate of 1 MB/s, and the maximum possible capacity in the tree topology is exactly 1 MB/s, because only one of all the links that go into the root can be active per slot. Our scheduler achieves around 45% of this value, which is a significant result considering the interference and topology constraints. As can be seen, when the network becomes overloaded, i.e., the number of nodes is greater than or equal to 11, the overall throughput achieved with the Web source is slightly higher

[6]We point out that for web-browsing, UDP has been used as transport protocol, because TCP excessively influences the system performance and its impact on protocol activity would cancel many of the phenomena we are interested in.

than that with CBR. This is mainly due to the fact that flows experience random bursty arrivals of packets followed by periods of inactivity. Therefore, it may happen that not all flows are active at the same time and traffic experiences a better statistical multiplexing. Fig. 3.31 shows the throughput in overload conditions versus the fairness index while changing the suppression factor α. We note that there is a tradeoff between the two. This is due to the fact that when α is large (close to 1) all nodes are eligible to be scheduled. This implies that those link activations that enjoy a high spatial reuse may be used very often, and thus the throughput will eventually benefit. However, this also favors those nodes whose interference is inherently low because of their position. Therefore the fairness index will drop. It is also evident that CBR traffic is more affected by α. We believe that this fact is due to the time distribution of the packet arrivals: Web traffic is bursty, and thus terminals in unfavorable positions just have to wait for some traffic to be delivered before having their chance to transmit. On the other hand, CBR will keep busy those nodes in low interference locations, and thus the fairness-throughput curve will be shifted toward the low-right corner of Fig. 3.31. Incidentally, we observe that when no constraints on node selection are imposed ($\alpha = 1$) the system fairness is nonetheless acceptable (0.65). On the other hand, the ratio between the maximum and minimum saturation throughput is 0.78. This means that even when the candidate selection is strict ($\alpha = 0$) the achieved throughput is still a significant fraction of its best possible value. Hence the scheduler achieves simultaneously high fairness and throughput. Moreover, it is really possible to trade off the two quantities (the curve is smooth and there is no sudden change as the suppression factor changes) and thus α is a design parameter that can be tuned to achieve a desired point in the tradeoff curve. All these observations and findings lead us to concluding that the proposed scheduler is flexible and can perform well in a wide range of scenarios.

3.6.4 Discussion

We have proposed a high performance centralized scheduler for wireless multi-hop networks based on the physical interference model, rather than the protocol model. The scheduler low computational complexity model makes it attractive for these networks, due to its robustnes to some important parameters (like the backlog distribution throughout the nodes) and to the ability to outperform state-of-the-art schedulers. In some cases, the gap with the theoretical optimum is halved.

3.7 Chapter Conclusions

The first part of this Chapter dealt with carrier sense for MIMO ad hoc networks. The performance of the proposed CS mechanism has been studied as a function of some design parameters (like the training sequence length) as well as some environment conditions (SNR, traffic load). We have shown in which scenarios the system works well and also suggested improvements on the present version.

We have analyzed how a simple Gaussian model can be used to estimate the interference behavior in a wireless ad hoc network for carrier sense threshold optimization. A Gaussian mixture model can correctly predict the network behavior and enable a quick optimization of the network parameters in

a static setting (i.e., the node placements and path losses are known). Moreover, whenever the carrier sense threshold has to be adjusted dynamically, we have shown (by analysis and simulation) that it depends linearly on the generalized node density defined as N/R^{α}, where N is the number of nodes in the network, α is the path loss coefficient and R is the network radius. We exploit this relationship by building a simple, fully distributed and effective algorithm for CS adaptation, that offers significant advantages over conventional IEEE 802.11 and is competitive with the state-of-the-art CS adaptation algorithm in [9]. Our future work will study the interaction between transmission power control and CS tuning and the performance evaluation of our system when important parameters such as the path loss coefficient are incorrectly estimated.

Finally, also the field of mesh scheduling can benefit from the proposed carrier sense model. It is possible to design a high performance centralized scheduler for wireless multi-hop networks based on the physical interference model, rather than the protocol model. The scheduler is based on a low complexity model for aggregate mutual interference between nodes, whose complexity is linear with the network size. We have evaluated the robustness of our algorithm with respect to some important system parameters (detection SIR and backlog distribution in the network) and we have shed some light on the dependence of the scheduler's performance on some of its parameters (number of dominant interferers or suppression factor). We are also able to outperform other proposed models which represent the benchmark for computationally efficient, physical-model based schedulers, achieving a gain larger than 50% in approaching the theoretical optimum.

Our future work will study the relaxation of some assumptions (for instance the perfect channel state information assumption). Moreover we are working toward a distributed version of our algorithm, for which problems like disseminating information about queue sizes, obtaining the status of the dominant interferers or coping with imperfect channel state estimates in the previous slot must be solved.

3.8 Acknowledgments

The first part of this chapter is the gist of Emanuele Coviello's master thesis, who worked under Prof. Zorzi and the author's supervision. In addition, the contribution of Abhijeet Bhorkar (UCSD) and Prof. Bhaskar Rao (UCSD) has been essential. The final part of this work (a low complexity scheduler for mesh networks) has been developed with Alessandro Erta, Leonardo Badia and Prof. Luciano Lenzini, all from the IMT Lucca.

References

[1] J. Deng, B. Liang, and P. K. Varshney, "Tuning the Carrier Sensing Range of IEEE 802.11 MAC," in *IEEE GLOBECOM*, Dallas, (TX, USA), Dec. 2004, pp. 2987–2991.

[2] X. Yang and N. Vaidya, "On Physical Carrier Sensing in Wireless Ad Hoc Networks," in *IEEE INFOCOM*, Mar. 2005, pp. 2525–2535.

[3] H. Zhai and Y. Fang, "Physical Carrier Sensing and Spatial Reuse in Multirate and Multihop Wireless Ad Hoc Networks," in *IEEE INFOCOM*, Barcelona (Spain), Apr. 2006.

[4] A. Hasan and J. G. Andrews, "The Critical Radius in CDMA Ad Hoc Networks," in *IEEE GLOBECOM*, Dallas (TX, USA), Dec. 2004, pp. 805–808.

[5] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan, "Understanding the real-world performance of Carrier Sense," in *ACM SIGCOMM*, Philadelphia, (PA, USA), Aug. 2005, pp. 122–126.

[6] M. M. Carvalho and J. J. Garcia-Luna-Aceves, "A Scalable Model for Channel Access Protocols in Multihop Ad Hoc Networks," in *ACM MOBICOM*, Philadelphia (PA, USA), Sep. 2004.

[7] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling Per-Flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks," in *IEEE INFOCOM*, Barcelona (Spain), Apr. 2006.

[8] E. Wong and R. Cruz, "A Spatio-Temporal Model for Physical Carrier Sensing Wireless Ad-Hoc Networks," in *IEEE SECON*, Reston (VA, USA), Sep. 2006.

[9] J. Zhu, X. Guo, L. L. Yang, W. S. Conner, S. Roy, and M. M. Hazra, "Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks," *Wiley Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 933–946, Dec. 2004.

[10] M. Zorzi, J. Zeidler, A. Anderson, B. Rao, J. Proakis, A. L. Swindlehurst, M. Jensen, and S. Krishnamurthy, "Cross–layer issues in MAC protocol design for MIMO ad hoc networks," *IEEE Wireless Commun. Mag.*, vol. 13, no. 4, pp. 62–76, Aug. 2006.

[11] M. Levorato, P. Casari, S. Tomasin, and M. Zorzi, "Physical Layer Approximations for Cross-Layer Performance Analysis in MIMO-BLAST Ad Hoc Networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4390–4400, Dec. 2007.

[12] K. Sundaresan, R. Sivakumar, M. Ingram, and T.-Y. Chang, "Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 350–365, Oct. 2004.

[13] J.-S. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: Enabling Spatial Reuse using MIMO channel-aware MAC," in *IEEE ICC*, Seoul (Korea), May 2005.

[14] J. C. Mundarath, P. Ramanathan, and B. Van Veen, "A Cross Layer Scheme for Adaptive Antenna Array based Wireless Ad hoc Networks in Multipath Environments," *ACM/Kluwer Journal on Wireless Networks*, vol. 13, no. 5, May 2006.

[15] A. Bhorkar, B. S. Manoj, and B. Rao, "Selection Diversity based MAC Protocol for MIMO Ad hoc Wireless Networks," in *IEEE GLOBECOM*, New Orleans (LA, USA), Dec. 2008.

[16] G. W. Stewart, "Determining Rank in the Presence of Error," University of Maryland, College Park, Tech. Rep. TR-92-108, 1992.

[17] G. Orfanos, J. Mirkovic, H.-J. Reumerman, and D. Denteneer, "A MAC Protocol for MIMO Based IEEE 802.11 Wireless Local Area Networks," in *IEEE WCNC*, Hong Kong (China), Mar. 2007.

[18] J. Mirkovic, H.-J. Reumerman, and D. Denteneer, "A MAC Protocol with Multi-user MIMO support for Ad Hoc WLANs," in *IEEE PIMRC*, Athens (Greece), Sep. 2007.

[19] M. K. Todd and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River (NJ, USA): Prentice Hall, 2000.

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

[21] V. Petrov, *Limit Theorems of Probability Theory, Sequences of independent random variables*. Oxford: Oxford Science Pubs., 1996.

[22] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[23] G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the Number of competing Terminals in an IEEE 802.11 network," in *IEEE INFOCOM*, S. Francisco (CA, USA), Apr. 2003.

[24] B. Hajek and G. Sasaki, “Link scheduling in polynomial time,” *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 910–917, Sep. 1988.

[25] L. Tassiulas and A. Ephremides, “Jointly optimal routing and scheduling in packet radio networks,” *IEEE Trans. Inf. Theory*, vol. 38, no. 1, pp. 165–168, Jan. 1992.

[26] M. Kodialam and T. Nandagopal, “Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem.” in *ACM MOBICOM*, San Diego (CA, USA), Sep. 2003, pp. 42–54.

[27] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu., “Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode,” in *ACM MOBIHOC*, Urbana-Champaign, (IL, USA), May 2005, pp. 78–89.

[28] L. Badia, A. Erta, L. Lenzini, and M. Zorzi, “A general framework of interference models for joint routing and link scheduling in wireless mesh networks.” *IEEE Netw.*, vol. 22, no. 1, pp. 32–38, Jan. 2008.

[29] A. Iyer, C. Rosenberg, and A. Karnik, “What is the right model for wireless channel interference?” in *Proc. Qshine*, Waterloo, (ON, Canada), Aug. 2006, pp. 805–808.

[30] J. Jun and M. Sichitiu, “The nominal capacity of wireless mesh networks,” *IEEE Commun. Mag.*, vol. 10, no. 5, pp. 8–14, May 2003.

[31] N. Ben Salem and J.-P. Hubaux, “A fair scheduling for wireless mesh networks,” in *WiMesh*, Santa Clara, (CA, USA), Sep. 2005.

[32] P. Djukic and S. Valaee, “Distributed link scheduling for TDMA mesh networks,” in *IEEE ICC*, Glasgow (UK), Jun. 2007.

[33] M. Kodialam and T. Nandagopal, “Characterizing the capacity region in multi-radio multi-channel wireless mesh networks.” in *ACM MOBICOM*, Cologne (Germany), 2005, pp. 73–87.

[34] G. Brar, D. Blough, and P. Santi, “A novel multiple access scheme over multi–packet reception channels for wireless multimedia networks,” in *ACM MOBICOM*, Los Angeles, (CA, USA), Sep. 2006, pp. 2–13.

[35] “Network simulator 2.” [Online]. Available: http://www.isi.edu/nsnam/ns/

[36] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, “Impact of interference on multi-hop wireless network performance,” *Wireless Networks*, vol. 11, no. 7, pp. 471–487, Jul. 2005.

[37] J. Chang and L. Tassiulas, “Routing for maximum system lifetime in wireless ad-hoc networks,” in *Allerton Conference*, Monticello, (IL, USA), Sep. 1999.

[38] S. Basagni and M. Conti and S. Giordano and I. Stojmenović (editors), *Mobile Ad Hoc Networking*. New York: John Wiley and Sons, Inc., 2004.

[39] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electronics Letters*, vol. 27, pp. 2145–2146, Apr. 1991.

[40] R. Jain, W. Hawe, and D. Chiu, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” Digital Equipment Corporation, Tech. Rep. DEC-TR-301, Sep. 1984.

[41] F. Rossetto and M. Zorzi, “Gaussian approximations for carrier sense modeling in wireless ad hoc networks,” in *IEEE GLOBECOM*, Wahington (DC, USA), Nov. 2007, pp. 464–473.

Chapter 4

Delay and throughput characterization of random multi-hop networks

Contents

4.1 Introduction 112
4.2 System model 113
4.2.1 Network setting 113
4.2.2 Physical layer 115
4.2.3 Network metrics 115
4.3 Analysis 116
4.3.1 Queueing analysis 116
4.3.2 Evaluation of packet success probabilities 118
4.3.3 Examples 122
4.4 Simulation Results 125
4.5 Chapter Conclusions 128
4.6 Acknowledgments 129
References 129

4.1 Introduction

One of the main design issues in multi-hop wireless networks is determining the number of hops between the source of information and the final destination [1]. For a given transmission power, a smaller hopping distance results in a larger received signal-to-interference-and-noise-ratio (SINR), which implies more reliability and/or a higher transmission rate over a single hop. However, as argued in [1], this does not necessarily translate to an end-to-end performance benefit, e.g., in terms of delay: each node that is added between the source of information and the final destination is also the cause of additional delay, since a packet has to be decoded, encoded and wait in the queue, before it is transmitted to the next node. Moreover, if the intermediate nodes can transmit only one at a time, e.g., to avoid intra-route interference, the throughput might suffer as well. Along the line of thought in [2], a meaningful performance analysis and the design of multi-hop ad hoc networks should be carried out with end-to-end constraints on delay and reliability in mind.

To date, excluding the literature on capacity scaling laws [3], we can divide the work on multi-hop networks into two categories. In the first one [4–8] (see the review paper [9] for a comprehensive list of references), a snapshot of the multi-hop network, i.e., a single hop of a typical route, is considered, with the implicit assumption that the destination lies at an infinite distance from its source. Single-hop metrics are devised and evaluated, that are related to a performance benefit at the end-to-end level. Such metrics include: the expected progress, i.e., the product (packet success probability) $\times$ (hop length), for a given spatial density of transmitters, which reflects the trade-off outlined in the previous paragraph [4]; the transmission capacity, i.e., the maximum density of transmitters allowed under a constraint on the success probability, for a given hop length [6]; and the spatial density of progress [7], i.e., the product (spatial density of successful transmissions) $\times$ (hop length) which is a logical combination of the previous two metrics. A central assumption in these papers is that the transmitters constitute a Poisson random process in the plane. This allows a precise characterization of the SINR statistics, hence the derivation of analytical results that demonstrate the effect of the channel (fading, interference and noise) and various physical layer parameters on the above network-wide metrics.

The second body of work [10–13] is, in a sense, complementary to the first. A well defined route is considered, where the distance to the final destination and the number of intermediate nodes, or *relays*, are specified. However, the impact of interference from other transmissions in the network is ignored. Assuming a channel model with path-loss, fading and noise, and no delay constraints, [11, 12] tackle the capacity-like problem of determining the end-to-end rate, i.e., the minimum achievable rate over all hops, when a TDMA-access protocol is employed. Alternatively, under a given delay constraint, [13] determines the number of hops and the rate allocation among them, such that the total power consumption is minimized. A similar problem is studied in [10], under an end-to-end success probability requirement.

Our work bridges the aforementioned research directions. We consider a random Poisson network with finite-distance, mutually interfering routes. To accomodate the randomness in the delivery of a packet over each hop, nodes are equipped with queues. A simple link-layer protocol is assumed,

Table 4.1. *Commonly used symbols*

Symbol	Meaning
p	source MAP
p_R	relay MAP
p_A	probability of new packet arrival at the source
λ	density of sources
p_n	success probability at the n^{th} hop
r_n	distance of n^{th} relay from its source

where, if a packet is not received correctly by a node, it is retransmitted by the previous node in the route at the next available opportunity. Central to our analysis is the derivation of conditions such that the node queues are stable, i.e., their lengths remain bounded over time and, as a result, the delay over a typical route is finite. Using an outage model for the packet success probability, we analytically evaluate the mean end-to-end delay, as well as the route/network throughput, and determine the answers to the following questions: what are the number of relays and their placements that minimize the delay; what MAP or traffic control must be enforced on the sources such that the delay remains finite, irrespective of the relay placement; and, finally, what is the impact of shifting the relays from their optimal positions. In face of simplifying model assumptions, we would like to emphasize that the contribution of this work is for the most part analytical, with the intent of providing design insights for practical multi-hop networks. A simulator is set up to verify the findings resulting from our model.

The rest of this Chapter is organized as follows. The system model is described in Section 4.2, and Section 4.3 includes our analysis. Numerical examples and simulation results are presented in Section 4.3.3 and Section 4.4, respectively. Our conclusions are summarized in Section 4.5. A list of symbols commonly used throughout the Chapter is given in Table I.

4.2 System model

4.2.1 Network setting

We consider a network composed by an infinite number of *routes* or *flows* on an infinite plane (see Fig. 4.1). Each route comprises a source, a destination at distance R, and $N-1$ relays on the line defined by the source-destination pair. We refer to sources, relays or destinations indiscriminately as *nodes*. The distance of the n^{th} relay, $n = 1, \ldots, N-1$, from its source is the same for all routes and denoted by r_n (with a slight abuse of notation, $r_0 = 0$ and $r_N = R$). The length of the n^{th} hop in the route is therefore $r_n - r_{n-1},\ n = 1, \ldots, N$. A typical N-hop route is shown in Fig. 4.2.

Each node has an infinite queue, where packets that are received from the previous node in the route can be stored in a first-in, first-out fashion. The source, in particular, is not preceded by a node and "receives" a packet from an upper layer of the protocol stack every N slots with probability p_A (when $p_A = 1$, the source is backlogged). Time is divided into packet slots and the following

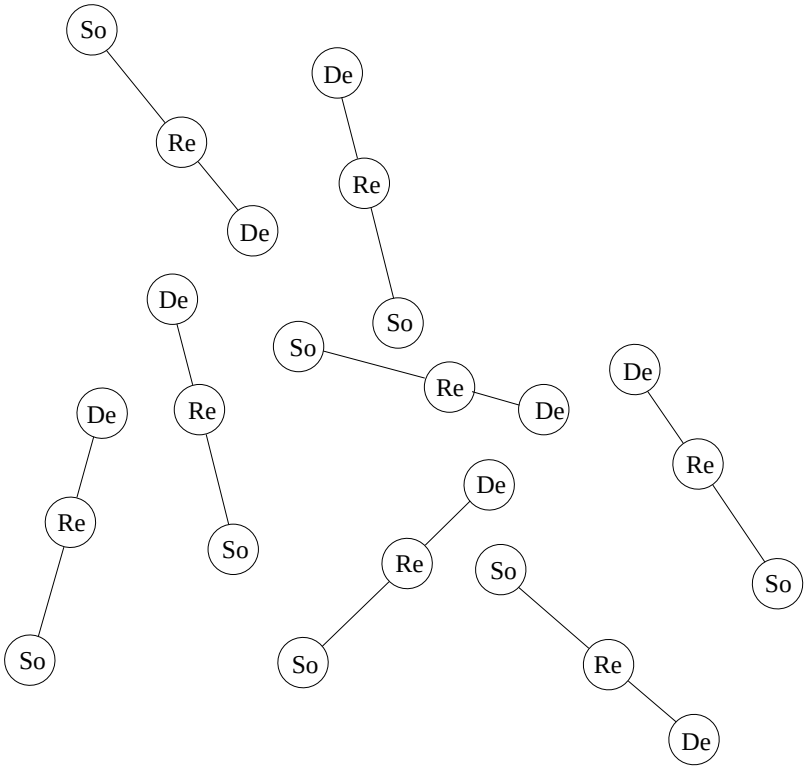

Figure 4.1. *A two-hop random network. So stands for source, Re for relay and De for destination. Note that the relay is not necessarily placed in the middle between source and destination.*

TDMA/ALOHA synchronous protocol is observed: the nodes at distance r_n from their source are allowed to transmit a packet, with a certain probability, only at times $n + mN,\ m \in \mathbb{N}$, i.e., the sources at slots $0, N, 2N, \ldots$, the 1^{st} relays at slots $1, 1 + N, 1 + 2N, \ldots$ and so on. A packet is received successfully by a node in the route, if the SIR in that slot is above a target threshold. If it is not, the transmitting node is informed via an ideal feedback channel and the packet remains at the head of its queue, at least for the next N slots, until its turn to attempt a retransmission. For simplicity, we assume that, upon the assigned TDMA slot, a source transmits with probability p (p-persistent sources) and a relay with probability p_R (p_R-persistent relays). Note that the main characteristic of the protocol is that intra-route interference is avoided, by having all other nodes be silent when a given node in the route is transmitting. Since we are focusing on routes with only a few hops, permitting intra-route spatial reuse would only yield a marginal performance benefit, at the price of increased complexity [12]. Moreover, we consider relays that do not generate traffic, i.e., their sole function is to forward the packets toward their final destination.

We assume that the network topology is such that, in every slot, the locations of the sources are drawn independently according to a spatially homogeneous Poisson process of intensity λ [6, 7] and the orientation of each destination with respect to its source is changed randomly. These assumptions ensure that the interference levels encountered over different hops/slots are randomized and will be discussed at greater length in Section 4.4.

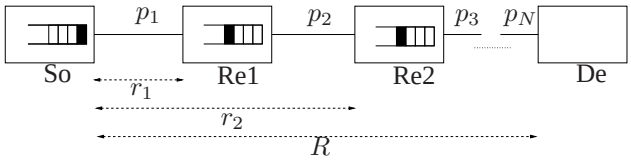

Figure 4.2. *An N-hop route.*

4.2.2 Physical layer

The channel between any two nodes at distance r includes Rayleigh fading and path-loss according to the law r^{-b}, where $b > 2$ is the path-loss exponent. All nodes have the same transmit power, which is normalized to one. In order to place emphasis on the interactions between the desired and interfering signals, we consider an interference-limited setting, i.e., thermal noise is considered negligible and disregarded. The SIR at the receiver of the n^{th} node of a given route is

$$\text{SIR}_n = \frac{A_n (r_n - r_{n-1})^{-b}}{\sum_{t \in \mathcal{T}_{n-1}} A_t d_t^{-b}},$$

where A_n is the fading coefficient between the $(n-1)^{\text{th}}$ and the n^{th} node and exponentially distributed with unit mean; $\mathcal{T}_{n-1}$ is the set of interfering nodes at the n^{th} hop; d_t is the distance between the interfering node t and the n^{th} node of the chosen route and A_t is the respective fading coefficient.

We assume that a packet is successfully received when the SIR is above a target threshold θ.[1] We can then define the probability of successful packet reception by the n^{th} node, p_n, over different fading and network realizations, as $p_n = \mathbb{P}(\text{SIR}_n > \theta)$. Under the network model described in the previous section, $\mathcal{T}_{n-1}$ can be considered as a Poisson process of interfering nodes with density λ_{n-1}. Using the result in [7], we have that

$$p_n = \mathrm{e}^{-\lambda_{n-1} c (r_n - r_{n-1})^2}, \; n = 1, 2, \ldots, N, \tag{4.1}$$

where c is a physical layer dependent constant given by

$$c = \Gamma(1+\alpha)\Gamma(1-\alpha)\pi\theta^{\alpha}, \tag{4.2}$$

with $\alpha = 2/b$ as the *stability exponent* [4]. Eq. (4.1) is the starting point of the analysis presented in Section 4.3.

4.2.3 Network metrics

Our metric of interest is the mean end-to-end delay D, i.e., the mean total time (in slots) that it takes a packet to travel from the source to the destination in a typical route. Ignoring propagation times, D is the sum of the mean *waiting times* and *service times* along the queues of the route. The waiting time at a given node is measured starting from the moment a packet arrives at that node's

[1] This implies that the receiver regards all interference as noise and no interference cancellation or multi-user detection techniques are employed.

queue, till it becomes the head-of-line packet, i.e., all packets in front of it have been successfully transmitted to the next node. The service time is measured from the moment a packet reaches the head of the queue, till it is successfully received by the next node. Mathematically,

$$D = \sum_{n=0}^{N-1} (Q_n + H_n), \tag{4.3}$$

where $Q_n,\ H_n$ stand for mean waiting and service times at node n, respectively. Note that, if the sources are backlogged, it is only meaningful to talk about end-to-end delay for a packet at the head of the source queue. Hence, only a mean service time H_0 is defined at the source queue.

In addition to the delay, we are interested in evaluating the Route Throughput (RT), defined as the expected number of packets successfully delivered to the destination per slot, over a typical route. At the network level, a metric of interest is the Network Throughput (NT) [7], defined as $\mathrm{NT} = \lambda \mathrm{RT}$.

4.3 Analysis

We begin our analysis by evaluating D, given the packet success probabilities $\{p_n\}$. The second part and third part of this section are devoted to the evaluation of these probabilities and the presentation of numerical examples, respectively.

4.3.1 Queueing analysis

The mean waiting and service times at a given node are functions of the packet arrival and departure probabilities to and from that node. Assume that the sources are backlogged and we are looking at the queue of the 1st relay. A packet arrival occurs at the end of slot mN, with probability $p_I = pp_1$. Provided that the queue is not empty, at the beginning of slot $1 + mN$, a packet departs from the head of the queue with probability $p_O = p_R p_2$. This procedure is repeated after N slots.

The queue of the relay is modelled as a Random Walk [14, 15], whose state is the number of packets in the queue at the end of slot mN. The queue is stable only when $p_I < p_O$. Under this requirement, the transition probability from state k to state $k+1$, $k \geq 1$, is $p_I(1-p_O)$, while it is $p_O(1-p_I)$ from state $k+1$ to state k, for $k \geq 0$. Since in state 0 the queue is empty, the transition probability from state 0 to state 1 is simply p_I. The steady state probability of being in state k, π_k, is

$$\pi_k = \frac{p_I}{p_O(1-p_I)} \cdot \rho^{k-1} \cdot \pi_0,\ k \geq 1, \tag{4.4}$$

where

$$\pi_0 = 1 - \frac{p_I}{p_O} \tag{4.5}$$

and

$$\rho = \frac{p_I(1-p_O)}{p_O(1-p_I)}.$$

Note that the relay queue is non-empty with probability p_I/p_O.

The mean waiting time at the 1^{st} relay, Q_1, can be computed by Little's theorem, as the average queue size, excluding the head-of-line packet, divided by the arrival rate [14], in this case p_I/N. Using (4.4), the average queue size is found to be

$$\sum_{k=1}^{+\infty}(k-1)\pi_k = \frac{p_I^2}{p_O}\frac{1-p_O}{p_O-p_I},$$

therefore,

$$Q_1 = N\frac{p_I}{p_O}\frac{1-p_O}{p_O-p_I}.$$

It is also straightforward to show that the service time for the head-of-line packet is

$$H_1 = \frac{N}{p_O} - N + 1,$$

so the total time in the queue of the 1^{st} relay is

$$\begin{aligned} Q_1 + H_1 &= N\frac{p_I}{p_O}\frac{1-p_O}{p_O-p_I} + \frac{N}{p_O} - N + 1 \\ &= N\frac{1-p_I}{p_O-p_I} - N + 1. \end{aligned} \tag{4.6}$$

If the queue at the 1^{st} relay is stable, i.e., $pp_1 < p_R p_2$, the packet arrival probability to the 2^{nd} relay is the probability that the former is not empty, multiplied by $p_R p_2$, i.e.

$$\frac{pp_1}{p_R p_2} \cdot p_R p_2 = pp_1.$$

It is clear that, as long as each queue is stable, the packet arrival probability to all relay queues is pp_1. From (4.3) and (4.6), D is computed as

$$D = \frac{N}{pp_1} + N\sum_{n=2}^{N}\frac{1-pp_1}{p_R p_n - pp_1} - N(N-1). \tag{4.7}$$

If the sources are not backlogged, then, provided that all queues are stable (including the source queue), the packet arrival probability to all queues is p_A [15]. The delay is

$$D = N\frac{1-p_A}{pp_1-p_A} + N\sum_{n=2}^{N}\frac{1-p_A}{p_R p_n - p_A} - N(N-1). \tag{4.8}$$

The RT can be computed as follows: every N slots, a packet is received by the final destination with probability pp_1 (p_A for non-backlogged sources). Hence,

$$\text{RT} = \begin{cases} \frac{pp_1}{N}, & p_A = 1 \\ \frac{p_A}{N}, & p_A < 1 \end{cases}. \tag{4.9}$$

4.3.2 Evaluation of packet success probabilities

Assume that the network operation starts at some arbitrary time, with an arbitrary number of packets in each queue. Since transmissions interfere with each other, the evolutions of different queues are coupled and a study of the network dynamics appears intractable. Similarly to [16], the symmetry and randomness present in our model allow us to study the network in a stationary regime, where a packet success at the n^{th} hop of the typical route occurs with probability p_n, which depends on the average interference environment, yet it is independent of time and location.

In order to ensure that the length of any queue remains bounded in time, we take a worst-case interference scenario, where all nodes in the corresponding slot are assumed to have a packet to transmit, and *demand* that the packet arrival probability is smaller than the resulting lower bound of the packet departure probability. This is a sufficient condition for the underlying Markov chain describing the queue occupancy to be positive recurrent [17]. Under this condition, we derive fixed-point equations that provide solutions for the packet success probabilities.

The following lemmas will be useful in the subsequent analysis and are stated without proof.

Lemma 1. *The function $f(x) = x\mathrm{e}^x,\ x \in \mathbb{R}$, is continuous, with a unique minimum $f_0 = -\mathrm{e}^{-1}$ at $x_o = -1$.*

Lemma 2. *The equation $y = x\mathrm{e}^x,\ y \in [-\mathrm{e}^{-1}, 0]$, has two solutions, $x_2 \leq -1 \leq x_1 \leq 0$. The largest of the two is given by $x_1 = W(y)$, where $W(y),\ y \geq -\mathrm{e}^{-1}$, is known as the Lambert function.*

We consider the backlogged and non-backlogged source cases separately.

Backlogged sources

As a starting point, assume that $N = 2$ and $p_R = 1$. Recalling (4.1), the success probability in the 1^{st} hop is given by

$$p_1 = \mathrm{e}^{-\lambda_0 c r_1^2} = \mathrm{e}^{-\lambda p c r_1^2},\ 0 \leq r_1 \leq R. \tag{4.10}$$

As a worst-case interference scenario, consider that, in odd slots, all the relays have non-empty queues. A lower bound to the success probability on the 2^{nd} hop is therefore $p_2' = \mathrm{e}^{-\lambda c (R - r_1)^2}$. A sufficient condition for the stability of the relay queues is

$$\begin{aligned} p p_1 &< p_2' \\ p\mathrm{e}^{-\lambda p c r_1^2} &< \mathrm{e}^{-\lambda c (R-r_1)^2}. \end{aligned} \tag{4.11}$$

In the special case where $r_1 = 0$, i.e., the relay is placed arbitrarily close to the source, (4.11) reduces to $p < \mathrm{e}^{-\lambda c R^2}$. Also, if $p = 1$, it is seen that (4.11) holds, if and only if $r_1 > R/2$, i.e., the relay is placed closer to the destination than the source. For $r_1 > 0$ and general values of p, we have the following proposition.

Proposition 1. *Let $r_o \in [R/2, R]$ be such that*

$$\lambda c r_o^2 \mathrm{e}^{-\lambda c (R-r_o)^2} = \mathrm{e}^{-1},$$

with $r_o \triangleq R$, if there is no solution to the equation. Then, for any $r_1 \in (0, r_o]$, (4.11) is satisfied if

$$p < -\frac{W(-\lambda c r_1^2 \mathrm{e}^{-\lambda c (R-r_1)^2})}{\lambda c r_1^2} \triangleq p^{(1)}(r_1), \tag{4.12}$$

while, for any $r_1 \in (r_o, R]$, it is satisfied for all p.

Proof. Multiplying both sides of (4.11) by $-\lambda c r_1^2$, we have

$$\begin{aligned} -\lambda c p r_1^2 \mathrm{e}^{-\lambda p c r_1^2} &> -\lambda c r_1^2 \mathrm{e}^{-\lambda c (R-r_1)^2} \\ f(-\lambda c p r_1^2) &> -g(r_1), \end{aligned} \tag{4.13}$$

where $g(r_1) \triangleq \lambda c r_1^2 \mathrm{e}^{-\lambda c (R-r_1)^2}, r_1 \in [0, R]$, is a continuous and strictly increasing function of r_1 with a maximum value $g(R) = \lambda c R^2$. If $g(R) \geq \mathrm{e}^{-1}$ or, equivalently, $R \geq 1/\sqrt{\lambda c \mathrm{e}}$, then $g(r_1)$ takes the value e^{-1} for some $r_1 = r_o$, i.e., $g(r_o) = \mathrm{e}^{-1}$. The distance r_o is always greater than or equal to $R/2$, a fact that we can prove with the help of Lemma 1:

$$g(R/2) = \lambda c (R/2)^2 \mathrm{e}^{-\lambda c (R/2)^2} \leq \mathrm{e}^{-1} = g(r_o),$$

or $r_o \geq R/2$, with the equality occurring when $R = 2/\sqrt{\lambda c}$. On the other hand, if $R < 1/\sqrt{\lambda c \mathrm{e}}$, then, $g(r_1) < \mathrm{e}^{-1}$ for all $r_1 \in [0, R]$, in which case we define $r_o = R$.

According to Lemma 1, the minimum value of $f(-\lambda c p r_1^2)$ is $f_0 = -\mathrm{e}^{-1}$. If $r_1 > r_o$, then $-g(r_1) < -\mathrm{e}^{-1} = f_0$, so (4.13) holds for any p. On the other hand, if $r_1 \leq r_o$ or $-g(r_1) \geq -\mathrm{e}^{-1}$, then, according to Lemma 2, the equation $f(-\lambda c p r_1^2) = -g(r_1)$ has two solutions $x_1 = -\lambda c p^{(1)} r_1^2$ and $x_2 = -\lambda c p^{(2)} r_1^2$, of which the former is obtained by the Lambert function, [2] which, when applied to both sides of (4.13), yields

$$\begin{aligned} -\lambda c p r_1^2 &> W(-g(r_1)) \\ p &< -\frac{W(-\lambda c r_1^2 \mathrm{e}^{-\lambda c (R-r_1)^2})}{\lambda c r_1^2} = p^{(1)}(r_1). \end{aligned}$$

□

Note that (4.13) is satisfied for all $p \in [0, p^{(1)}) \cup (p^{(2)}, +\infty)$. However, from a practical standpoint, such as energy consumption, the subset of larger values of p can be discarded.

For a given relay position, Proposition 1 provides a limit on p, such that the relay queue is guaranteed to be stable. However, it does not take into account the fact that p is a probability, i.e. $p \in [0, 1]$. The following lemma addresses the question whether $p^{(1)}(r_1)$ in (4.12) is greater than unity.

Lemma 3. *The function $p^{(1)}(r_1)$ is strictly increasing in r_1. If $R \in \left(0, \frac{2}{\sqrt{\lambda c}}\right]$, then $p^{(1)}(R/2) = 1$. If $R > \frac{2}{\sqrt{\lambda c}}$, then $p^{(1)}(r_1) < 1$, for $r_1 \in (0, r_o]$.*

[2] Note that the two solutions coincide at $r_1 = r_o$.

Proof. Using the property $W(y)\mathrm{e}^{W(y)} = y$, $p^{(1)}(r_1)$ can be rewritten as

$$p^{(1)}(r_1) = \mathrm{e}^{-\lambda c(R-r_1)^2 - W(-\lambda c r_1^2 \mathrm{e}^{-\lambda c(R-r_1)^2})},$$

which is strictly increasing in r_1, since $W(y)$ is strictly increasing in y. Setting $p^{(1)}(r_1)$ equal to unity, we have

$$\begin{aligned} W(-\lambda c r_1^2 \mathrm{e}^{-\lambda c(R-r_1)^2}) &= -\lambda c r_1^2 \\ -\lambda c r_1^2 \mathrm{e}^{-\lambda c(R-r_1)^2} &= -\lambda c r_1^2 \mathrm{e}^{-\lambda c r_1^2} \\ r_1 &= \frac{R}{2}. \end{aligned}$$

By the definition of the Lambert function, the first equality can only be valid when $\lambda c r_1^2 \leq 1$. Since the equality occurs at $r_1 = R/2$, it can only hold when $R \leq 2/\sqrt{\lambda c}$.

On the other hand, if $R > 2/\sqrt{\lambda c}$, note that

$$g\left(\frac{1}{\sqrt{\lambda c}}\right) = \mathrm{e}^{-\lambda c\left(R-\frac{1}{\lambda c}\right)^2} < \mathrm{e}^{-\lambda c\left(\frac{1}{\sqrt{\lambda c}}\right)^2} = \mathrm{e}^{-1} = g(r_o),$$

or, equivalently, $r_o > 1/\sqrt{\lambda c}$. Since the maximum value of $p^{(1)}(r_1)$ is $p^{(1)}(r_o) = 1/(\lambda c r_o)^2$, it follows that $p^{(1)}(r_1) < 1$, for $r_1 \in (0, r_o]$.

□

Lemma 3 determines the range of R for which, placing the relay closer to the destination than to the source is sufficient for the stability of the relay queues, independently of the value of p.

If (4.11) is satisfied, we are guaranteed that pp_1 is smaller than the *actual* packet success probability in the 2^{nd} hop, p_2. According to the queueing analysis in Section 4.3.1, the probability that the typical relay queue is not empty at the end of an even slot is then pp_1/p_2. The density of active relays at the 2^{nd} hop is therefore $\lambda_1 = \lambda pp_1/p_2$. From (4.1), we obtain

$$p_2 = \mathrm{e}^{-\lambda \frac{pp_1}{p_2} c(R-r_1)^2}. \tag{4.14}$$

This fixed-point equation with respect to p_2 can also be written in the form

$$f\left(-\frac{\lambda pp_1 c(R-r_1)^2}{p_2}\right) = -\lambda pp_1 c(R-r_1)^2. \tag{4.15}$$

Lemma 1 dictates that (4.15) has two solutions, as long as $\lambda pp_1 c(R-r_1)^2 \leq \mathrm{e}^{-1}$. This is obviously satisfied when $r_1 = R$. For $r_1 < R$, multiplying both sides by $\lambda c(R-r_1)^2$, it is easy to see that (4.11) becomes

$$\lambda pp_1 c(R-r_1)^2 < \lambda c(R-r_1)^2 \mathrm{e}^{-\lambda c(R-r_1)^2} \leq \mathrm{e}^{-1}.$$

Once again, applying the Lambert function to (4.15) gives the desirable solution for p_2 [3]

[3] It can be verified that the other solution for p_2 is decreasing with r_1, and therefore is ignored.

$$\begin{aligned} p_2 &= \frac{-\lambda p p_1 c (R-r_1)^2}{W(-\lambda p p_1 c (R-r_1)^2)} \\ &= \mathrm{e}^{W(-\lambda p p_1 c (R-r_1)^2)}, \end{aligned} \tag{4.16}$$

where we have once more used the property $W(y)\mathrm{e}^{W(y)} = y$, to derive the second equality.

If $p_R < 1$, (4.11) is modified as

$$p\mathrm{e}^{-\lambda p c r_1^2} < p_R \mathrm{e}^{-\lambda p_R c (R-r_1)^2}. \tag{4.17}$$

Note that, if (4.17) is satisfied, the packet success probability in the 2^{nd} hop is still given by (4.14). This is because the probability that a relay is active, $pp_1/(p_R p_2) \cdot p_R = pp_1/p_2$ - and thus the interfering relay density - is unchanged. The effect of $p_R < 1$ on the delay performance is not apparent; p_R appears in the denominator of (4.7), which is potentially detrimental. However, as seen from (4.17), $p_R < 1$ may also result in a larger range of acceptable values for p, over which the delay can be minimized. The effect of $p_R < 1$ on the delay performance is examined via a numerical example in the next section.

When $N > 2$, sufficient conditions for the stability of all relay queues are obtained by demanding that pp_1 is smaller than the worst-case success probability at the n^{th} hop, i.e.,

$$p\mathrm{e}^{-\lambda p c r_1^2} < \mathrm{e}^{-\lambda c (r_n - r_{n-1})^2},\ n = 2, \ldots, N. \tag{4.18}$$

If these conditions are satisfied, the active relay density at the n^{th} hop is $\lambda_{n-1} = \lambda p p_1 / p_n$. This, together with (4.1), yields

$$p_n = \mathrm{e}^{-\lambda \frac{p p_1}{p_n} c (r_n - r_{n-1})^2}. \tag{4.19}$$

Similarly to (4.16), the solution to the above equation is

$$\begin{aligned} p_n &= \frac{-\lambda p p_1 c (r_n - r_{n-1})^2}{W(-\lambda p p_1 c (r_n - r_{n-1})^2)} \\ &= \mathrm{e}^{W(-\lambda p p_1 c (r_n - r_{n-1})^2)}. \end{aligned} \tag{4.20}$$

Since $\{p_n\}$ are decreasing functions of the respective hop lengths $\{r_n - r_{n-1}\}$, the question is raised if there is an optimal relay placement, that minimizes D. The following lemma addresses this issue.

Lemma 4. *The mean end-to-end delay in (4.7) is minimized when hops $2, 3, \ldots, N$ are equidistant.*

Proof. Suppose r_1, therefore p_1, are fixed. Defining $h(r_n - r_{n-1}) = (p_n - pp_1)^{-1}$, D is minimized over r_n if $h'(r_n - r_{n-1}) = h'(r_{n+1} - r_n)$. Since $h'(\cdot)$ is strictly increasing in its argument (the proof of which we omit), it follows that $r_n - r_{n-1} = r_{n+1} - r_n$, for $n = 2, \ldots, N$. □

Non-backlogged sources ($p_A < 1$)

Assume that $p = p_R = 1$ and $N = 2$. Assuming again a worst-case interference scenario, where all sources and relays are active, we have

$$\begin{aligned} p_1' &= \mathrm{e}^{-\lambda c r_1^2} \\ p_2' &= \mathrm{e}^{-\lambda c (R-r_1)^2} \end{aligned}$$

Given a relay placement $r_1 \in [0, R]$, a sufficient condition for stability is

$$p_A < \mathrm{e}^{-\lambda c \max\{r_1, R-r_1\}^2},$$

from which it is seen that p_A cannot take values larger than $\mathrm{e}^{-\lambda c (R/2)^2}$. Conversely, for a given p_A, the acceptable range of relay positions is

$$R - \sqrt{-\frac{\log p_A}{\lambda c}} < r_1 < \sqrt{-\frac{\log p_A}{\lambda c}},$$

Following the steps of the backlogged source case, the success probabilities in the 1^{st} and 2^{nd} hops are given by

$$\begin{aligned} p_1 &= \frac{-\lambda p_A c r_1^2}{W(-\lambda p_A c r_1^2)} \\ p_2 &= \frac{-\lambda p_A c (R-r_1)^2}{W(-\lambda p_A c (R-r_1)^2)}. \end{aligned} \tag{4.21}$$

The extension of the above results to $N > 2$ is straightforward. Using the same approach as in Lemma 4, we can show that D in (4.8) is minimized when *all* hops are equidistant. Finally, similar observations to the backlogged source case hold for the effect of $p, p_A < 1$ on the delay performance.

4.3.3 Examples

Unless otherwise stated, $N = 2$, $p = p_R = 1$, $\lambda = 4 \times 10^{-4}$ nodes/m^2, $\theta = 6$ dB and $b = 4$. Note that any choice of parameters that yields the same value for $\lambda \pi R^2$ (the average number of sources in a circle centered at the typical source with radius R) leads to the same numerical results.

We first explore the impact of the number of hops on the performance, for backlogged sources. In Figs. 4.3 and 4.4, the minimum delay - over all possible relay positions - and the respective NT are plotted vs. R, for different numbers of hops. We note that there is an optimal number of hops for each R which minimizes delay and maximizes NT simultaneously. Moreover, reproducing these plots for different values of the product λc reveals that the distance between switching points (values of R for which N and $N+1$ hops yield the same delay) is proportional to $1/\sqrt{\lambda c}$; this agrees with the intuition that increasing the network density or making the SIR constraint tighter requires more hops in order to minimize the delay, for a given R. Another observation is that the lower envelope of the delay curves is approximately linear. A similar conclusion is reached in [18], regarding the time it takes for a path to form between the source and the destination in a dynamic connectivity network.

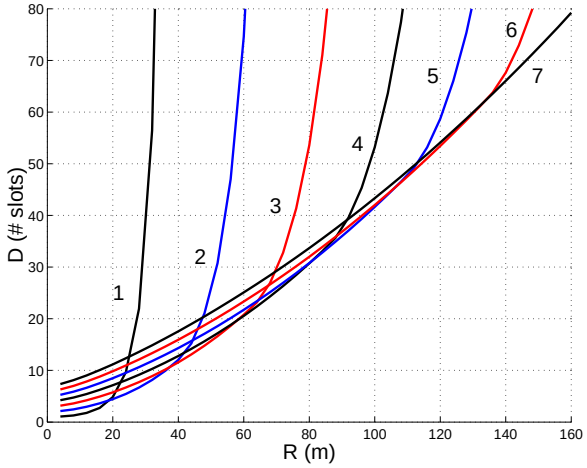

Figure 4.3. *D vs. R, for $N = 1, 2, \ldots, 7$ hops and backlogged sources ($p = p_R = 1$). For each R, D is minimized over all relay placements.*

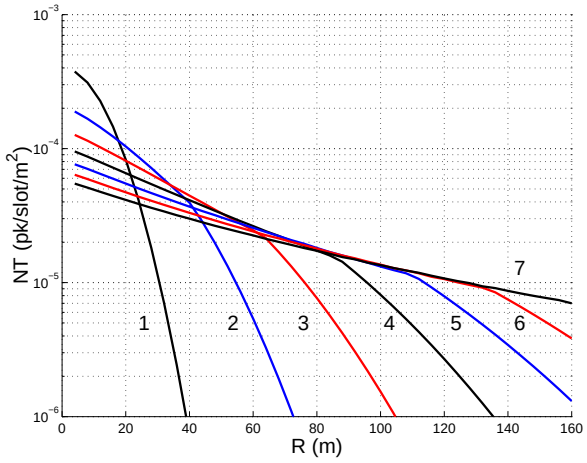

Figure 4.4. *NT vs. R, for $N = 1, 2, \ldots, 7$ hops and backlogged sources ($p = p_R = 1$).*

Enforcing a MAP p at the source and p_R at the relay introduces two more parameters over which the delay can be optimized - see (4.7). In Fig. 4.5, the delay is minimized over various subsets of $\{r_1, p, p_R\}$ and plotted vs. R, for a two-hop network. Note that, jointly optimizing over p and r_1, than simply over r_1, yields a benefit for $R \simeq 35$m. This is also roughly the distance at which three-hop routing results in smaller delay than two-hop routing when $p = 1$ (Fig. 4.3). Moreover, the effect of optimizing over p_R does not become significant except for larger distances. The delay for a single-hop network is also shown for comparison; it is seen that optimizing over p reduces the performance gap between single-hop and two-hop routing significantly. Alternatively, while the single-hop strategy is simple from a routing perspective, its delay performance has a high degree of sensitivity to the choice of p.

Figs. 4.6 and 4.7 demonstrate the sensitivity of a two-hop backlogged network to imperfect choices of r_1 or p. In Fig. 4.6, the delay is plotted vs. r_1, for $p = 1$. As expected, there exists a relay placement ($r_1 \simeq 13$ m), for which the success probabilities p_1 and p_2 can be optimally adjusted

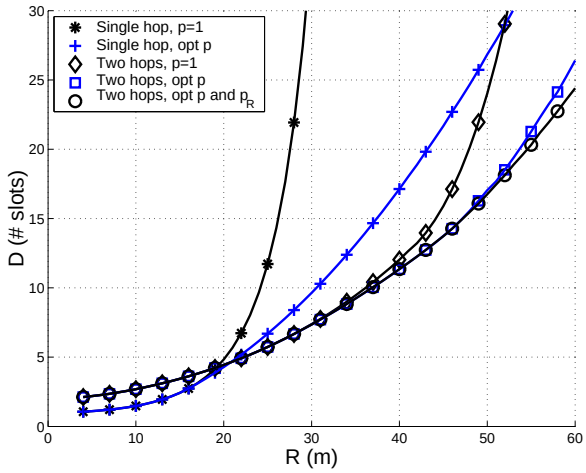

Figure 4.5. *D vs. R for optimized/non-optimized MAP and backlogged sources.*

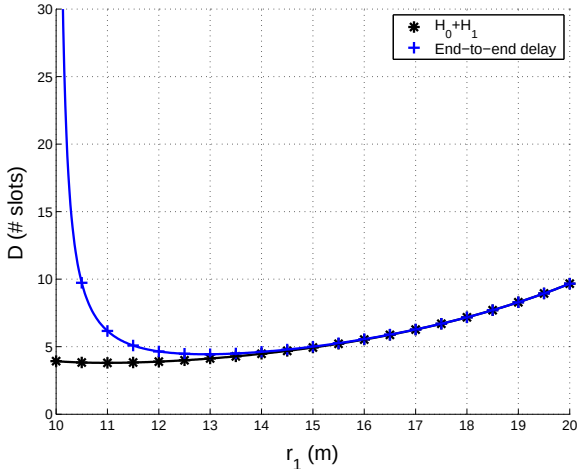

Figure 4.6. *D vs. r_1 for backlogged sources ($N = 2$, $p = p_R = 1$, $R = 20$ m).*

to minimize (4.7). Below this critical value, as $r_1 \to R/2$, the difference $p_2 - p_1$ tends to zero and the queueing delay becomes infinite. On the other hand, as $r_1 \to R$, then $p_2 \to 1$ and the service time at the source H_0 dominates the delay performance. Fig. 4.6 indicates that, should it not be possible to have a relay at the optimal position, it is preferable to select one that lies closer to the destination than to the source, as the delay penalty is far smaller in the first case.

In Fig. 4.7, we set $R = 30$ m and plot the delay vs. p for different relay positions. The minimum delay is achieved for $p = 1$ and $r_1 = 17$ m. Placing the relay below the half-point at $r_1 = 9R/20$, imposes a limit on the acceptable values of p, given by Proposition 1; if p approaches this limit, the delay can quickly become worse than that of a single-hop system. When $r_1 = 2R/3 > R/2$, similarly to Fig. 4.6, the delay is quite insensitive to the value of p.

The non-backlogged source scenario is examined in Figs. 4.8 and 4.9. Fig. 4.8 presents the dependence of the delay on r_1, for different arrival probabilities p_A. The minimum delay, irrespective of p_A, is achieved at $r_1 = R/2$, as predicted in Section 4.3.2. It is also intuitive that, as the traffic load

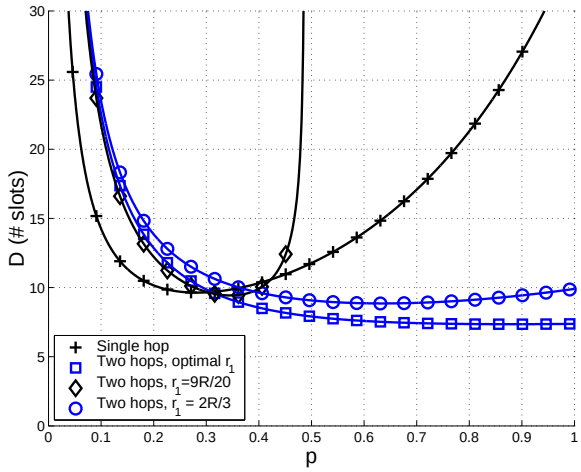

Figure 4.7. *D vs. p, for different relay placements and backlogged sources ($p_R = 1$, $R = 30$ m).*

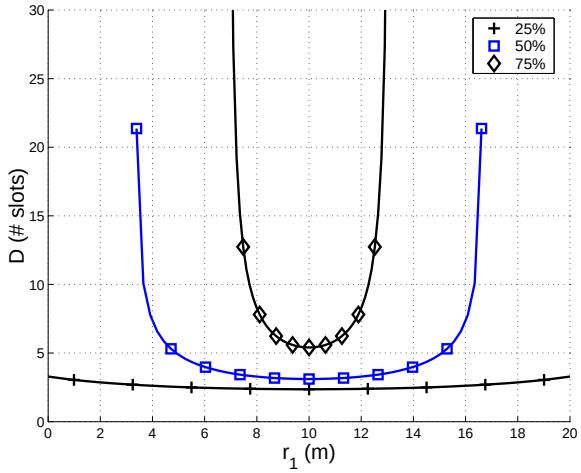

Figure 4.8. *D vs. r_1, for different percentages of the maximum allowable p_A ($N = 2$, $p = p_R = 1$, $R = 20$ m).*

increases, the delay becomes more sensitive to relay positions away from the half point. In Fig. 4.9, we obtain the number of hops that minimizes the delay, as a function of R. The curve marked as 100% shows the number of hops that maximizes the minimum RT value that renders the queues unstable. Each of the other curves shows the optimal hop count for a given RT requirement, as a percentage of this maximizing value. We observe that, for a given R, a smaller number of hops is required to minimize the delay as the throughput requirement is relaxed.

4.4 Simulation Results

A network simulator was constructed in MATLAB, in order to verify our analytical results. A number of routes were scattered uniformly over a 1800×1800 m^2 square. In order to minimize the impact of edge effects, metrics were collected only for the routes whose source was situated inside an inner square of side 1000m. In every slot, the positions of the sources in the inner square and the

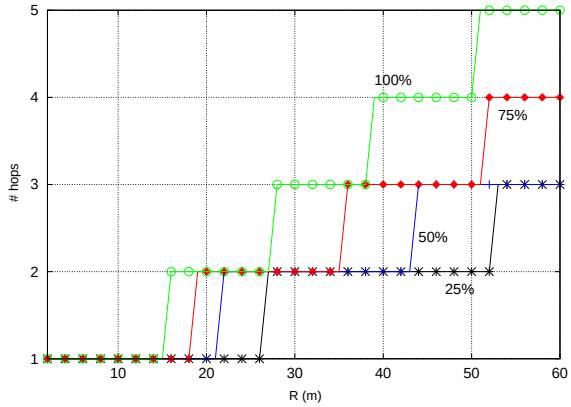

Figure 4.9. *Optimum number of hops vs. R, for different percentages of the minimum unstable RT ($N = 2$, $p = p_R = 1$).*

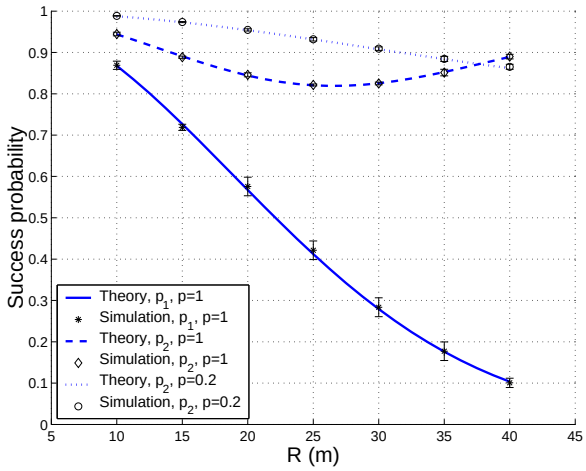

Figure 4.10. *Comparison of theoretical and simulated packet success probabilities for a two-hop network and backlogged sources ($p_R = 1,\ r_1 = 0.6R$).*

outer square annulus, as well as the orientation of their destinations, were randomly generated. This mimicked the assumption of a new network realization in every slot, while allowing us to keep track of the queue evolutions of the inner routes and compute the stationary packet success probabilities. The simulator was run for a sufficient number of slots to guarantee adequate statistical confidence (99% confidence intervals are shown) and the main parameter values were $\lambda = 4 \times 10^{-4},\ \theta = 6\,\mathrm{dB},\ b = 4$.

Figs. 4.10 and 4.11 show the packet success probabilities for a two-hop and a three-hop network, respectively, as functions of R. The sources are backlogged and the relay distances from the source are fixed fractions of R. It can be seen that the simulation results and the theoretical ones, given by (4.16) and (4.20), are in good agreement. Fig. 4.11 also implicitly verifies the fact that, under stable operation, the arrival rate to the second relay is p_1.

Fig. 4.12 shows simulation results for the packet success probabilities in a two-hop network with non-backlogged sources when r_1 and p ($= p_R$) are varied, to confirm the theoretical results given by (4.21). Note that, as stated in Section 4.3.2, the simulated p_1 and p_2 do not depend on the value of p,

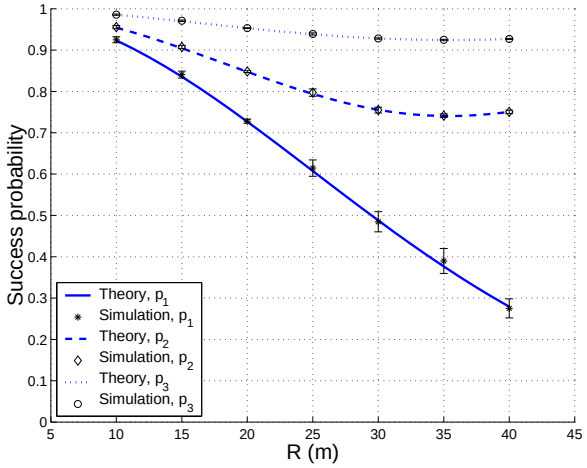

Figure 4.11. *Comparison of theoretical and simulated packet success probabilities for a three-hop network and backlogged sources ($p = p_R = 1,\ r_1 = 0.45R,\ r_2 = 0.8R$).*

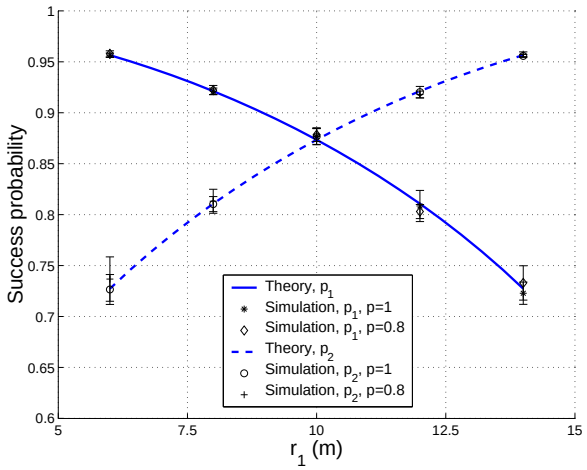

Figure 4.12. *Comparison of theoretical and simulated success probabilities for a two-hop network and non-backlogged sources ($p_A = 0.3,\ R = 20$ m).*

nor p_R, under stable operation.

Some more comments are in order regarding our analytical framework. In Section 4.2, we assumed that the network is completely rearranged at the beginning of every slot. This is a strong assumption in terms of node mobility, as, in most practical networks, nodes remain rather stationary over several packet slots. However, even in a static or slow-mobility network, enforcing a MAP on the sources and relays can emulate node mobility at the slot level, since, effectively, different subsets of nodes are active in each slot. In Fig. 4.13, we explore the validity of this claim by plotting the average success probabilities in the 1^{st} and 2^{nd} hops over many *static* network topologies, for different values of $p = p_R$. For small p, the degree of randomization is such that the simulation results agree with p_1 and p_2 given by (4.10) and (4.16). As p increases, the interference levels in the 1^{st} and 2^{nd} hops become more correlated, so, if a packet transmission is successful in the 1^{st} hop, it is likely that that will be the case in the 2^{nd} hop. As a result, (4.16) provides a pessimistic estimate of the actual

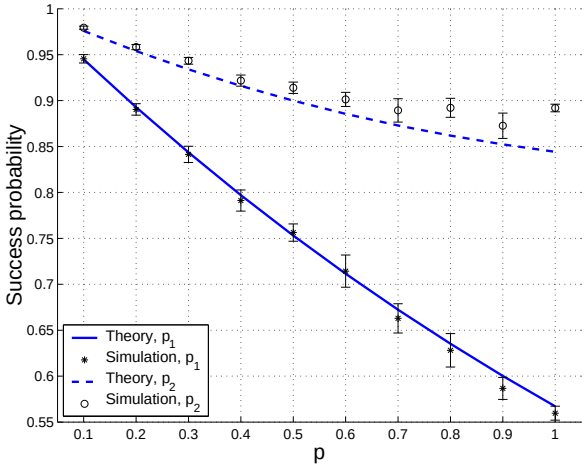

Figure 4.13. *Simulated packet success probabilities for a static two-hop network and backlogged sources ($R = 20$m, $r_1 = 12$m). The theoretical values correspond to the system model described in Section 4.2.*

success probability in the 2^{nd} hop.

Finally, the stability conditions in Section 4.3.2 are sufficient but not necessary; consequently, the delay given by (4.7) or (4.8) is not necessarily infinite when these conditions are violated. It can be verified, e.g., that in the two-hop, backlogged, $p = 1$ scenario, there exists an $\epsilon > 0$, such that p_2, given by (4.16), is greater than p_1 at $r_1 = R/2 - \epsilon$, provided that $R > 2/\sqrt{\lambda c}$. We simulated this scenario and observed the time-traces of a set of randomly picked relay queues. It appears that the queue evolutions depend on the initial conditions, e.g., if a large enough fraction of nodes have non-empty queues at the beginning of the simulation, the interference level in the network is such that transmissions in the second hop are mostly unsuccessful. This creates a positive feedback effect, i.e., more interference, and, after some time, the size of all the observed queues grows with time. In this regime, all the relays are backlogged and $p_2 = \mathrm{e}^{-\lambda c(R-r_1)^2} < p_1$. Note that these simulation results serve to illustrate some trends in the network behavior and are in no way conclusive in terms of stability, when the conditions in Section 4.3.2 are not satisfied. In the absence of a framework to characterize the network dynamics, taking a worst-case interference scenario is a conservative approach, that nevertheless guarantees network stability.

4.5 Chapter Conclusions

We have conducted a study of random interference-limited multi-hop networks, using tools from basic queueing theory and stochastic geometry. The mean end-to-end delay and throughput in a typical route were evaluated as functions of physical and MAC layer parameters. In summary:

- The number of relays and their placements were determined such that the delay is minimized, for a given distance to the final destination. Interestingly, the number of hops that minimizes the delay also maximizes the throughput.

- The MAP that must be enforced on backlogged sources, such that the delay remains finite,

irrespective of the relay placement, was characterized. It was shown that the closer the relay is placed to the source in a two-hop network, the smaller the acceptable range of values that the MAP can take, such that the delay is bounded.

- The delay can be very sensitive to the relay placement depending, e.g., on the traffic load. This observation has implications in terms of routing protocols, i.e., selecting a relay from a random node population in a practical network.

The focus of this Chapter has been to develop a tractable analytical network model and verifying its results via simulation. Extending the model in order to accomodate more practical scenarios, e.g., asynchronous flows, random final destination distances and random relay populations, is an important area of current research by the authors. Network simulations that accomodate these scenarios are also being set up. Our preliminary results seem to indicate that the design insights obtained by our analysis are valid in more general settings.

4.6 Acknowledgments

This work is the result of a collaboration with Kostas Stamatiou (UCSD), Martin Haenggi (University of Notre Dame), Tara Javidi (UCSD) and James Zeidler (UCSD).

References

[1] M. Haenggi and D. Puccinelli, "Routing in ad hoc networks: a case for long hops," *IEEE Commun. Mag.*, vol. 4, p. 93101, Oct. 2005.

[2] J. Andrews et al., "Rethinking information theory for mobile ad hoc networks," *IEEE Commun. Mag.*, 2008, submitted to the IEEE Transactions on Information Theory.

[3] F. Xue and P. R. Kumar, "Scaling laws for ad hoc wireless networks: an information theoretic approach," *Series of Foundations and Trends in Networking, Now Publishers Inc.*, vol. 1, pp. 145–270, Jul. 2006.

[4] E. S. Sousa and J. A. Silvester, "Optimum transmission ranges in a direct-sequence spread-spectrum multihop packet radio network," *IEEE J. Sel. Areas Commun.*, vol. 8, pp. 762–771, Jun. 1990.

[5] A. Armaroli and M. Zorzi, "Advancement optimization in multihop wireless networks," in *IEEE VTC Fall 03*, Orlando (FL, USA), Oct. 2003.

[6] S. P. Weber, X. Yang, J. G. Andrews, and G. de Veciana, "Transmission capacity of wireless ad hoc networks with outage constraints," *IEEE Trans. Inf. Theory*, vol. 51, pp. 4091–4102, Dec. 2005.

[7] F. Baccelli, B. Błaszczyszyn, and P. Mühlethaler, "An Aloha protocol for multi-hop mobile wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, pp. 421–436, Feb. 2006.

[8] S. P. Weber, J. G. Andrews, and N. Jindal, "The effect of fading, channel inversion and threshold scheduling on ad hoc networks," *IEEE Trans. Inf. Theory*, vol. 53, p. 41274149, Nov. 2007.

[9] K. Stamatiou and J. G. Proakis, "Assessing the impact of physical layer techniques on ad hoc network performance," *Elsevier Physical Communication*, vol. 1, p. 8491, 2008.

[10] M. Haenggi, "On routing in random rayleigh fading networks," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 1553–1562, Jul. 2005.

[11] O. Oyman and S. Sandhu, "A Shannon theoretic perspective on fading multihop networks," in *CISS 06*, Princeton (NJ, USA), Mar. 2006.

[12] M. Sikora, J. N. Laneman, M. Haenggi, D. J. Costello, and T. Fuja, "Bandwidth- and power-efficient routing in linear wireless networks," *Joint Special Issue of IEEE Transactions on Information Theory and IEEE Transactions on Networking*, vol. 52, pp. 2624–2633, Jun. 2006.

[13] D. Rajan, "Power efficient delay allocation in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 56, pp. 1813–1825, Jul. 2007.

[14] S. Karlin and H. M. Taylor, *An introduction to stochastic modeling*, 3rd ed. Boston (MA, USA): Academic press, 1994.

[15] J. Hsu and P. P. Burke, "Behavior of tandem buffers with geometric input and markovian output," *IEEE Trans. Commun.*, vol. 24, no. 3, pp. 358 – 361, Mar. 1976.

[16] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed-point analysis of single-cell IEEE 802.11 WLANs," *IEEE/ACM Trans. Netw.*, vol. 15, pp. 588–601, Jun. 2007.

[17] S. Asmussen, *Applied probability and queues*, 2nd ed. Springer-Verlag, 2003.

[18] R. Ganti and M. Haenggi, "Dynamic connectivity and packet propagation delay in aloha wireless networks," in *Asilomar Conference on Signals, Systems and Computers*, Asilomar (CA, USA), Nov. 2007, pp. 143 – 147.

Chapter 5

Conclusions

The main theme of this thesis has been how the interactions between physical and MAC/routing layer affect protocol design. Such a question has been investigated in a variety of contexts, and the foremost lesson has been that physical layer and MAC/routing protocols should not be disjointly designed. This does not necessarily mean that there must be couplings in terms of cross-layer design, but the choice of the physical layer greatly affects the MAC and viceversa.

In MIMO networks, it has been shown that suitable space time coding can provide range extension comparable to beamforming but at a fraction of the delay, an essential matter for broadcast packets. Moreover, MIMO signal processing is extremely useful to make Network Coding (a layer 2/3 technique) more efficient in the wireless environment and generate very powerful cooperative protocols, which reward nodes for their cooperative behavior.

As far as carrier sense is concerned, a special busy channel detection mechanism must be devised for MIMO in order to exploit its potential spatial reuse. The results have shown that realistic MIMO carrier sense mechanism can reap most of the available spatial degrees of freedom. On the other hand, for single antenna networks we have designed algorithms and analytical models to compute the optimal carrier sense threshold in a variety of environments. Finally, these models have led to the creation of a low complexity, high performance scheduler for mesh networks.

The third chapter has shown how physical layer modelling is not at crossroads with protocol modelling and it can also yield important design insights. Chapter 4 has presented a framework to analyze a wide class of wireless networks of different networks, and it explicitly computes the end-to-end performance of the network. The dependence of the results on many physical layer parameters is made explicit and simple yet insightful relationships between PHY, MAC and routing metrics have been observed.

Appendix A

Complete List of Papers

This Appendix presents a complete list of papers published, accepted or submitted during the Ph.D. program. For convenience, the papers are grouped according to their main topic.

A.1 Papers on MIMO signal processing

The work in [1] analyses how specially crafted space time coding can bridge the gap with beamforming in terms of BER for broadcast traffic. The second part in [2] focuses on network performance, while the journal version [3] performs a synthesis of the two and adds new results.

On the other hand, MIMO_NC has been first introduced in [4], together with the results on the diversity order for NC and MIMO_NC. Super MIMO_NC is proposed as a way to get around this problem in [5] and a study of the sensitivity of these methods to channel estimation errors is reported in [6]. A more mature contribution is [7], which joins [4] and [5], and proposes extensions and new results.

Phoenix has been defined in [8], together with the analytical model. On the other hand, [9] has carried out extensive performance evaluation for clustered networks and [10].

Finally, a review of hybrid cooperative-network coding protocols has been carried out in [11].

A.2 Papers on carrier sense

The study of MIMO carrier sense has been the focus of the collaboration with the master student Emanuele Coviello, and the gist of it is presented in [12].

On the other hand, work on static and dynamic network optimization is carried out in [13] and [14], respectively. Finally, the mesh scheduler based on the model in [13] is exposed in [15].

A.3 Papers on stochastic geometry and other topics

The submitted paper [16] deals with stochastic geomertry and its uses for wireless networks analysis. The papers [17, 18] stem from Andrea Munari's Master Thesis on directional antennas. Finally, the article [19] is the product of an internship in Ericsson on cellular networks.

Published or submitted papers

[1] F. Rossetto and M. Zorzi, "On gain asymmetry and broadcast efficiency in MIMO ad hoc networks," in *IEEE ICC*, Istanbul (Turkey), Jun. 2006.

[2] ——, "A Space Time based Approach to Solving the Gain Asymmetry in MIMO ad hoc Networks," in *IEEE VTC Spring*, Melbourne (Australia), May 2006.

[3] F. Rossetto and M. Zorzi., "A Low Delay MAC solution for MIMO ad hoc networks," *IEEE Trans. Wireless Commun.*, (accepted).

[4] E. Fasolo, F. Rossetto, and M. Zorzi, "Network Coding meets MIMO," in *NetCod 2008*, Hong Kong (China), Jan. 3-4 2008.

[5] ——, "On encoding and rate adaptation for MIMO_NC," in *IEEE ISCCSP*, Malta, Mar. 12-14 2008.

[6] F. Rossetto and M. Zorzi, "On the sensitivity of MIMO NC to channel estimation errors," in *EUCAP*, (submitted).

[7] E. Fasolo, F. Rossetto, and M. Zorzi, "Network Coding meets MIMO: A look into the rate/diversity tradeoff in wireless Network Coding," *IEEE Trans. Commun.*, (submitted).

[8] E. Fasolo, A. Munari, F. Rossetto, and M. Zorzi, "Phoenix: A hybrid cooperative-network coding protocol for fast failure recovery in ad hoc networks," in *IEEE SECON 2008*, San Francisco (CA, USA), 16-20 June 2008.

[9] A. Munari, F. Rossetto, and M. Zorzi, "On the viability of a Cooperative-Network Coding Protocol in Clustered Networks," in *IEEE MILCOM*, San Diego (CA, USA), Nov. 2008 (accepted).

[10] ——, "Phoenix: Making Cooperation more Efficient through Network Coding in Wireless Networks," *IEEE Trans. Wireless Commun.*, (submitted).

[11] F. Rossetto and M. Zorzi., "A new frontier of cross layer design: cooperative Network Coding," *IEEE Wireless Commun. Mag.*, (submitted).

[12] E. Coviello, A. Bhorkar, F. Rossetto, B. D. Rao, and M. Zorzi, "A Robust approach to Carrier Sense for MIMO ad hoc networks," in *IEEE ICC*, Dresden (Germany), Jun. 2009 (submitted).

[13] F. Rossetto and M. Zorzi, "Gaussian approximations for carrier sense modeling in wireless ad hoc networks," in *IEEE GLOBECOM*, Wahington (DC, USA), Nov. 2007, pp. 464–473.

[14] ——, "Enhancing spatial reuse in ad hoc networks by carrier sense adaptation," in *IEEE MILCOM*, Orlando (FL, USA), Oct. 2007.

[15] L. Badia, A. Erta, L. Lenzini, F. Rossetto, and M. Zorzi., "A Physical Model Scheduler for Multi-Hop Wireless Networks Based on Local Information," in *IEEE MASS*, Atlanta (GA, USA), Oct. 2008.

[16] K. Stamatiou, F. Rossetto, M. Haenggi, T. Javidi, J. R. Zeidler, and M. Zorzi, "Delay and throughput characterization of random multi-hop networks," in *IEEE INFOCOM*, 2009 (submitted).

[17] A. Munari, F. Rossetto, and M. Zorzi, "A new Cooperative Strategy for Deafness Prevention in Directional Ad Hoc Networks," in *IEEE ICC*, Glasgow (UK), Jun. 2007.

[18] ——, "Cooperative cross layer mac protocols for directional antenna ad hoc networks," *ACM MC2R*, vol. 12, no. 4, Apr. 2008.

[19] F. Rossetto, M. Meyer, and M. Zorzi, "Packet Centric vs. Radio Centric Link Layer Approaches: a quantitative analysis," in *IEEE VTC Fall*, Baltimore (MD, USA), Oct. 2007.