To my mother and my father

Più rileggo questo elenco più mi convinco che esso è effetto del caso e non contiene alcun messaggio. Ma queste pagine incomplete mi hanno accompagnato per tutta la vita che da allora mi è restata da vivere, le ho spesso consultate come un oracolo, e ho quasi l'impressione che quanto ho scritto su questi fogli, che tu ora leggerai, ignoto lettore, altro non sia che un centone, un carme a figura, un immenso acrostico che non dice e non ripete altro che ciò che quei frammenti mi hanno suggerito, né so più se io abbia sinora parlato di essi o essi abbiano parlato per bocca mia.

Umberto Eco,
"Il nome della rosa"

E io risposi: "La mia parola non ha ancora spostato alcuna montagna, e ciò che io ho detto non ha raggiunto gli uomini. E' vero, io sono andato dagli uomini, ma non sono ancora giunto da loro".

Friedrich Nietzsche,
"Così parlò Zarathustra"

-Sarà un bel bambino che avrà tante avventure e troverà ciò che cerca,- disse la fata Reàna entrando dalla finestra e ponendosi sul calto del letto.

Giuliano Scabia,
"Nane Oca"

Yes we can

Barack Obama

To my mother and father. We have been divided early, but you gave me all I need. I won't forget.

To my sister Annalisa, who has been sustaining me by actions, words and example.

To my brother Davide, who initiated me to engineering.

To my beloved Giulia. Your delicate but strong presence raised me up again after a long time I had been on the floor, and it layed several milestones on the road I'm walking, with you.

To my friends Neno, Enrico, Carlo and Sergio. For the laughs of many days, and because you were all around me that day.

To Paolo. We started together this adventure, and since that time you have been much more than a colleague. Thanks my friend!

Also thanks to Leonardo, who not only has a wonderful mind, but also a huge heart.

To my advisor prof. Michele Zorzi, who taught me how to do research and gave me the chance to pursue my ideas. Also thanks to prof. Urbashi Mitra, whose support has been truly important to me.

I want just to mention an important figure in my life I met only through his books. Thanks Tiziano!

Marco

Contents

List of Figures

Abstract

Multiuser detection and multiple-input multiple-output systems are emerging as key technologies to deploy high-performance networks. Despite the great amount of research of the recent years on the development of transmitter/receiver architectures, limited work on MUD/MIMO based networking has been done so far.

In this thesis, we focus on ad hoc networks exploiting these advanced architectures to increase the parallelism of communications. This appears to be an important issue, as nowadays ad hoc networks implementing carrier sense and collision avoidance mechanisms achieve poor spatial reuse and incur failure due to the well-known hidden and exposed terminal problems.

In this kind of network, where nodes simultaneously access the channel, a carrier sense approach is evidently not a viable solution, or however not effective, especially when considering successive interference cancellation based receivers. We present and investigate in depth different possible solutions for the control of interference in the network.

We propose a coordination scheme for MIMO-BLAST ad hoc networks that enables distributed access control by receivers. We also address asynchronous scenarios in which nodes are allowed to access the channel without any carrier sense. In this case, interference distribution and control mechanisms have complicated interactions, that must be taken into account when designing the network protocols.

As efficiency is a key issue in this scenario, we investigate techniques that enable adaptability and resilience to channel conditions' variations, such as cooperation and advanced error control schemes.

Besides the wide discussion on the design of the network, an important contribution of this thesis is represented by the many novel analytical tools developed throughout the chapters.

Sommario

Le architetture di livello fisico basate su Multiuser detection e sistemi multiple-input multiple-output stanno emergendo come tecnologie chiave per lo sviluppo di reti wireless ad elevate prestazioni. Nonostante queste architetture siano state largamente studiate a livello di sistema punto-punto, manca ancora una completa comprensione di come poi questi si integrino con la rete nel suo complesso.

Lo studio presentato in questa tesi discute l'impiego di queste tecnologie al fine di aumentare il parallelismo delle comunicazioni nelle reti *ad hoc*. Questo rappresenta un'evoluzione importante per le reti moderne, ed è tematica attualissima nella comunità di ricerca. Infatti, le implementazioni correnti di reti *ad hoc*, basate su protocolli di carrier sense e collision avoidance, raggiungono prestazioni limitate da un basso risuo spaziale della risorsa radio, anche a causa di inefficenze intrinseche come ad esempio i ben noti problemi di terminale esposto e terminale nascosto.

Al fine di avere una rete *ad hoc* che sfrutti appieno le potenzialità di queste architetture, che offrono una considerevole resistenza all'interferenza, è evidente la necessità di pensare ad uno scenario dove link tra diverse coppie sorgente-destinazione siano attivati nella stessa area. I protocolli di carrier sense e collision avoidance, il cui obiettivo è appunto quello di evitare comunicazioni concorrenti e potenzialmente mutualmente distruttive, sono chiaramente inadeguati a gestire questo tipo di rete. In questa tesi vengono presentati e discussi diversi approcci e tematiche per il controllo dell'interferenza in questo scenario.

La principale difficoltà in uno scenario di attivazione simultanea dei link è l'imprevedibilità e l'alta variabilità delle condizioni di canale che interesseranno la trasmissione del pacchetto dati.

La prima classe di approcci studiati propongono uno schema che permette la coordinazione tra i nodi della rete. Una fase iniziale di handshake, dove le richieste di attivazione dei link da parte delle sorgenti avvengono simultaneamente, permettendo ai ricevitori di essere in parte coscienti dell'interferenza che interesserà la ricezione dei dati nella fase successiva.

Una proposta alternativa ad approcci che propongono la riduzione dell'incertezza sull'interferenza, e il conseguente controllo sull'attivazione dei link, prevede la costruzione di meccanismi in grado di fronteggiare l'imprevedibilità delle condizioni di canale. In questa tesi vengono studiate nel dettaglio le complesse tematiche di uno scenario in cui le trasmissioni non vengono imbrigliate in rigide strutture di comunicazione e i nodi sono liberi di accedere in modo asincrono. Viene dimostrato come il meccanismo di controllo dell'errore sia un elemento chiave in questo tipo di reti. Infatti, non solo l'incertezza sulle condizioni di canale rende necessaria l'implementazione di protocolli atti al recupero di pacchetti errati, ma si instaurano delle complesse interazioni tra il meccanismo di controllo dell'errore e la statistica dell'interferenza, di cui bisogna tener conto nel progetto complessivo del sistema.

All'interno di questa seconda classe di reti, vengono inoltre studiate tecniche che aumentino l'efficienza delle comunicazioni. Nello scenario affrontato, aumentare l'efficienza delle singole trasmissioni ha un evidente impatto anche sul bilancio globale d'interferenza nella rete. In particolare vengono considerati protocolli di cooperazione e meccanismi avanzati di controllo dell'errore.

Oltre alla discussione e alle diverse proposte per il progetto dei protocolli di comunicazione contenuti in questa tesi, vengono presentati alcuni nuovi strumenti analitici per la valutazione delle prestazioni e lo studio del della rete.

Chapter 1

Introduction

Contents

1.1 Preliminary Discussion

Ad hoc networks are made of autonomous nodes that can connect to each other without the need for infrastructured administration or maintenance. Wireless technologies potentially enable anytime–anywhere networking, allowing nodes to, *e.g.*, share data and access distributed services in a seamless and easy way. Furthermore, such decentralized networks allow for fast deployment in emergency or military scenarios, besides being suited for commercial applications and for quick communications setup in any environment where a cabled network is infeasible or not affordable.

The fully distributed and autonomous framework poses fundamental design challenges. The design of effective and efficient distributed protocols controlling the access to a shared radio resource has been the focus of considerable research effort in the past decade. The task is to achieve reasonable aggregate throughput of the network while preserving connectivity, fairness and communications' reliability. The fully distributed scenario, where nodes organize themselves in a network, the lack of an *a priori* knowledge of the topology, and the channel impairments due to fading and shadowing, make this task very challenging.

Most practical implementations rely on the well known carrier sense multiple access with collision avoidance (CSMA/CA) protocol, described in the 802.11 standard [8]. The carrier sense (CS) mechanism is based on a threshold. The threshold defines two channel states, namely *busy* and *idle* channel. A node with a packet to be delivered listens to the channel in order to detect the presence of ongoing communications in its neighborhood. If the perceived power is below a certain threshold (idle channel) then the node starts to

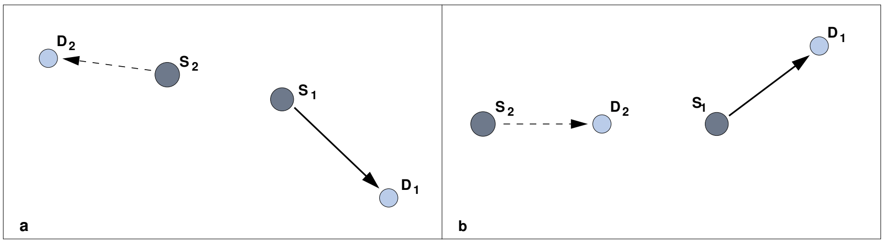


Figure 1.1. *Example of topologies. S_1, S_2 are sources with D_1 and D_2 as destinations, respectively.*

transmit the packet, otherwise (busy channel) it refrains from transmission. The goal of CS is to protect ongoing communications from interference, preventing multiple transmissions in the same neighborhood.

It is clear that the CS mechanism can fail due to random noise, fading and the different positions of the source with respect to its intended destination. Noise can make a source perceive an idle channel as busy. Fading and different geographical position can make a source perceive a different level of interference with respect to that perceived by its destination. Furthermore, the source, measuring the incoming interference power, becomes somewhat *aware* of the presence of transmitters in its neighborhood, but does not gain much information about the interference its transmission would generate to their receivers. Thus, on one hand, a source is not able to know interference at its intended receiver, so that it cannot guarantee the successful delivery of its own packet. On the other hand, it does not know how its transmission will affect active receivers. Consider the topologies in Fig. 1.1, where source S_2 has to transmit a packet to D_2, while S_1 is already transmitting a packet to D_1. In the case depicted in Fig. 1.1.a, S_2, due to the transmission by S_1, is likely to sense a busy channel, and refrain from transmission. Nevertheless, the two sources are likely to generate limited interference at the receiver of the other communication. Thus, CS allows the activation of only one link at a given time, while both could be potentially simultaneously activated. Conversely, in the case depicted in Fig. 1.1.b, S_2 is likely to sense the channel as idle even during a transmission by S_1. Thus, if S_2 starts to transmit and S_1 is transmitting to D_1, D_2 reception would probably be affected by an overwhelming interference.[1]

Thus, Collision avoidance (CA) mechanism improves the performance achieved by CS in terms of probability of success of an ongoing communication. CA provides the exchange of short control messages, namely request to send (RTS) and clear to send (CTS), before a source is allowed to transmit data over the channel. If a source senses an idle channel, it transmits an RTS to its intended destination, that replies with a CTS if it successfully decoded the RTS and senses the channel as idle. RTS and CTS reception causes nodes to avoid transmission for a time interval included in the messages (that indicates the duration of the communication they are associated with). Thus, by CA, interference at the receiver is checked before data transmission and CTS transmission prevents transmission by receiver's

[1] These scenarios are generally referred to as *exposed* and *hidden terminal* problems.

neighbors, so that the probability of successful decoding is increased.

Hence, the CA mechanism reduces the probability that multiple links are activated at a given time in a certain area, improving the success probability of performed transmissions. Fast fading and random noise can reduce the effectiveness of this mechanism, as they can cause potential interferers to fail control messages decoding. However, the main issue of CA is that it is in many cases excessively conservative, blocking communications that could have been successfully deployed.

From the previous discussion, it is clear that, though the most widely implemented protocol for ad hoc networks, CSMA/CA achieves poor performance in terms of aggregate throughput and fairness. It is possible to trade off spatial reuse and reliability changing the sensing threshold and the power with which RTSs/CTSs are transmitted. However, the mechanism is intrinsically sub-optimal, and it is not possible to achieve optimal performance by reducing or increasing the portion of area locked by control messages reception and channel sense.

A higher degree of coordination, that anyway requires the exchange of further control messages, is a possible way to achieve an improved spatial reuse. It is important to observe that in order to guarantee the reception of the control packets we have to use something similar to CS or a time division multiple access (TDMA) scheme. Therefore, a coordination phase would have a considerable cost in terms of throughput and latency.

Thus, the big question is how to control interference in the network without recurring to complicated and costly approaches. In this thesis, we take another approach. Instead of trying to design a protocol that avoids multiple simultaneous transmissions in the same area, we consider the use of advanced physical layer architectures that are intrinsically resilient to interference, such as multiuser detection (MUD) and multiple-input multiple-output (MIMO) systems. It is clear that, if we reduce the effective interference perceived by receivers as a function of the incoming interference power, we can deploy a greater number of simultaneous transmissions in a given area of the network. In a certain sense, we are reducing the mutual coupling between simultaneously active links. In traditional physical layer architectures, interference couples communications in such a strong way that often analytical models make use of the concept of *collision*. In a collision model, if an interfering node is active within a certain range of a given receiving node, the latter fails to decode the packet. As discussed before, the design of traditional systems is based on a collision model, as the goal is to avoid interfering transmissions within a certain area surrounding the transmitter and the receiver.

In MUD and MIMO systems, interference has a lower coupling effect on communications. Thus, the design of protocols cannot be based on a collision model, as multiple simultaneous communications can be deployed in the same area. The use of MUD and MIMO technologies in ad hoc networks opens a hard design challenge. The control of nodes' access in a network where potentially more than one link can be activated in an area of the network is a problem that cannot be solved with a criterion similar to that of CSMA/CA.

In our scenario, carrier sense is not a viable solution for access control. In fact, we cannot define a threshold of tolerable interference in many powerful receiver architectures. For instance, if we use a successive interference cancellation (SIC) receiver, a powerful interfering signal can be successfully decoded and cancelled from the overall received signal, so that its contribution to the perceived interference is nulled or at least greatly reduced. Moreover, the effective SINR of a signal is often a function of the number and the contribution to the received interfering power of each of the interfering nodes. Thus the estimation of the overall interference, as that provided by CS, does not give an accurate perception of the interference perceived at the receiver.

A straightforward extension of CSMA/CA rationale, in which a maximum number of users is allowed to access the channel simultaneously, appears restrictive, and, anyway, of difficult realization. In fact, the maximum number of simultaneous communications is, in our case, even more dependent on the topology of the network and the channel coefficients of the various links than in the standard receiver case. Moreover, it is not easy for a node to estimate and keep track of the number of ongoing communications in its neighborhood. Estimation through incoming power without a detailed knowledge of channels statistics can lead to significant error probability. Nodes can estimate the number of ongoing communications decoding control messages. Anyway, nodes are deaf to incoming signals during transmission and can miss a significant fraction of control packets exchange in the network.

Despite the great amount of work done on the MUD PHY layer, limited work on CDMA MUD network analysis and design has been done so far. In [1], Tse *et al.* define the effective interference and bandwidth for large systems with power control and random spreading sequences, and investigate the capacity achieved employing several multiuser receivers for a single–cell cellular network. Nie *et al.* [2] analyze the capacity of multi–cell cellular environments through reverse–link analysis. Ulukus *et al.* in [3] propose an iterative algorithm for optimizing the capacity of cellular networks with DS–CDMA MF systems.

Moreover, very little work exists regarding the issues arising when using MUD systems in ad hoc networks. In [4,5], the authors analyze a single-code DS-CDMA MF system based on the ALOHA access protocol. In [6] a system with iterative multiuser detection and DS-CDMA MF detection for decoding the packet and the preamble, and with an ALOHA based access scheme is presented. [7] analyzes the performance of a DS-CDMA MF slotted system with ARQ and random arrivals through a processor-sharing system model.

Anyway, the design of an ad hoc network exploiting MUD or MIMO technologies is still an open problem, and we believe that there are a lot of issues that need an in depth investigation.

There is an important aspect that we want to highlight. In CSMA/CA interference is treated mostly during access control, and if a node successfully gains access to the channel and no protocol errors occur, it can be almost neglected when designing transmission parameters and error control protocols. In a scenario with multiple simultaneous access, interference evidently must be considered in the design of each part of the system. Access control does not have the goal of avoiding interference, but to generate an interference with a certain characterization. Thus, rate, power and error control have to face with new source of uncertainty.

In fact, in this scenario there is a complex interaction among interference and control mechanisms. Consider, for instance, error control. A retransmission-based error control scheme increases the birth rate (due to retransmission of packets erroneously decoded) of interfering transmissions. A packet encoding error control scheme results in longer transmissions. In both cases reliability comes at the cost of an increased interference load in the network. However, while in the former case the channel variation rate increases, as we get a greater number of short transmissions, the latter correlate interference, as we have a lower number of longer transmissions. Channel unpredictability affects the effectiveness of rate control decisions, that are generally made during the handshake. Note that a degraded success rate causes a greater retransmission probability in the former case, so that the interference level further increases.

In this thesis we investigate in depth issues concerning interference in ad hoc networks with multiple simultaneous access. Analytical tools able to manage the kind of interactions discussed above are still missing in the literature. Thus, besides the design proposal, the investigation of specific issues and the discussion provided in the various chapters, one of the important contribution of this thesis is the development of the instruments needed for our deductions. We will discuss more in detail the content and the organization of the thesis in the next section.

1.2 Methodology and Organization of the Thesis

An effective deployment of ad hoc networks with multiple simultaneous access cannot extend traditional control mechanisms. We have to design the network with a different view in mind. As introduced above, the goal of control is rather different, as we want to generate an interference that matches the physical layer capabilities, rather than avoiding interference.

The presence of non-negligible interference during the transmission results in several effects that are not well understood yet. Therefore, we have to attack the problem from a broad perspective, investigating different solutions and approaches. We can divide the approaches taken in this thesis into two main classes.

In the former class of approaches, that we refer to as *coordination*, nodes coordinate their transmissions exploiting a TDMA structure in order to avoid overwhelming interference at the receivers. We called this approach *coordination* as nodes synchronize[2] their communications to allow receivers to acquire information on how interference will affect the reception of their intended packets.

[2] We mean that nodes share the same frame/slot time structure, we do not require symbol synchronization in our frameworks.

The latter approach addresses asynchronous communications, where nodes can start transmission at any time. In this case, the transmitter and receiver have limited information on the interference that will affect the communication, as interfering transmissions can start and end unpredictably.

The two approaches are completely different and present different interference issues.

In the former approach the network tries to prevent destructive interference (and in particular we propose a distributed receiver-driven access control). Thus, once resources are properly allocated, transmissions can be performed with limited uncertainty, derived by the partial knowledge of which of the links will be activated. However, a conservative allocation grants considerable reliability, and strong adaptability to channel variations is not strictly required to ensure the robustness of the system.

In Chapter 2, we present a network design following the first approach. We consider a MIMO-BLAST architecture where nodes are equipped with multiple antennas. We set up a flexible link activation scheme, based on a TDMA structure following the considerations listed above.

In the second approach, adaptability to quick channel variations and the interactions described in the previous chapters become an important issue. In Chapter 3 we thoroughly study interference effects in a network with error and rate control. We stress interference setting up a light load control, based on a random backoff mechanism, while nodes do not perform channel sensing before accessing the channel. We develop a novel recursive analytical framework for the study of interference distribution in the network with various error control principles. We present some relevant observations on how to develop error control in our scenario.

In Chapter 4, we provide an insightful investigation of the benefits granted by cooperative behaviors to our network. In particular, we focus on the integration of cooperative transmission with hybrid ARQ. Efficiency is a key issue in our network, as it influences the interference load in the network. Cooperation, providing improvement in terms of channel diversity, can potentially increase the efficiency of the communications. We also show how the intrinsic parallelism of communications of MUD systems allows an effective deployment of cooperation.

In Chapter 5, we derive the optimal access and transmitted power control for our scenario. In this case we investigate a centralized scenario with full knowledge at the controller. In our opinion, the analysis of the structure of the optimal policy provides some further insight for our design task. We especially focus our interest on the interactions between the Markov chains modeling the various sources due to interference. Our framework considers an infinite horizon average cost per stage problem based on Dynamic Programming. The solution of this problem poses several technical challenges concerning the structure of the Markov chain modeling the channel.

We also introduce the concept of *process distortion*, that is, an interference measure based on how transmission from other nodes interferes with the overall Markov process. This has as an important application to cognitive networking, where unlicensed users can access the channel with a constraint on the performance loss caused to primary users. However, can be extended to design the level of interference a node can generate to other nodes of a general ad hoc network to optimize performance.

In Chapter 6 we present an analytical framework for the study of access control for ad hoc networks based on the recursive update of interactive Markov chains. Although our proposal investigates a traditional scenario, we include this part of the work in this thesis, as it represents a step forward in network modeling, that opens up to further developments.

Appendix A briefly summarizes some additional work addressing issues out of the main scope of this thesis. In particular, we addressed the modelling of error control schemes through Markov processes.

Appendix B reports a complete list of my contributions to the literature.

It is possible to observe that the thesis is quite heterogeneous as for the issues addressed in the various chapter and the methodology of the investigation. However, when trying to attack a so wide and unexplored field, concerning the design of a new way of networking, we believe that to focus on a single scenario and a single framework would be a limiting approach, that potentially fails to give insight on the big picture.

While addressing the various issues of this thesis we consider various scenarios and frameworks, inheriting in most of the cases the language, notation, assumptions, tools and the methodology of investigation from the related literature.

References

[1] D. N. C. Tse and S. V. Hanly, “Linear Multiuser Receivers: Effective Interference, Effective Bandwidth and User Capacity,” *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 641–657, Mar. 1999.

[2] H. Nie, P. T. Mathiopoulos, and G. K. Karagiannidis, “Reverse link capacity analysis of cellular CDMA systems with controlled power disparities and successive interference cancellation,” *IEEE Trans. Wireless Commun.*, vol. 5, no. 9, pp. 2447–2457, Sep. 2006.

[3] S. Ulukus and A. Yener, “Iterative Transmitter and Receiver Optimization for CDMA Networks,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1879–1884, Nov. 2004.

[4] A. Yener and R. D. Yates, “Multiuser access detection for CDMA systems,” in *CISS'98*, pp. 17–22.

[5] D. Raychaudri, “Performance analysis of random access packet switched Code Division Multiple Access systems,” *IEEE J. Select. Areas Commun.*, vol. 29, pp. 895–901, Jun. 1981.

[6] C. Schlegel, R. Kempter, and P. Kota, “A novel random wireless packet multiple access method using CDMA,” *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1362–1370, Jun. 2006.

[7] B. Lu, X. Wang, and J. Zhang, “Throughput of CDMA data networks with multiuser detection, ARQ, and packet combining,” *IEEE Trans. Wireless Commun.*, vol. 52, no. 5, pp. 811–822, May 2004.

Chapter 2

Coordination

Contents

2.1 Introduction

As we briefly discussed in the introduction, one of the possible choices is to build a structure for the communications in which nodes exchange control packets and coordinate transmission in order to avoid receivers' overload. The deployment of fast and effective coordination phases is granted by MUD.

We will discuss the details of the system later in the chapter. However, thanks to MUD, we superpose the various phases of the communications of the nodes via a TDMA structure. In particular, we divide time into frames, with each frame divided into four slots, here nodes transmit request, confirmation, data and acknowledgment packets, respectively. Thus, sources simultaneously send data transmission request and destinations simultaneously send out confirmation packets. In this way, receivers, through decoding requests to transmit intended to other destinations, can estimate the interference in its neighborhood that will affect its communication during data transmission. The destination can use this important information to decide how many and which transmission requests grant, thus realizing a distributed access control mechanism. We stress that in these system, control protocols must be designed with physical layer in mind as resource allocation performed by destinations has to be strongly tied with receiver's architecture interference resilience capabilities to be effective.

The scheme we present in the following is a receiver-driven access control. It is interesting to observe that, in our scheme, receivers control access only for what concerns data transmission, while access during request and confirmation phases is left unruled. Thus, the level of parallelism and resilience to interference during control messages phases, as many potentially many sources can simultaneously access the channel without carrier sense or interference-based decisions is of fundamental importance. Anyway, we implement a random backoff mechanism to diminish congestion probability. A careful design of the whole system, taking into account for the extreme importance of control messaging, is the key to a extremely performing network.

In the rest of the chapter, we will focus on a powerful MIMO-BLAST multiple antenna physical layer, that grants considerable interference resilience, multiuser detection and transmitter multiplexing capabilities. We exploit the capabilities of the BLAST system allowing each of the active transmitters and receivers to establish multiple links at the same frame. This can partially reduce the additional delay resulting from the rigid TDMA structure.

With the shift towards higher frequency bands, the integration of multiple antennas in a single terminal is progressively becoming feasible. The use of multiple antennas has shown great promise in providing higher spectral efficiencies on wireless links than traditional communication systems, but their adaptation to ad hoc networks is non-trivial nonetheless.

Multiple antennas allow for more advanced communication paradigms. Examples include: beamforming, whereby nodes can steer transmissions so as to cover a certain portion of space; diversity, which greatly mitigates the effects of multipath propagation; SIR maximization through array processing, whereby the array reception pattern can be adapted to amplify or suppress the power received from certain directions.

These techniques could have a great impact in ad hoc networks. Directional transmissions would both decrease interference and amplify power gain toward wanted recipients, increasing spectral efficiency and spatial reuse. On the other hand, they introduce further challenges, such as how to deal with gain asymmetries (different array gains at different nodes) and deafness (a node is not aware of what other nodes do). Some works on this topic are summarized in Section 2.2.

The transmission and reception of signals through multiple antennas can be holistically viewed as a MIMO system. Since the pioneering work by Foschini [1], MIMO has attracted significant attention as the key technology to achieve high spectral efficiency exploiting rich scattering environments. MIMO enables the protection of communications in the "space" (*i.e.*, antenna) domain, by processing and transmitting signals through different antennas, according to predefined schemes (*e.g.*, Space–Time Codes, STC [2]). A subset of STCs, namely Layered STCs (LSTC), jointly use encoding and parallel transmissions, sending out multiple flows using different array elements. A special case of LSTC is V–BLAST [3], where the encoding component is absent, and all resources are used for parallelizing transmissions. This approach is also called Spatial Multiplexing (SM). It has been shown [4] that there exists a tradeoff between diversity and SM gain in MIMO networks: V–BLAST achieves the greatest SM depth, whereas codes such as [5] are optimal in a diversity sense.

MIMO techniques can be applied to ad hoc networks with significant benefits. If multiple bit sequences are sent by different nodes, each using multiple antennas, all streams can be taken as a separate contribution by the intended receiver. If some channel information is available, the receive antennas' outputs can be recombined and processed such that the sent data can finally be recovered. The primary consequence is the coexistence of multiple data packets in the network (*i.e.*, without collisions), provided that some degree of coordination is obtained among transmitters. Moreover, by splitting a single packet transmission among multiple antennas (e.g., with V–BLAST), a node is allowed a higher raw bit rate, which is proportional to the number of antennas used [6,7]. We wish to highlight that if full channel state information at the transmitter (CSIT) is available, beamforming techniques can exploit it and lead to better link-level performance. However, as explained more extensively in Section 2.2, it might prove hard to devise protocols that coordinate independent nodes so that their use of beamforming does not give rise to deafness or similar problems. On the contrary, MIMO transmissions can provide good performance even in the absence of CSIT, without resorting to explicit power gain shaping at the transmitter. In our work, we will focus on this specific scenario.

The advantages described above encourage the consideration of a MIMO physical layer in ad hoc networks, but leave many issues open about correct management of its potential. Using a more powerful physical layer in combination with existing MAC protocols for ad hoc networks (such as 802.11 [8]) may not necessarily be the best choice. A better design paradigm should jointly account for PHY and MAC features in a cross–layer fashion, and strive to take advantage of all available degrees of freedom, for example by allowing some exchange of information between different layers.

In this Chapter we investigate the advantages, drawbacks and possible tradeoffs that arise at the MAC layer when dealing with ad hoc networks with V–BLAST at the PHY layer. Exploiting this knowledge enables an effective design of completely distributed channel access mechanisms that manage transmission requests and grants[1] to trade off higher bit rates (*i.e.*, perceived throughput) for more resilient multiuser detection (*i.e.*, interference rejection) through a proper use of V–BLAST's successive interference cancellation capabilities.

This work has been done in collaboration with Paolo Casari.[2] Stefano Tomasin contributed to the deployment of the performance approximation for the BLAST receiver. The reference papers are [J1ml, J4ml, C02ml, C03ml, C04ml, C05ml, C07ml, C09ml] (see Appendix B).

The Chapter is organized as follows. In Section 2.2 we summarize the literature on the use of multiple antennas in ad hoc networks, for both beamforming and MIMO. In Section 2.3 we give an overview of the decision feedback multiuser detector and spatial multiplexing system operating at the physical layer and . In Section 2.4 we introduce a new MAC protocol, and describe in detail the proposed policies for channel access and traffic management in Section 2.5. In Section 2.6.2, we present numerical results assessing the performance achieved by the proposed protocols. The performance analysis of BLAST is described in Section 2.7, where we introduce the Gaussian, enumeration and pruned tree techniques. Extensive numerical results for both link and network level performance are discussed in Section 2.8, comparing bit-by-bit simulations and the semianalytical techniques. Section 2.9 presents an extensive investigation of network issue via simulations carried out exploiting the performance approximation of the physical layer.

2.2 Related Work on Ad Hoc Networks with Multiple Antennas Systems

The integration of multiple antennas in ad hoc networks is a relatively recent topic. In [9], the authors focused on purely directional transmissions and designed Multihop MAC (MMAC), a routing-aware protocol that bridges longer distances by both coordinating farther nodes using RTS/CTS exchanges over multiple hops and exploiting the higher gains and lower overall interference achieved by directional communications.

As introduced in Section 2.1, such protocols suffer from "deafness." For example, directional transmissions or half-duplex operations may leave nodes unaware of ongoing communications (deaf), which makes distributed coordination difficult. A solution based on

[1] By grant, we indicate a Clear–To–Send message that enables a transmission. In the following, we will interchangeably use either term.

[2] Part of the work presented in the following is included also in his thesis.

busy tones is provided in [10], which requires more complex hardware. It should be noted that receiver-side omnidirectional reception is possible even in the presence of receive beamforming, *e.g.*, by implementing parallel processors that weight the antenna outputs differently: each processor could beamform toward a different direction, ultimately obtaining omnidirectional coverage from the superposition of the processors' outputs.

Ramanathan *et al.* [11] proposed UDAAN, a set of integrated MAC, routing, neighbor discovery and signaling protocols for ad hoc networks with directional antennas. They also built a field demonstration using horn antennas that is, to the best of our knowledge, the most comprehensive mobile ad hoc network testbed deployed so far.

In [12], a MAC protocol is considered where nodes send and receive data directionally thanks to some topology awareness. In [13] another MAC protocol is proposed with directional RTS/CTS exchange. Thus, RTSs possess a longer reach, but are sent directionally in one beam at a time, so that many transmissions are needed to cover the whole horizon. This approach notifies farther nodes, thus mitigating deafness and establishing longer links, but incurs longer handshake latencies.

These papers present very interesting contributions, showing benefits of directional communications in ad hoc networks. However, very simplified propagation and antenna models are typically taken into account. This may not be sufficiently accurate, especially when achieving directionality through arrays of simple (*e.g.*, dipole) antennas.

A different approach is to regard the whole set of multiple transmit and receive antennas as a MIMO system. In this case, all transmissions are omnidirectional, and multiple superimposed signals can be separated and detected through some specific signal processing at the receiver. The problem of studying and optimizing MIMO links under different objectives and constraints has been widely addressed in the literature. Two recent books on the topic are [2, 14]. The specific application of MIMO systems to ad hoc networks, however, has received less attention.

MIMO systems are indeed a means of performing Multi-Packet Reception (MPR). In [15] and [16] (see also the references therein), the impact of MPR on random access MAC protocols is considered. A Multi-Queue Service Room protocol is envisioned as the optimal solution for reaching maximum throughput with random access, and is compared to less complex but suboptimal protocols. One of the main conclusions is that cross–layer sharing of simple parameters is crucial to successfully design protocols for MPR channels. Recently, an access scheme to exploit MPR with CDMA, while meeting QoS requirements, was also proposed in [17], while the effect of MPR capabilities on the throughput capacity of ad hoc networks was studied in [18].

In many papers an information-theoretic point of view is taken, by defining throughput as the maximum mutual information between a received and a transmitted signal. Throughput is then optimized, *e.g.*, under maximum power constraints [19]. These works are interesting, but typically require the ideal assumption that the channel capacity is exactly reached, and often neglect specific networking issues, for instance a particular MAC implementation.

A networking-based approach is carried out in [20] with MIMA-MAC, an access protocol specifically designed for ad hoc networks with up to two antennas per node. The devised MAC includes a contention-based and a contention-free period, used to set up links among receivers using two antennas to decode data coming from up to two transmitters using one antenna each. The small number of nodes considered and the constraint to use at most one antenna for transmission represent significant limitations.

In [21], the authors propose that nodes transmit busy tones over one of the sub-bands provided by a MIMO–OFDM system for signaling their intention to transmit. Prior to sending the busy tone, each transmitter chooses a random channel sense time. If it hears more than a certain number of tones it defers transmission, otherwise it activates its own tone and sets up the link.

Another very interesting work on MIMO ad hoc networks is [22]. A MAC protocol is designed based on IEEE 802.11 DCF [8], and is modified to exploit spatial diversity. RTSs and CTSs are used along with PHY preambles that allow to estimate the channel and consequently decide the correct transmission data rate. STCs are used to achieve full diversity. An analysis of the impact of MAC on routing is also carried out, evaluating the relation between the delay incurred before sensing a free channel and the advancement obtained with a one-hop transmission. Directional antennas are explored as a special case of multiple antenna communications.

In [23], Time-Reversal (TR) STCs are considered. After showing that an optimal maximum likelihood decoder achieves the maximum diversity order in an intersymbol-interference multiple-access channel, the authors prove that using lower complexity linear MMSE detectors based on the TR-STC structure achieves only slightly deteriorated performance. MIMO links and STCs have also been used in [24] for addressing the problem of efficient broadcasting in ad hoc networks with multiple antennas.

A different method for managing radio links with multiple antennas is given in [25]. There, a centralized controller is able to estimate concurrent resource usage and to schedule links to exploit the benefits of MIMO such as SM and interference suppression, along with increased transmit rate. The final objective is a proportional fair scheduling of transmissions, that accounts for bottleneck links, and is achieved by graph coloring. An online algorithm is also designed. This last contribution, although interesting, makes some very strong assumptions on the PHY layer, *e.g.*, that any transmission uses the full channel capacity and that signaling at the MAC level is perfect.

When assessing the performance of an ad hoc network with multiple antennas, an accurate characterization of the underlying PHY level is of paramount importance. In MIMO communications, a detailed model of the receiver capabilities is needed, in particular for nonlinear multiuser detectors like BLAST [1,26].

From the MAC point of view, some of the works above rely on the exchange of signaling messages among communication parties. MIMO links are inherently omnidirectional and not prone to deafness in the sense used for directional antennas. Moreover, with the use of a proper receiver and with a sufficient degree of spatial diversity available, SM would allow significant bit rate improvements.

A number of issues arise when designing MAC protocols for MIMO ad hoc networks [27]. In our approach, we start from an accurate PHY model and construct MAC and link management protocols in a completely cross–layer fashion, in order to exploit all available degrees of freedom. Our protocol heavily relies on the exchange of information between the PHY and MAC layers, with a twofold objective. First, unlike 802.11, we want the MAC to coordinate transmissions in order to favor parallel communications, while avoiding channel overload. Secondly, we want to drive the reception of SM signals so that wanted ones are sufficiently protected from interference, using a mechanism to prevent some nodes from transmitting if needed. In order to do this, we let the MAC use the knowledge of ongoing neighboring handshakes to decide whether or not to grant some requested transmissions, so that the interference cancellation capabilities of the MIMO receiver are properly exploited without its being overloaded. We will also show how our cross–layer approach significantly outperforms a traditional layered solution.

2.3 System Description and Transmitter/Receiver Structure

We consider a packet wireless network with nodes having N_A antennas. As transmission scenario we assume a frequency–flat block fading channel. Nodes can both transmit and receive information, in a peer-to-peer fashion. All nodes are in communication range of each other, *i.e.*, we consider a completely connected network. To keep the analysis simple, we assume bit-synchronous node operations, although this is not a requirement when using BLAST receivers (see [26] and references therein).

2.3.1 Node operations

Transmitting nodes—When operating as a transmitter, the generic node p splits each packet into sub-packets called *streams*. Each stream is sent from one antenna, so that u_p streams are multiplexed using u_p antennas during transmission. Assuming that N_{Tx} nodes with indices $\{1, 2, \ldots, N_{\mathrm{Tx}}\}$ are transmitting, the total number of simultaneously transmitted streams is $U = \sum_{p=1}^{N_{\mathrm{Tx}}} u_p$. We can identify each transmit antenna with a transmit antenna index (TAI), starting with the first antenna of the first node and ending with antenna $u_{N_{\mathrm{Tx}}}$ of node N_{Tx}.

Let us define the column vector $\boldsymbol{s}'(t) = [s'_1(t), \ldots, s'_U(t)]^T$ whose entry $s'_i(t)$ is the symbol transmitted from antenna with TAI i at time tT, where T is the symbol period and ^{T} denotes the transpose operation.

We assume that node p always transmits with a total power P_{tot} that is uniformly distributed among the u_p antennas. Hence, the power of the transmitted data signal on the antenna with TAI i belonging to node p is $\sigma^2_{s'}(i) = \mathrm{E}[|s'_i(t)|^2] = P_{\mathrm{tot}}/u_p$ where $\mathrm{E}[\cdot]$ denotes expectation. For the sake of a simpler notation, we omit in the following the time index t in all signals.

Receiving nodes—When operating as a receiver, the generic node q uses all the N_A antennas and the column vector of the N_A received samples can be written as $\boldsymbol{r}^{(q)} = \tilde{\boldsymbol{H}}^{(q)}\boldsymbol{s}' + \boldsymbol{\nu}'^{(q)}$, where $\boldsymbol{\nu}'^{(q)}$ is the column noise vector of length N_A, and $\tilde{\boldsymbol{H}}^{(q)}$ is the $N_A \times U$ channel matrix whose entry $\tilde{\boldsymbol{H}}^{(q)}_{\ell,m}$ represents the complex baseband channel gain between the transmit antenna of TAI m and the ℓth receive antenna.

In order to limit the node complexity, we assume that each receiving node has a partial knowledge of the channels, *i.e.*, node q can only know the channel gains associated to N_d transmit antennas (called *internal antennas*, IA), whose TAIs are in the sub-set $\mathcal{N}^{(q)} = \{n_1, n_2, \ldots, n_{N_d}\}$, for which we assume perfect channel estimation. Without restriction, we assume that the TAIs of known antennas are the first N_d, *i.e.*, $\mathcal{N}^{(q)} = \{1, 2, \ldots, N_d\}$. During data transmission, in general, $\mathcal{N}^{(q)}$ includes the TAI of all granted transmissions intended to reach node q and the TAI of some other interfering transmissions.

We define $\boldsymbol{H}^{(q)}$ as the matrix containing the columns $\tilde{\boldsymbol{H}}^{(q)}_{\cdot,i}$, with $i \in \mathcal{N}^{(q)}$, and we define the N_d-size vector of data symbols belonging to $\mathcal{N}^{(q)}$ as $\boldsymbol{s}^{(q)} = [s'_1, s'_2, \ldots, s'_{N_d}]^T$. The variance of the entries of $\boldsymbol{s}^{(q)}$ is $\sigma^2_{s^{(q)}}(i) = \sigma^2_{s'}(i), \forall i$.

The transmitting antennas whose signals are not detected by node q are instead indicated as *external antennas* (EA) to q. According to $\mathcal{N}^{(q)}$, we assume that the TAI of EA are $N_d + 1, N_d + 2, \ldots, U$ and the corresponding channel matrix $\bar{\boldsymbol{H}}^{(q)}$ contains the columns $\tilde{\boldsymbol{H}}^{(q)}_{\cdot,i}$, with $i = N_d + 1, N_d + 2, \ldots, U$. The transmitted symbols of EA to q are denoted as $\bar{\boldsymbol{s}}^{(q)} = [s'_{N_d+1}, s'_{N_d+2}, \ldots, s'_U]^T$. With the definitions of IA and EA, we can rewrite the received signal as

$$\boldsymbol{r}^{(q)} = \boldsymbol{H}^{(q)}\boldsymbol{s}^{(q)} + \bar{\boldsymbol{H}}^{(q)}\bar{\boldsymbol{s}}^{(q)} + \boldsymbol{\nu}'^{(q)} . \tag{2.1}$$

The explicit modeling of a pure interference term is an important point in our ad hoc scenario. Indeed, terminals may be able to detect only a limited number of signals, or they may decide to neglect low-power interference and reduce processing, *e.g.*, for energy saving purposes.

In order to simplify notation we will omit in the following the index of the receive node $^{(q)}$ in all variables, since we will always refer to a single node.

2.3.2 BLAST receiver

In order to extract a sufficient statistics for detection, the receiving node multiplies the vector of the received samples by a matrix matched to the channel. By defining the $N_d \times N_d$ matrix $\boldsymbol{R} = \boldsymbol{H}^H\boldsymbol{H}$, where ^{H} is the Hermitian operator, the obtained vector is $\boldsymbol{z} = \boldsymbol{H}^H\boldsymbol{r} = \boldsymbol{R}\boldsymbol{s} + \boldsymbol{\nu} + \boldsymbol{i}_{\mathrm{EA}}$, where $\boldsymbol{\nu} = \boldsymbol{H}^H\boldsymbol{\nu}'$ and $\boldsymbol{i}_{\mathrm{EA}} = \boldsymbol{H}^H\bar{\boldsymbol{H}}\bar{\boldsymbol{s}}$ accounts for the interference due to EA. We recall that the receiver is not required to know the statistics of $\boldsymbol{i}_{\mathrm{EA}}$.

The BLAST receiver performs the detection of the streams in stages. At each stage the stream with the highest signal to noise plus interference ratio (SINR) is detected, and its contribution is removed from the vector $\boldsymbol{z}$ before the next stage [1,3]. The ordered TAI set is $\{k_1, k_2, \ldots, k_{N_d}\}$, which is a permutation of the integers $1, 2, \ldots, N_d$.

Let k_i be the TAI of the stream detected at the ith stage and $\boldsymbol{z}(i)$ the vector obtained from $\boldsymbol{z}$ after the removal of the contributions due to streams with TAI in $\mathcal{K}(i) = \{k_1, k_2, \ldots, k_{i-1}\}$ where we set $\mathcal{K}(1) = \varnothing$ and $\boldsymbol{z}(1) = \boldsymbol{z}$. The detection of stream k_i is performed by combining $\boldsymbol{z}(i)$ with the weighing vector $\boldsymbol{w}(i)$ to obtain the sample $\tilde{s}_{k_i} = \boldsymbol{w}(i)^T \boldsymbol{z}(i)$, which is applied to a threshold detector to provide the symbol estimate $\hat{b}_{k_i}$. The estimated symbol is multiplied by the standard deviation of the transmitted symbol to obtain $\hat{s}_{k_i} = \sigma_s(k_i)\hat{b}_{k_i}$. After detection, the contribution of stream k_i is removed from $\boldsymbol{z}(i)$ to obtain

$$\boldsymbol{z}(i+1) = \boldsymbol{z}(i) - \boldsymbol{R}_{\cdot,k_i}\, \hat{s}_{k_i}\,, \quad i = 1, 2, \ldots, N_d - 1\,, \tag{2.2}$$

where $\boldsymbol{R}_{\cdot,k_i}$ is the k_ith column of $\boldsymbol{R}$.

In the zero forcing (ZF) approach [1], the weighing vector aims at minimizing the interference, regardless of a possible noise enhancement. Let $\boldsymbol{R}(1) = \boldsymbol{R}$ and then compute $\boldsymbol{R}(i)$, $i = 2, 3, \ldots, N_d - 1$ by nulling the k_ith row and column of $\boldsymbol{R}(i-1)$. The weighing vector $\boldsymbol{w}(i)$ is the k_ith column of $\boldsymbol{R}^+(i)$ (the Moore–Penrose pseudoinverse of $\boldsymbol{R}(i)$ [28]), *i.e.*, its mth element is

$$w_m(i) = [\boldsymbol{R}^+(i)]_{k_i,m}, \tag{2.3}$$

$m = 1, 2, \ldots, N_d$.

Provided that the number of receive antennas is larger than the number of residual IA, $N_d - i$, and that $\boldsymbol{R}(i)$ is full rank, the given weighing vector completely cancels the interference due to streams $k_{i+1}, k_{i+2}, \ldots, k_{N_d}$. However, when $N_A < N_d - i$, $\boldsymbol{R}(i)$ has rank smaller than $N_d - i$, and using $\boldsymbol{R}^+(i)$ leaves some residual interference due to IA after weighing. Alternatively, a minimum mean square error (MMSE) criterion could be adopted for the choice of the weighing vector [29], which jointly considers noise and interference. In this part of the thesis we only consider the ZF approach, since we verified by simulation that MMSE does not bring any significant advantage and its derivation is entirely analogous to the ZF case.

Lastly, for real constellations, a receiver with improved performance has been derived in [26]. Accordingly, in the forthcoming analysis we shall consider the real part $\mathfrak{Re}\,[\boldsymbol{R}]$ instead of $\boldsymbol{R}$.

The performance of the considered architecture for various configurations of the number of transmitting and receiving nodes is shown in Fig. 2.1

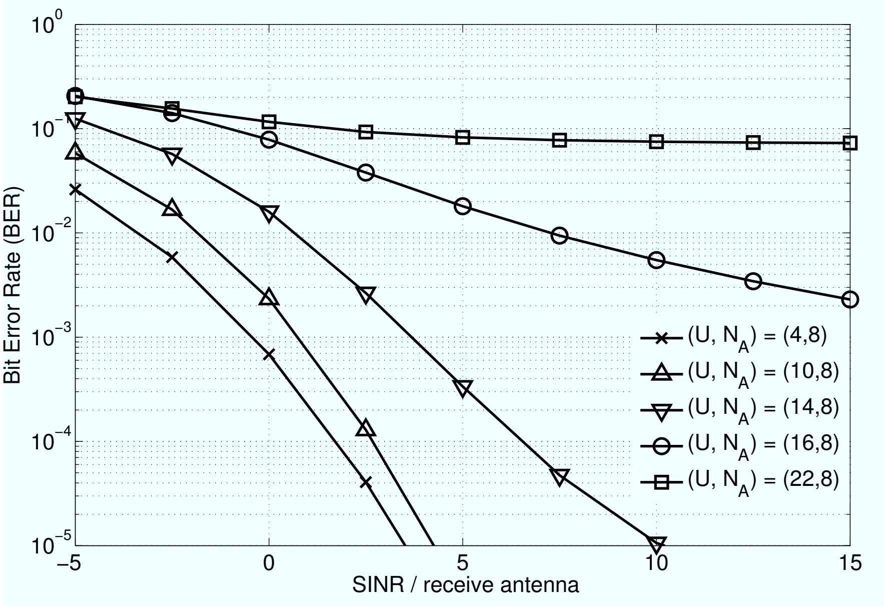


Figure 2.1. *BER performance of decision-feedback multiuser detection with $N_A = 8$ receiving antennas for different numbers of transmit antennas U as a function of the SINR perceived at the receiver.*

2.4 Cross–Layer Design of Medium Access Control for MIMO Ad Hoc Networks

2.4.1 Impact of PHY on MAC

The well known collision avoidance approach described in the 802.11 standard [8] makes use of control messages (RTS/CTS) in order to mitigate the hidden terminal problem, thus preventing collisions that would result in loss of data and waste of resources. In a MIMO ad hoc network, however, this is not always the best solution. Specifically, the receiver structure we presented in Section 2.3 is able, given some channel knowledge, to separate incoming PDUs which would then not result in a collision, but could instead be detected separately. This crucial channel knowledge at the receiver is obtained through training preambles preceding packet transmission.[3] The networking protocols may then choose how many and which channels to estimate, taking into account that the limited receiver capabilities allow locking onto at most N_S^{max} sequences simultaneously. While doing this, the protocols must be aware of the tradeoff existing between the amount of wanted data to detect and the interference protection granted to those data. In other words, trying to detect too many wanted data packets could leave limited resources for interference cancellation, leading to data loss. Note that, even with channel estimation and spatial demultiplexing, the MIMO receiver it-

[3] Estimating the channel usually requires to calculate correlation over known signals preambles, that are typically provided by PN sequences. For a more in-depth discussion on channel estimation, the interested reader is referred to [27,30,31].

self is still vulnerable to "hidden terminals" in some sense: if the receiver is not aware of interfering nodes nearby, it cannot estimate their channel and cancel them.

A properly designed MAC protocol can offer much help here. In particular, the concurrent channel access typically found in ad hoc networks can be exploited, instead of being suppressed. Collision avoidance schemes, such as 802.11, try to avoid concurrency by blocking the nodes that receive an RTS or a CTS. Instead of blocking, we want to encourage simultaneous transmissions. We also want to make the receivers aware of potential interferers, and to exploit the spatial demultiplexing capabilities of MIMO processing. To this aim, we start with an assessment of the receiver performance when receiving data PDUs and signaling packets. Even if not exhaustive, this study is indeed important for two reasons. First, it yields some insight on data transmission capabilities and, more specifically, on how many PDUs can be spatially multiplexed. Second, it allows to understand the probability that superimposed signaling messages are correctly received. This latter parameter is quite crucial, since if this probability is sufficiently high, signaling packets can be relied upon as a source of information on neighboring traffic and handshakes.

For this study, we place a node in the center of a circular area of given radius R to act as a receiver. The intended transmitter is moved from 40 to 140 m away from this receiver. This node sends data in blocks of 1000 bits per antenna (*e.g.*, 2000 bits spatially multiplexed through 2 antennas, 4000 bits through 4 antennas, and so on). The maximum power is constrained, and equally divided among the used antennas, since this is the best choice in the absence of channel state information at the transmitter. Moreover, we randomly place inside the area some further (interfering) data senders, which always transmit 1000 bits of data at full power through one antenna. Those nodes falling below a threshold distance $d^* < R$ are considered IAs, thus their contribution can be detected and canceled (provided that the limit on the maximum number of channel estimations is not exceeded, after which these nodes become EAs). Conversely, the nodes falling beyond d^* are always treated as unknown interferers (EAs). The reasoning here is that a node could either have a limited knowledge of its neighborhood, or not wish to detect all incoming signals, but only those with a sufficiently high received power, in order to guarantee detection performance. We have set R and d^* so that the probability that a signal is detected is 50%.

The results of this test are given in Fig. 2.2, using 8 interferers in addition to the intended transmitter. The curves show that there exists a tight relationship between the number of used antennas (thus, bit rate) and the average received power, thus the maximum coverage distance affordable. For example, with a 90% minimum success ratio objective, a transmitter could reach 70 m, 90 m and 110 m, using 8, 4 and 2 antennas respectively. This maximum number of antennas as related to the distance of a node is called the "class" of that node, and is especially useful in a multiple-receiver context, where the transmitter could send data to many neighbors at once. In this case, the class of a neighbor represents the maximum number of antennas allowed when transmitting to a set of receivers including that neighbor.[4] To

[4] This choice of setting classes based on distance is made for the sake of a clearer explanation, but is by no

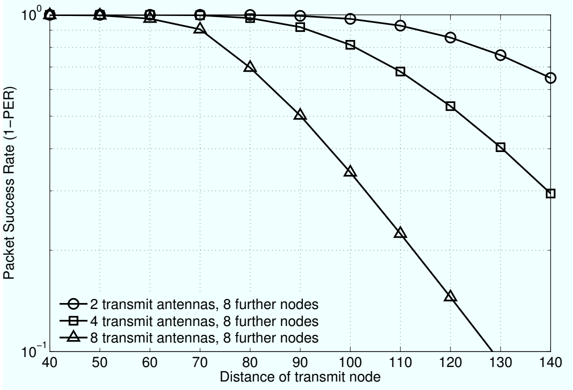


Figure 2.2. *Probability of capturing a data packet in the presence of interfering traffic versus the distance of the transmitter, for varying number of antennas used by the transmitter.*

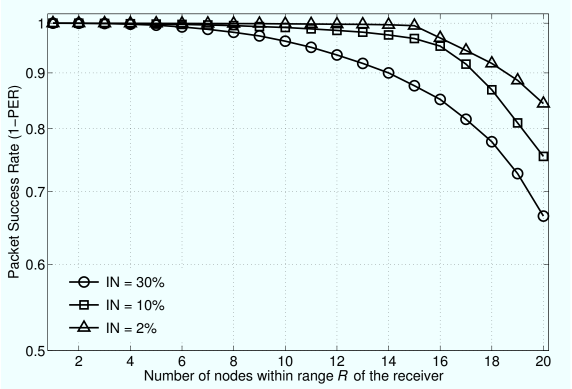


Figure 2.3. *Probability of capturing a signaling packet versus the number of nodes within range R of the receiver, for varying number of interfering nodes over the total number of transmit nodes.*

encourage parallelism, RTS/CTS messages do not block transmissions in our scheme, but rather are used for traffic load estimation. Since signaling packets are shorter and transmitted with a single antenna at full power, we expect them to be detectable in large quantities without significant errors. To verify this intuition, we have considered a similar scenario as before, with 1 to 20 nodes transmitting simultaneously 200 bits long RTSs to a receiver placed at the center of a circular area. Again, all nodes beyond d^* are considered unknown interferers. Besides the previously explained reason, here d^* also functions as a measure of a node's knowledge of its neighboring network activity. We simulate such higher tracking capabilities through a greater d^*, and vary it such that the average number of interfering nodes (IN) over all nodes is 30%, 10% and, as a limit case, 2%. Recall that these nodes are all EAs, and thus increase the interference level for the reception of wanted packets. Conversely, the nodes within d^* are EAs only if their transmissions exceed the estimation capabilities at the receiver. Fig. 2.3 summarizes signaling packet capture performance. With the same settings as in Fig. 2.2 (IN $= 30\%$), there is still a fairly high probability of detecting a good percentage of the signaling packets, translating into 13 to 15 correct detections with $N_A = 8$ antennas, even if more packets are sent. This value improves if the node can afford to increase its neighborhood knowledge (IN $= 10\%$, 2%). The insight gained here is that relying on the exchange of signaling packets prior to data transmission is in fact possible, because it is highly likely that a substantial fraction of these packets is received correctly.

2.4.2 Cross–layer MAC Design

To gather most of these advantages, we resort to a framed communication structure, with four phases. For this scheme to work correctly, all nodes have to share the same frame

means necessary for correct protocol operation. For example, one could choose the class based on average (estimated) link error probability, link utilization, and so forth.

synchronization. These phases are designed according to the standard sequence of messages in a collision avoidance mechanism, and are summarized as follows.

RTS phase—In this phase, all senders look into their backlog queue, and if it is not empty they compose transmission requests and pack them into a single RTS message. Each packet in the queue is split into multiple PDUs of fixed length, such that each PDU can be transmitted through one antenna. For this reason, any request has to specify the number of PDUs to be sent simultaneously, in addition to the intended destination node. How to associate a destination node with a suitable number of transmit antennas is an *RTS policy*, and depends on the degree of spatial multiplexing sought, as well as the local traffic intensity, thus the queue level of the sender. Any RTS may contain several such requests. Moreover, an RTS is always sent with one antenna and at full power.

CTS phase—During the RTS phase, all nodes that were not transmitters themselves receive multiple simultaneous RTSs, and apply the reception algorithm of Section 2.3 to separate and decode them. In the CTS phase, when responding to the correctly received RTSs, nodes have to account for the need to both receive intended traffic (thus increasing throughput) and protect it from interfering PDUs (thus improving reliability). The constraint in this tradeoff is the maximum number of trackable channels, *i.e.*, the maximum number of training sequences a node can lock onto. We name a *CTS policy* the way the former is traded off for the latter, *e.g.*, controlling the number of allowed senders and/or the number of allowed antennas. Since this is a design decision, we defer the description of the compared CTS policies to the following Section. CTSs are also sent out using one antenna and at full power.

DATA phase—All transmitters receive superimposed CTSs and, after BLAST detection, they follow CTS indications and send their PDUs. Each PDU has a fixed predefined length and is transmitted through one antenna, but a node can send multiple PDUs simultaneously, possibly to different receivers.

ACK phase—After detection, all receivers evaluate which PDUs have been correctly received, compose a cumulative PDU–wise ACK, and send it back to the transmitters. After this last phase, the data handshake exchange is complete, the current frame ends and the next is started. Note that this corresponds to the implementation of a Selective Repeat Automatic Repeat reQuest (SR–ARQ) protocol, where PDUs are individually acknowledged and, if necessary, retransmitted.

Before going more deeply into CTS policy definition, we remark that a random backoff is needed for nodes that do not receive a CTS, as otherwise persistent attempts may lead the system into deadlock. Here, we make use of a standard exponential backoff. Accordingly, before transmitting, nodes wait for a random number of frames, uniformly distributed in the interval $[1, BW(i)]$, where i tracks the current attempt, and $BW(i) = 2^{i-1}W$, with W a fixed backoff window parameter. An accurate study of the effects of different backoff strategies can be found in [32].

Before proceeding, we highlight that we only require that nodes be frame–synchronous, even if for simplifying the system description, we have referred to [26] in Section 2.3, where synchronization is assumed at the symbol level. In fact, instead of operating on a per–symbol basis, the receiver can first detect one whole PDU and then cancel it, detect and cancel the second PDU and so forth, until the last one is detected. Frame synchronization is not a strong requirement, and can be easily implemented with current technology.

2.5 RTS and CTS Policies

The last details we need to specify about our MAC protocol are RTS and CTS policies, which are especially important in this context, since efficient data exchange requires that the receivers' detection capabilities are not exceeded, and that sufficient knowledge of the neighborhood is available. Before dealing with MIMO-specific policies in Sections 2.5.1 and 2.5.2, we introduce here a simpler baseline protocol that we will use later for comparison. The definition of this protocol is necessary, since the approaches described in Section 2.2 cannot be directly compared to our solution, because of either the absence of a specific MAC scheme [19], the optimization of MAC around some fixed PHY parameters such as the number of antennas [20], the diverse issues related to different modulation and signaling schemes [21], the attention devoted to achieving full diversity instead of full parallelism [22], or the idealized assumptions about a MIMO PHY level and MAC signaling [25].

Our baseline, instead, is meant as an example of how a layered networking solution would behave when set up on top of a SM-capable MIMO PHY level. Furthermore, it is directly comparable with our policies, as it takes into account the PHY used (unlike [25], that focuses on link capacity) and is sufficiently general not to depend on the number of antennas per node (unlike [20]). Our baseline works as follows. When a node has a packet to transmit, it senses the channel, gaining access if it finds it free. In order to obtain an optimistic upper bound on the performance of this protocol, we assume that the transmitter selection is "ideal", in the sense that one node among the RTS senders is chosen to transmit, whereas the others back off. Also, note that the random choice of a node does not yield significant drawbacks, because the nodes of the network simulated in Section 2.6 are within coverage of one another (see the same Section for details about this choice) and therefore any transmission would silence all other handshakes. When a node is granted access, it sends an RTS and waits for a CTS from the recipient. To be consistent with the following MIMO transmission policies, data packets are divided in PDUs, each 1000 bits long. Now, the best transmission enhancement obtainable within this protocol is to increase the raw bit rate as much as possible. To this aim, PDUs are split in chunks, one per each available antenna, and transmitted in parallel through all antennas. If more than one PDU belongs to the same packet, all PDUs are transmitted sequentially in the same way. For example, if a packet is formed of a number of 1000-bits PDUs and $N_A = 8$, each antenna will send one 125-bits chunk per PDU. Before returning to the idle state or performing another transmission attempt, the node waits for an ACK from the receiver, reporting which PDUs were detected correctly. In case of errors, only the erroneous PDUs are retransmitted.

This baseline is a reasonably simple protocol, yet it makes use of MIMO capabilities and maintains other features similar to 802.11, such as carrier sense and contention-based channel access, with no cancellation of interference coming from other nodes. Basically, the baseline protocol is a carrier-sense multiple access scheme with collision avoidance, just using a more powerful MIMO PHY layer. Results based on this scheme will show that a straightforward use of a layered solution on top of the more powerful MIMO PHY is a significantly suboptimal choice.

2.5.1 RTS policy

Let the set of neighbors of a given node s be denoted as $\mathcal{V} = \{v_1, v_2, \ldots\}$. Also let a_{sv_j} be the class of v_j, $j = 1, 2, \ldots$, which indicates the maximum number of antennas that s can use when transmitting to any set of nodes that includes v_j. Since we wish to encourage spatial multiplexing, we restrict a_{sv_j} to be either $\alpha_1 = 2$, $\alpha_2 = 4$, or $\alpha_3 = 8$. For clarity, we refer to Fig. 2.2, and set the class of the neighbors according to three threshold distances δ_1, δ_2, δ_3 corresponding to the maximum reach achievable with α_1, α_2, α_3 transmit antennas, respectively. Then, s sets $a_{sv_j} = \alpha_m$ if and only if $\delta_{m-1} < d(s, v_j) \leq \delta_m$, where $d(x, y)$ is the distance between node x and node y, and $\delta_0 = 0$. Note that distances are directly related to the average received SNR value, so that an objective function can be chosen for setting threshold distances based on either metric.[5]

Let us focus on node n. The algorithm begins with step $i = 1$. If the node queue is non-empty, a request is created as follows. The node sets $k_1 = 1$ and reads the k_1th packet's destination, d_{k_1}, and the number of packet PDUs still unsent, p_{k_1}. Then, it compares p_{k_1} with k_1's class, $a_{nd_{k_1}}$. If $p_{k_1} \geq a_{nd_{k_1}}$, the unsent PDUs saturate the node class, hence forbidding any further spatial multiplexing. In this case, the request pair $(d_{k_1}, a_{nd_{k_1}})$ is inserted in the RTS, and the RTS is sent right away.

Conversely, if $p_{k_1} < a_{nd_{k_1}}$, the pair (d_{k_1}, p_{k_1}) is put in the RTS. Node n keeps memory of the queue indices of all packets selected for transmission, maintaining them in set $\mathcal{S}_i$, where i is the step index. It also accumulates in the variable $A(i)$ the total number of antennas allotted until step i. At step 1, $\mathcal{S}_1 = \{k_1\}$, and by calling $M(1) = \min\{a_{nd_{k_1}}, p_{k_1}\}$ the number of antennas allocated to packet k_1, we have $A(1) = M(1)$. Since $p_{k_1} < a_{nd_{k_1}}$, node d_{k_1} could potentially sustain a transmission with $a_{nd_{k_1}} - p_{k_1}$ further antennas (in the absence of interference). In this case there is still room for sending one or more further PDUs taken from other packets. Therefore, the node proceeds to step $i = 2$ and scans its queue, until it finds a packet k_2 whose destination's class matches the condition $a_{nd_{k_2}} > A(1)$. This means that the destination d_{k_2} can stand the transmission of the $A(1)$ PDUs already allocated, plus

[5] Later, we will use a 90% target success rate, and accordingly set $\delta_1 = 70\,\text{m}$, $\delta_2 = 90\,\text{m}$ and $\delta_3 = 110\,\text{m}$ as per the results of Fig. 2.2.

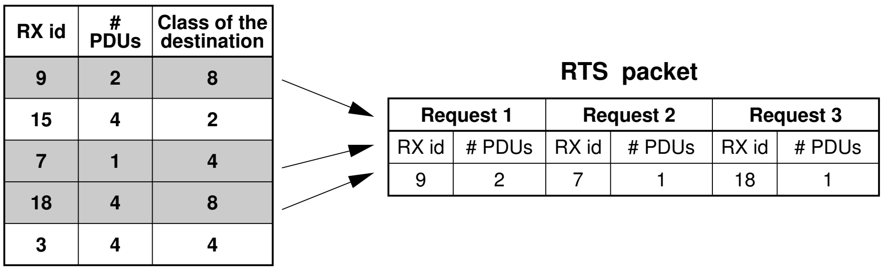
one or more of its own. The sender then sets $\mathcal{S}_2 = \mathcal{S}_1 \cup \{k_2\}$, calculates the number of PDUs allotted to packet k_2 as $M(2) = \min\big\{\min\{a_{nd_{k_1}}, a_{nd_{k_2}}\} - A(1), p_{k_2}\big\}$, so as not to violate the antenna constraints $a_{nd_{k_1}}$ and $a_{nd_{k_2}}$ and taking into account that $A(1)$ antennas have already been allotted. Then, it inserts in the RTS the pair $(d_{k_2}, M(2))$, and finally updates $A(2) = A(1) + M(2)$. If there is still room for transmission without violating antenna constraints, *i.e.*, if $\min_{j\in\mathcal{S}_2}\{a_{nd_j}\} > A(2)$, the node proceeds to step $i = 3$, searching again for a packet k_3 in its queue whose class $a_{nd_{k_3}} > A(2)$, and so on. In general, at step i, the node explores the queue for a packet k_i with a feasible class $a_{nd_{k_i}} > A(i-1)$. Then $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{k_i\}$, $M(i) = \min\big\{\min_{j\in\mathcal{S}_i}\{a_{nd_j}\} - A(i-1), p_{k_i}\big\}$, and $A(i) = A(i-1) + M(i)$. The request $\big(d_{k_i}, M(i)\big)$ is put in the RTS. The algorithm then proceeds to step $i+1$ if and only if $\min_{j\in\mathcal{S}_i}\{a_{nd_j}\} > A(i)$ and a packet such that $a_{nd_{k_{i+1}}} > A(i)$ is found in the queue. As an example, consider Figure 2.4. The RTS is formed by first allotting the 2 PDUs required by node 9, whose class is 8. Node 15 cannot be accommodated, because its class is 2, and two antennas have been already allotted to send two PDUs to node 9. However, node 9 can sustain six more antenna transmissions, which allows to accommodate one PDU for node 7. Notice that node 7's class (4) now represents the most restrictive constraint, and that three PDUs have already been requested so far. This allows a third request for one PDU to node 18, which completes the RTS construction. Note that while this excludes a packet for the low class node 15 from being served, it will be the first one to be considered for transmission in the next frame.

2.5.2 CTS policies

At first, the node sorts all requests contained in every correctly decoded RTS in order of decreasing received power, and divides them in two subsets, namely $\mathcal{W}$ and $\mathcal{U}$, respectively standing for *wanted* and *unwanted*. The first set contains all requests directed to the node, the second set all other requests. Recall that if a request by node s_k implies the transmission of, say, r_k PDUs, the receiver has to account for channel estimation resources that will be needed for all PDU transmissions. Since the maximum number of simultaneous PDUs per receive antenna is limited to N_S^{max}, each time a transmission is granted the number of available tracking resources is decreased by r_k. Therefore, for each request considered, the receiver inserts in the CTS the pair $(s_k, \bar{r}_k)$, where $\bar{r}_k = \min\{r_k, N_S^{max} - \sum_{j=1}^{k-1} r_j\}$, until there are no more available tracking resources. Grants are given according to one of the following policies, with the understanding that no more than N_S^{max} PDUs can be granted.

do Not Follow Traffic (NFT)—In this case, the node grants the requests in $\mathcal{W}$ until either they are all granted or all available channel estimation resources are used, and does not consider $\mathcal{U}$ at all.

Follow Traffic (FT)—In FT, the node always grants the first (highest-power) request in $\mathcal{W}$ and then considers all other requests in $\mathcal{W} \cup \mathcal{U}$, re-ordered by decreasing received power. At step k, if the processed request belongs to $\mathcal{U}$, no grants are given in the CTS, but the number of estimation resources available is decreased according to $\bar{r}_k$. An example is given



RX id	# PDUs	Class of the destination
9	2	8
15	4	2
7	1	4
18	4	8
3	4	4

RTS packet

Request 1		Request 2		Request 3	
RX id	# PDUs	RX id	# PDUs	RX id	# PDUs
9	2	7	1	18	1

Figure 2.4. *Example of application of the RTS policy. No request for nodes 15 and 3 is included in the RTS, because the maximum number of antennas allowed toward these nodes is too small. In addition, allowing transmission to node 3 or allowing a request of more than one PDU to node 18 would overload the reception capability of node 7.*

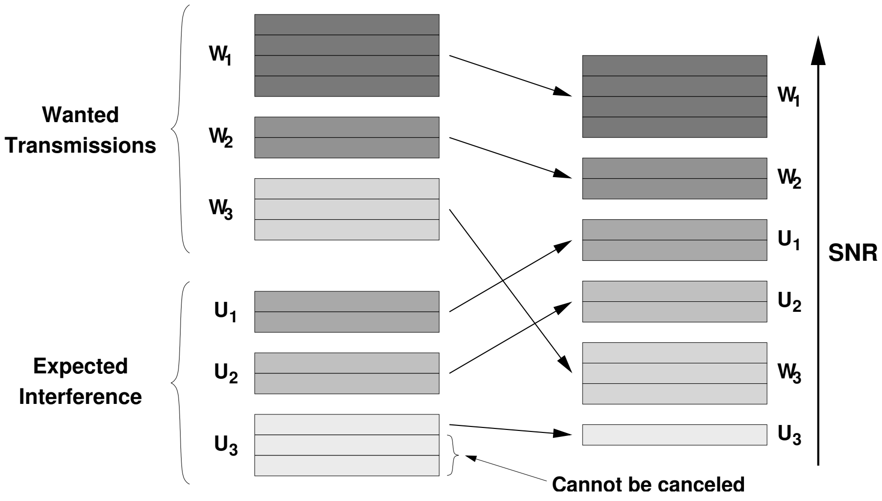


Figure 2.5. *Example of application of the FT policy. Darker shades of gray represent higher receive SNRs. Some of the unwanted PDUs by U_3 cannot be canceled due to limited channel estimation capabilities, and are left as unknown interference.*

in Figure 2.5, for 3 wanted and 3 unwanted traffic requests, each with a different number of associated PDUs. By assigning the channel estimation resources to the requesting nodes in order of decreasing SNR, the receiver can accommodate all wanted requests. Moreover, it can detect (and cancel) the interference from U_1 and U_2, whereas only one PDU from U_3 can be canceled, due to lack of further resources. This policy strives to guarantee some throughput through the allowance of one transmission in $\mathcal{W}$ but prioritizes protection from strong interference by merging $\mathcal{W}$ and $\mathcal{U}$ when choosing which channels to track. In order to show that these are both necessary, we also consider the two following modifications of FT.

Partially Follow Traffic (PFT)—With PFT, a node gives priority to wanted transmissions, processing first all requests in $\mathcal{W}$. If there is any tracking resource left, it then begins to consider requests in $\mathcal{U}$ until all resources are exhausted, enabling the cancellation of some

Transmitter-side MAC Operations

```
// Initialize the step index $i$, the number of allotted antennas $A$, the set of
// receivers $\mathcal{S}$, and the number of failures $N_{fail}$
$i = 1$; $A(0) = 0$; $\mathcal{S}_0 = \varnothing$; $N_{fail} = 0$
// RTS phase: add users until class constraints are violated
while $\min_{j \in \mathcal{S}_i} a_{nd_j} > A(i-1)$ do
  // Is there a packet in the queue that complies with the current
  // constraints?
  if $\exists$ a packet $k_i$ s.t. $a_{nd_{k_i}} > A(i-1)$ then
    // Add user as receiver
    $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{k_i\}$
    // Determine number of PDUs to send that does not violate any
    // current class constraint
    $M(i) = \min\{\min_{j \in \mathcal{S}_i}\{a_{nd_{k_j}}\} - A(i-1), p_{k_i}\}$
    $A(i) = A(i-1) + M(i)$
    Insert request $(d_{k_i}, M(i))$ in RTS
  end if
end while
Send RTS
// Data phase: check CTS
if One or more CTS received then
  Send data PDUs according to CTSs
  $N_{fail} = 1$
else
  Backoff for $b$ frames, $b$ uniformly distributed in $[1, W \cdot 2^{N_{fail}-1}]$
  $N_{fail} = N_{fail} + 1$
end if
if ACK received then
  Mark all ACK'ed PDUs
  Remove from the queue all packets whose PDUs have been all ACK'ed
end if
```

Figure 2.6. *Pseudo-code description of transmitter-side MAC operations.*

neighboring interference. This variant privileges wanted traffic over protection against interference.

FT Without Interference Cancellation (FT–WIC)—This policy operates as FT, but does not perform cancellation of interfering requests in $\mathcal{U}$. This implies that the only means of protection given to data is refraining from transmission if there are too many powerful interferers. This scheme is therefore expected to have poor performance and is considered here to stress the importance of interference cancellation.

For a better understanding of transmitter- and receiver-side operations, we report in Figs. 2.6 and 2.7 a pseudo-code description of our MAC algorithm, where FT has been chosen as the specific CTS policy. Observe that all described policies are cross–layer. On the one hand, they manage network access by selecting which incoming PDUs to decode. They perform this control by accounting for the physical layer, which processes PDUs in order of av-

Receiver-side MAC Operations

```
// Initialize number of trackable training sequences, $N_s$
$N_s = N_s^{max}$
// CTS phase: apply CTS policy
if One or more RTSs received then
    Create ordered sets $\mathcal{W}$ and $\mathcal{U}$
    Let $\mathcal{I}_{\mathcal{W}}$ be the ordered set with the indices of the packets in $\mathcal{W}$
    Let $\mathcal{I}_{\mathcal{U}}$ be the ordered set with the indices of the packets in $\mathcal{U}$
    // Grant at least one wanted request
    $i = \mathcal{I}_{\mathcal{W}}(1)$
    Read source $s_i$ and number of PDUs $p_i$ for the packet with index $i$
    Insert grant $(d_i, p_i)$ in CTS
    $N_s = N_s - p_i$
    $\mathcal{I}_{\mathcal{W}} = \mathcal{I}_{\mathcal{W}} \setminus \{i\}$
    // Manage other requests in order of decreasing received power
    while $N_s > 0 \wedge (\mathcal{I}_{\mathcal{W}} \neq \varnothing \vee \mathcal{I}_{\mathcal{U}} \neq \varnothing)$ do
        Let $i$ be the request with greatest power between $\mathcal{I}_{\mathcal{W}}(1)$ and $\mathcal{I}_{\mathcal{U}}(1)$
        $N = \min\{p_i, N_s\}$
        $N_s = N_s - N$
        if $i \in \mathcal{I}_{\mathcal{W}}$ then
            Insert grant $(d_i, p_i)$ in the CTS
        end if
    end while
end if
Send CTS
// Data phase: receive data PDUs
if Data PDUs received then
    De-multiplex PDUs and extract wanted ones
    Send ACK for correctly received PDUs belonging to requests in $\mathcal{W}$
end if
```

Figure 2.7. *Pseudo-code description of receiver-side MAC operations. The chosen CTS policy is FT.*

erage received power. On the other hand, they force the multiuser detector to decode subsets of PDUs that correspond to different operating points on the throughput–reliability tradeoff. All these decisions are taken based on information about per–stream powers, a parameter provided by PHY which is simple, as suggested in [16], but crucial. Note also that i) CTS policies are the only way to reduce data traffic in this kind of networks, since RTSs/CTSs are not used for channel reservation, but rather as an indication of intention/clearance to transmit, and ii) both RTS and CTS policies favor the creation of multiple point–to–point links, all potentially making use of SM. This is made possible by inserting multiple requests (grants) in the RTS (CTS), each composed of multiple PDUs. An accurate comparison of the CTS policies is carried out in the following Section.

Notice that the policies proposed here are not bound to a MIMO PHY but, on the contrary, they are suitable for use with any decision-feedback multiuser detection-capable PHY level. In other words, our policies can operate on top of any PHY that successively detects multiple signals, and cancels their contribution from the received signal prior to the following detections. We chose V–BLAST as one such PHY, since it is a good representative and has recently received a lot of attention [7].

2.6 CTS Policies Comparison

2.6.1 Simulation Setup

In order to evaluate our MAC schemes specifically designed for use with a decision feedback multiuser detector receiver, as well as the related RTS/CTS policies, we deploy 25 nodes with 8 antennas each in a square grid topology with 5×5 nodes and nearest neighbors $25\ \mathrm{m}$ apart. All nodes are static, and we assume that the frame synchronization assumption holds throughout the simulation.Traffic is generated according to a Poisson process of rate λ packets per second per node. Each generated packet is made of k $1000\,\mathrm{bits}$-long PDUs, with k randomly chosen in the set $\{1, 2, 3, 4\}$. Unsent packets are buffered. We test this specific configuration because nodes are all within coverage range of each other: this is a demanding scenario in terms of interference, required resources, and efficient protocol design. Transmissions follow the MAC protocol described in Section 2.4.2 and the policies of Section 2.5. All other relevant simulation parameters are given in Table 2.1. We have built a fully detailed MATLAB simulator that accurately reproduces the multiuser detection algorithm at the symbol level, on top of which we stack the framed MAC described in Section 2.4.2 and either the baseline or one of the MIMO-specific RTS/CTS policies.

2.6.2 Numerical Results

In Fig. 2.8 we compare all CTS policies in terms of aggregate network throughput as a function of traffic. Throughput is measured here in $\mathrm{Mbit/s}$ in the whole network. NFT

Description	**Value**
Number of nodes	25
Antennas per node, N_A	8
Operating band	5.8 GHz ISM
Data rate per antenna	7.5 Mbps
Digital modulation	BPSK
Type of traffic	Poisson, constant rate λ
Backoff window parameter, W	1
Maximum backoff window	32 frames
Signaling packet length	200 bits
PDU length	1000 bits
PDUs per packet	k, rand. chosen in $\{1, 2, 3, 4\}$
Queue buffer capacity	120 PDUs
Packet timeout	2500 frames (0.53 s)
No. of trackable sequences, N_S^{max}	32
$\{\delta_1, \delta_2, \delta_3\}$	$\{70, 90, 110\}$

Table 2.1. *Relevant Simulation Parameters*

shows very poor performance for all traffic values, for two reasons: i) it allows the transmission of all requested PDUs, regardless of whether the receivers can separate them, and ii) it does not cancel any interferer. PFT performs slightly better since, while still granting every requested PDU, it incorporates a mechanism that exploits unused estimation resources for canceling the strongest interfering PDUs (recall that every policy always considers decreasing received powers when selecting what to grant or to cancel). Yet, PFT cannot cope with excessive traffic load. In fact, beginning from λ between 700 and 800, the amount of requested traffic leaves less room for cancellation of unwanted signals, and causes a throughput decrease. The key reason why FT performs better than PFT is that FT allots estimation resources to both wanted and interfering PDUs, while still ensuring that at least one wanted PDU is granted. In the worst case, under exceedingly high traffic, 1 wanted PDU (the one with highest power) would be protected against the $N_S^{max} - 1$ highest-power interferers, in an attempt to let some wanted data get through. The net effect is to activate more frequently short-distance links that can sustain more SM, as will be shown in Fig. 2.10. The importance of interference protection is well highlighted by the FT-WIC policy results. From Fig. 2.8, we recognize the same trend experienced by NFT, just shifted up to some extent. This shift is explained by the FT–like behavior, as FT-WIC limits traffic in the presence of interfering PDUs. Nonetheless, FT-WIC still exhibits poor performance, because it lacks the most important interference cancellation feature. Note that the baseline transmission policy performs poorly as well, because it does not make full use of the MIMO capabilities, resulting in very low throughput. We highlight that both NFT and FT-WIC experience a slight linear increase at very low traffic, before reaching a saturation throughput value that remains then constant at

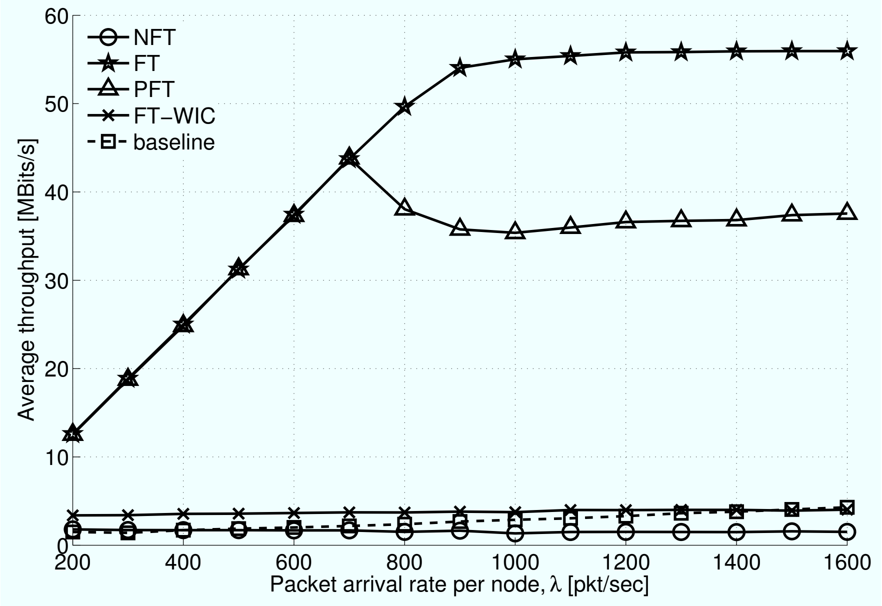


Figure 2.8. *Throughput for all CTS policies versus traffic.*

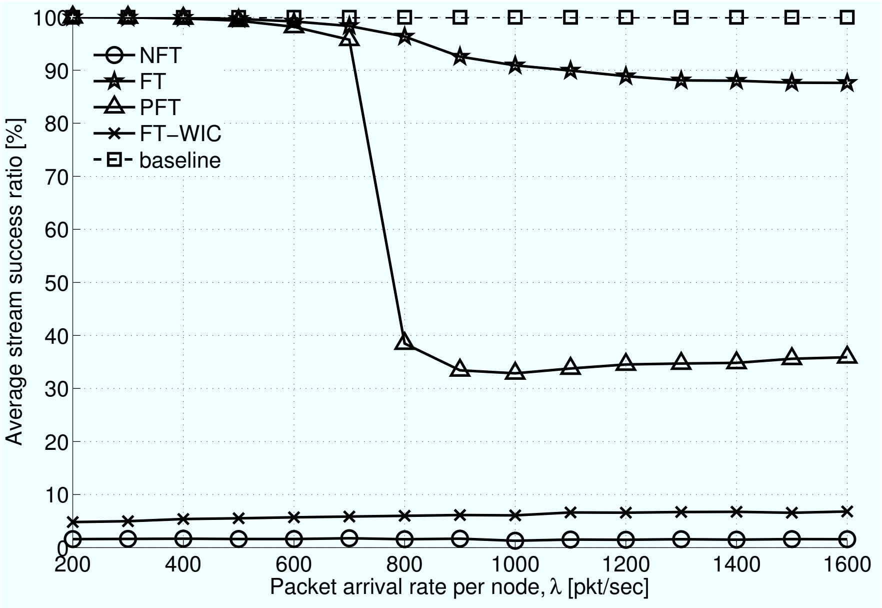


Figure 2.9. *Transmission success ratio of a PDU for all CTS policies versus traffic. Notice the more effective interference protection capabilities of FT, that allow a good success ratio even at high traffic.*

higher traffic. The initial increase is not shown here both because it is an expected behavior, and because we wish to focus on more specific policies such as PFT and FT. Moreover, it is expected that the throughput does not fall to zero, since some of the signals are eventually transmitted to nearby nodes, where the SNR allows some non-zero probability of correct reception.

As a side remark, the simulations show that the decision feedback multiuser detector

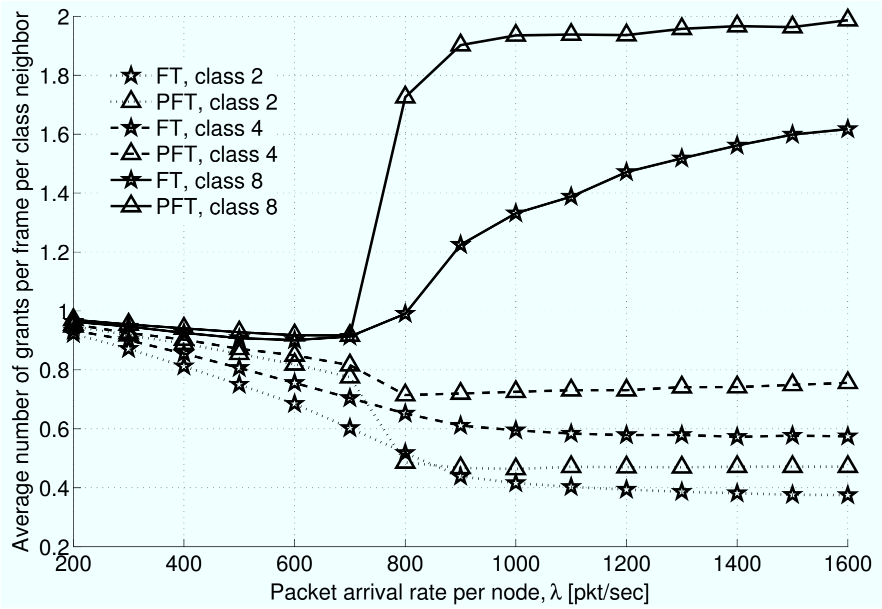


Figure 2.10. *Number of grants given to neighbors with different reception capabilities per frame versus traffic. Only the FT and PFT policies are displayed.*

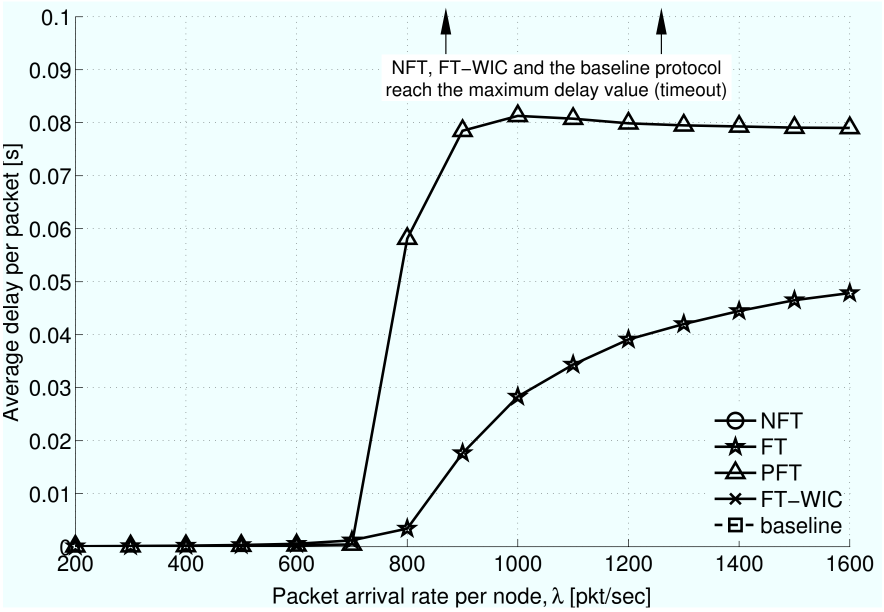


Figure 2.11. *Delay before a correct packet transmission (including queueing delay) versus traffic. Some curves are not displayed to focus on the more interesting comparison between FT and PFT.*

of Section 2.3, with FT, is able to support up to 12 successful PDUs per frame on average, which is larger than the maximum number of antennas per node, *i.e.*, 8, even in a fully connected network. This is a very interesting result: it substantiates the need for both a well-designed physical layer and a management protocol, and shows that the number of terminal antennas is a soft limit in MIMO ad hoc networks, if efficient RTS/CTS policies favoring the effective rejection of multiple access interference are provided. Fig. 2.9 shows the average ratio of successfully received to sent PDUs, and basically confirms the previous statements. FT achieves the best results and still almost ensures a 90% probability of correct detection at the highest traffic. On the contrary, NFT and FT-WIC incur a very low probability of detection success, and PFT stands in between, its chances being smaller than 40% at high λ. Conversely, the success ratio of the baseline protocol is near 100% as expected, since very few transmissions take place due to the collision avoidance mechanism.

To corroborate the claim that the use of FT at high traffic turns into a more likely activation of short links, we depict in Fig. 2.10 the number of grants given to each neighbor depending on the maximum number of antennas allowed for use with that neighbor (shortly referred here as its "class"). We only show PFT and FT, which achieve the most significant results, since they experience much less congestion than NFT and FT-WIC. We observe that, after starting from nearly one grant per node per class, both FT and PFT incur a progressively stronger decrease in the number of transmissions allowed toward neighbors in classes 2 and 4. On the other hand, transmissions toward class 8 neighbors increase more steeply than the others decrease. Such a behavior can be explained by observing that, in FT, it is highly likely that the request with strongest power comes from a close class 8 node and that other resources are dedicated to dealing with interference. The number of class 8 grants increases also for PFT, but this is only a consequence of the greater flexibility given by class 8 nodes, that can afford higher SM and thus allow the receiver to give more grants (see the RTS policy in Section 2.5.1).

The results described before are also confirmed by Figs. 2.11 and 2.12, which show the average packet delay in seconds and the average queue length, respectively. Consistently with previous results, we observe that only PFT and FT provide a limited delay, even for higher traffic values. More specifically, PFT reaches a saturation queueing plus transmission delay of approximately $0.08\,\mathrm{s}$, corresponding to 370 frames being necessary for a packet to reach the head of the queue and being correctly transmitted. For the same traffic values, the higher throughput achieved by FT is still capable of keeping the network uncongested, explaining the smoother increase in delay. Similar considerations apply to the behavior of the queue length as a function of traffic. In this case, the lower PFT throughput does not allow sufficient packet delivery capabilities, hence the node buffers are filled at $\lambda \geq 800$. In FT, instead, a higher amount of data gets through, resulting in a shorter queue length. All other policies, including the baseline protocol, perform much worse, as their average delay is close to the upper bound imposed by the value of the timeout.

Overall, the presented results show that the effective cross–layer design that led to FT achieves satisfactory performance, as it allows high throughput and success ratio, hence limited delay and backlog. The results also highlight the difference between FT-WIC and FT, thus the importance of interference cancellation when protecting wanted data, especially in a context where simultaneous channel access is encouraged in order to exploit SM-capable receivers. Finally, we remark that considering received requests in order of decreasing re-

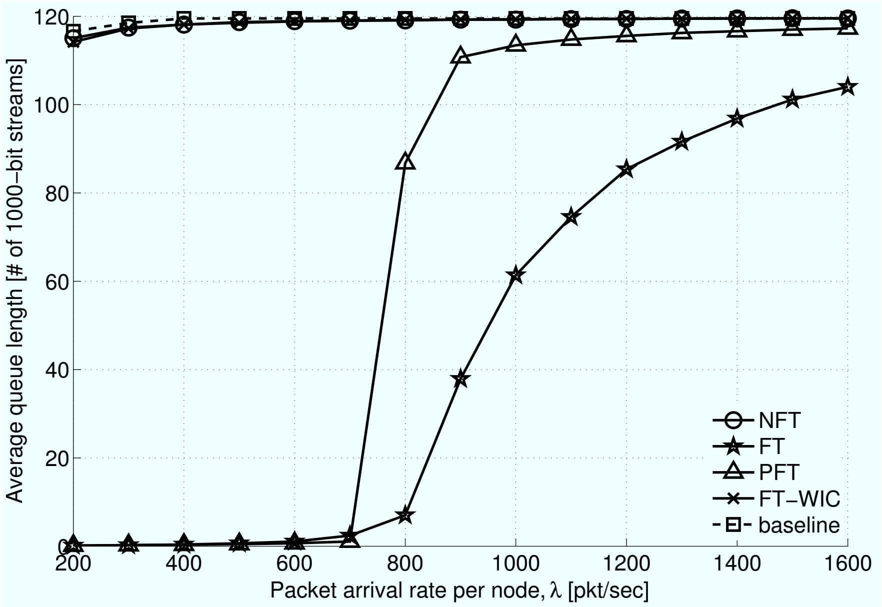


Figure 2.12. *Queue length for all CTS policies versus traffic.*

ceived power tends to favor shorter links (with greater SINR) at high traffic. This may have an impact on routing, as under heavy traffic it may be more convenient to set up longer paths with multiple, more robust hops. A more detail evaluation of this effect involves multihop topologies and routing issues, and is left for future study.

2.7 Receiver Performance Approximation

In this section we aim at evaluating the BER of BLAST given the channel matrix $\tilde{\boldsymbol{H}}$. The performance of a SM system is affected by a) imperfect IA cancellation due to detection errors, b) EA interference, c) residual interference due to the pseudo-inverse of $\boldsymbol{R}(i)$, and d) noise. The impact of imperfect cancellation on the detection of forthcoming streams is modeled in this paper with two approaches. In the first approach, denoted *Gaussian technique* and described in Section 2.7.1, the interference is considered as additional Gaussian noise. In the second approach of Section 2.7.2, denoted *enumeration technique*, we exhaustively enumerate all the various configurations of detection errors and compute the conditional BER for each configuration. Although the enumeration technique is more accurate than the Gaussian technique, the number of configurations to be explored increases exponentially with the number of IA, with a consequent increase in computation time. Hence, in Section 2.7.3 we derive a suboptimal technique, denoted *pruned tree technique*, where only the most relevant error configurations are explored.

In all cases, EA interference is approximated as Gaussian noise, with autocorrelation matrix

$$\boldsymbol{R}_{\mathrm{EA}} = \mathrm{E}[\boldsymbol{i}_{\mathrm{EA}}\boldsymbol{i}_{\mathrm{EA}}^H] = \boldsymbol{H}^H\bar{\boldsymbol{H}}\bar{\boldsymbol{\Sigma}}[\boldsymbol{H}^H\bar{\boldsymbol{H}}]^H, \tag{2.4}$$

where $\bar{\boldsymbol{\Sigma}}$ is a diagonal matrix with entries $\bar{\boldsymbol{\Sigma}}_{\ell,\ell} = \sigma_{s'}^2(N_d + \ell)$, $\ell = 1, 2, \ldots, U - N_d$. Then the power of EA interference after weighing can be written as $\sigma_{\mathrm{EA}}^2(k_i) = ||\boldsymbol{w}(i)^T \boldsymbol{R}_{\mathrm{EA}}||^2$.

2.7.1 Gaussian technique

We model the residual interference due to errors in the detection process as an *error signal* with Gaussian statistics, zero mean, and variance depending on the error rate. This approach has already been considered in literature, for some particular transmission scenarios. Using the Gaussian approximation the average BER has been obtained for Rayleigh fading channels when $N_d \leq N_A$, [33,34]. In this section we provide the general expression even for the case of $N_d > N_A$ and for any channel $\boldsymbol{H}$, rather than for a specific channel statistics. This provides a tool for evaluating network performance even when active nodes have different channel statistics.

Assuming a BPSK transmission, the transmitted signal s_k takes a value $\pm\sigma_s(k)$, that depends on the number of active antennas. Then, the error signal for stream k is

$$e_k = \hat{s}_k - s_k \in \{-2\sigma_s(k), 0, 2\sigma_s(k)\}\,. \tag{2.5}$$

When the received stream k is affected by a BER $P_e(k)$, $e_k = 0$ with probability $1 - P_e(k)$, $e_k = -2\sigma_s(k)$ with probability $P_e(k)/2$ and $e_k = 2\sigma_s(k)$ with probability $P_e(k)/2$. Note that an erroneous cancellation actually doubles interference. Hence, the variance of the interference signal due to stream k can be written as

$$\sigma_e^2(k) = \mathrm{E}[e_k^2] = 4\sigma_s^2(k)P_e(k) \tag{2.6}$$

and the SINR for stream k_i can be written.

$$\gamma(k_i) = \frac{|\sigma_s^2(k_i)\boldsymbol{w}(i)^T\boldsymbol{R}_{\cdot,k_i}|^2}{\frac{N_0}{2}||\boldsymbol{w}(i)^T\boldsymbol{H}^H||^2 + \sigma_{\mathrm{EA}}^2(k_i) + \sum\limits_{k\in\mathcal{K}(i)} |\boldsymbol{w}(i)^T\boldsymbol{R}_{\cdot,k}|^2\sigma_e^2(k) + \sum\limits_{k\in\mathcal{N}\smallsetminus\mathcal{K}(i+1)} |\boldsymbol{w}(i)^T\boldsymbol{R}_{\cdot,k}|^2\sigma_s^2(k)} \tag{2.7}$$

The first term of the denominator in (2.7) is the power of noise, with power spectral density $N_0/2$ since we are considering a real constellation. The second term of the denominator accounts for the interference due to EA. The power of the residual interference due to imperfect cancellation is provided by the third term in the denominator. With the last term in the denominator we have also inserted the interference due to IA not yet canceled, since when the number of receiving antennas is less than the number of undetected IA, the vector $\boldsymbol{w}(i)$ is not able to completely remove the interference due to streams $k_{i+1}, k_{i+2}, \ldots, k_{N_d}$. Note that, from the definition of $\mathcal{K}(i)$, $\mathcal{N} \smallsetminus \mathcal{K}(i+1)$ is the set of all IA not yet canceled, not including the IA k_i.

The BER for a BLAST transmission over channel $\boldsymbol{H}$ with an uncoded BPSK modulation, is given by $P_e(k_i) = Q\big(\sqrt{\gamma(k_i)}\,\big)$, where $Q(\,\cdot\,)$ is the complementary Gaussian distribution.

2.7.2 Enumeration technique

As we will show in Section 2.8.1, the Gaussian technique is not always accurate, especially with an increasing number of transmitting antennas. Hence, we propose here the *enumeration technique* which takes into account the exact interference statistics by enumerating, at stage i of BLAST, all possible error configurations of the previously detected streams $k_1, k_2, \ldots, k_{i-1}$ and their impact on the probability of erroneous detection of stream k_i. A similar approach has been considered in [35] for the computation of the average BER with the assumptions of two transmit antennas, high signal to noise ratio, and Rayleigh fading. Here we generalize the technique for any channel statistics and any number of antennas.

Define the ordered error vector until detection of stream k_i as $\boldsymbol{e}^{(i)} = [e_{k_1}, e_{k_2}, \ldots, e_{k_{i-1}}]^T$, where entries are provided by (2.5). The error probability on stream k_i can be conditioned on the error configuration of previously canceled streams $\bar{\boldsymbol{e}}$, obtaining, for $i = 2, 3, \ldots N_d$,

$$\mathrm{P}[e_{k_i} = a] \;=\; \sum_{\substack{\bar{e}_\ell \in \mathcal{V}_\ell \\ \ell=1,\ldots,i-1}} \mathrm{P}[e_{k_i} = a \big| \boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}]\, \mathrm{P}[\boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}]\,, \tag{2.8}$$

where $\mathcal{V}_\ell = \{-2\sigma_s(k_\ell), 0, 2\sigma_s(k_\ell)\}$, contains the variance of $\bar{\boldsymbol{e}}_\ell$, $\ell = 1, \ldots, i-1$. Moreover, $a \in \{-2\sigma_s(k_i), 2\sigma_s(k_i)\}$, the summation is taken over all possible error configurations $\bar{\boldsymbol{e}}$ of the $(i-1)$ detected streams, and $\mathrm{P}[\cdot]$ denotes probability. For the first detected stream, we have no previous errors, thus

$$\mathrm{P}[e_{k_1} = +2\sigma_s(k_1)] = \mathrm{P}[e_{k_1} = -2\sigma_s(k_1)] = P_e(k_1)/2\,. \tag{2.9}$$

The expression (2.8) can be computed more efficiently by considering the associated tree of error configurations. An example of error tree is shown in Fig. 2.13 (where for simplicity we set $\sigma_s(i) = 1$, $i = 1, 2, \ldots, N_d$). Starting from the root (level 0), the ith level of the tree corresponds to the detection of stream k_i, and each node corresponds to a different error configuration of the previously detected symbols. In particular, for the first stream k_1, corresponding to the first level after the root, there is only one configuration, since there are no previous detections. Three branches depart from the root, corresponding to the three possible error configurations of $\hat{s}_{k_1}$, *i.e.*, $-2\sigma_s(k_1), 0, 2\sigma_s(k_1)$. The second level corresponds to the detection of the second stream k_2 and there are three possible error configurations of the previously detected stream k_1.

The general level i has a total of 3^i nodes, each representing one term of the summation in (2.8). A node at level i is identified by an error configuration vector $\boldsymbol{e}^{(i)}$ and an error value e_{k_i}. For the computation of the error probability of each term of (2.8) we must consider the signal at the input of the detector, given a specific error configuration. Given an error vector $\boldsymbol{e}^{(i)}$, the signal at the detector input can be written as

$$\tilde{s}_{k_i}|\boldsymbol{e}^{(i)} = s_{k_i} + \sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} + \sum_{k \in \mathcal{N} \smallsetminus \mathcal{K}(i+1)} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k} s_k + \boldsymbol{w}(i)^T \boldsymbol{\nu} + \boldsymbol{w}(i)^T \boldsymbol{i}_{\mathrm{EA}}\,. \tag{2.10}$$

Let us define the two conditional SINRs as

$$\gamma_{s_{k_i}=\pm\sigma_s(k_i)|\boldsymbol{e}^{(i)}} = \frac{\left| \pm \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_i} \sigma_s(k_i) + \sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} \right|^2}{\frac{N_0}{2} ||\boldsymbol{w}(i)^T \boldsymbol{H}^H||^2 + \sigma^2_{\mathrm{EA}}(k_i) + \sum_{k \in \mathcal{N} \setminus \mathcal{K}(i+1)} |\boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k}|^2 \sigma_s^2(k)}, \tag{2.11}$$

where we observe that the interference due to detection errors produces a shift in the position of the received signal, and hence changes its power in the numerator of the SINR.

We must now consider the following cases, according to the transmitted signal and the level of interference due to error propagation. The conditional error probability for stream k_i, assuming equally likely data symbols, can be written as

$$\mathrm{P}\left[e_{k_i} = 2\sigma_s(k_i) \,\middle|\, \boldsymbol{e}^{(i)}, \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_i} \sigma_s(k_i) \geq \sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} \right] = Q\left(\sqrt{\gamma_{s_{k_i}=-\sigma_s(k_i)|\boldsymbol{e}^{(i)}}} \right) \tag{2.12}$$

$$\mathrm{P}\left[e_{k_i} = -2\sigma_s(k_i) \,\middle|\, \boldsymbol{e}^{(i)}, \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_i} \sigma_s(k_i) \geq -\sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} \right] = Q\left(\sqrt{\gamma_{s_{k_i}=\sigma_s(k_i)|\boldsymbol{e}^{(i)}}} \right) \tag{2.13}$$

$$\mathrm{P}\left[e_{k_i} = 2\sigma_s(k_i) \,\middle|\, \boldsymbol{e}^{(i)}, \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_i} \sigma_s(k_i) < \sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} \right] = 1 - Q\left(\sqrt{\gamma_{s_{k_i}=-\sigma_s(k_i)|\boldsymbol{e}^{(i)}}} \right) \tag{2.14}$$

$$\mathrm{P}\left[e_{k_i} = -2\sigma_s(k_i) \,\middle|\, \boldsymbol{e}^{(i)}, \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_i} \sigma_s(k_i) < -\sum_{\ell=1}^{i-1} \boldsymbol{w}(i)^T \boldsymbol{R}_{\cdot,k_\ell} \boldsymbol{e}_\ell^{(i)} \right] = 1 - Q\left(\sqrt{\gamma_{s_{k_i}=+\sigma_s(k_i)|\boldsymbol{e}^{(i)}}} \right) \tag{2.15}$$

The probability of a generic node at level i characterized by the error vector $\bar{\boldsymbol{e}}^{(i)}$ and the error value e_{k_i}, can be obtained as $\mathrm{P}\left[e_{k_i} = \bar{e}_{k_i} | \boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}^{(i)} \right], i = 2, 3, \ldots, N_d$, where $\mathrm{P}[e_{k_1} = \bar{e}_{k_1}]$ is provided by (2.9). This calculation needs averaging over the transmitted symbol, since the same $\boldsymbol{e}^{(i)}$ has a different impact on the error probability, according to the signal sent. Further, averaging $\mathrm{P}\left[e_{k_i} = \bar{e}_{k_i} | \boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}^{(i)} \right]$ over all $\boldsymbol{e}^{(i)}$ yields the total probability of wrong detection for stream k_i.

Lastly, note that we could also consider the various error configurations of EA for the SINR evaluation. In this case, since EA are not detected, the error signal would be equal to $-\sigma_{s'}(k)$ or $\sigma_{s'}(k)$ with equal probability 1/2, for $k = N_d + 1, N_d + 2, \ldots, U$. Even though this modeling would provide more accurate results than approximating EA as Gaussian noise, it would further increase the computational complexity of the enumeration technique.

2.7.3 Pruned tree technique

In order to reduce the computational complexity of the enumeration technique, we limit the exploration of the tree to just a few branches. Sub-trees departing from nodes with more than n_e errors are approximated with an upper bound on the error probability, assuming that when more than n_e errors occur in interference cancellation, the average error probability in forthcoming detections is high. Hence, all nodes of the subtrees having a sub-root with $n_e + 1$ errors are characterized by an error probability of 0.5.

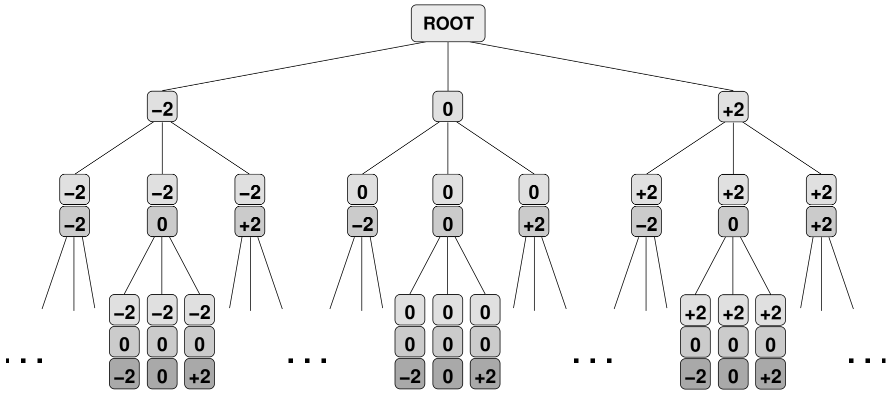


Figure 2.13. *Error configuration tree for* $\sigma_s(i)=1$, $i=1,2,\ldots,N_d$.

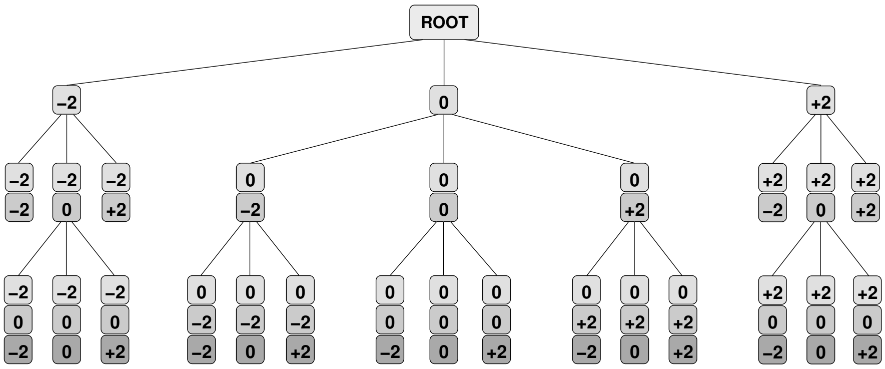


Figure 2.14. *Pruned error tree for* $n_e=2$ *and* $N_d=4$, $\sigma_s(i)=1$, $i=1,2,3$.

For example, for $n_e = 0$ we obtain

$$\mathrm{P}[e_{k_i} \neq 0]|_{n_e=0} \;\approx\; \mathrm{P}[e_{k_i} \neq 0|\boldsymbol{e}^{(i)} = \mathbf{0}]\mathrm{P}[\boldsymbol{e}^{(i)} = \mathbf{0}] + \frac{1}{2}\left[1 - \mathrm{P}[\boldsymbol{e}^{(i)} = \mathbf{0}]\right] . \tag{2.16}$$

This case is similar to the upper bound derived in [36], where it was assumed that given a decision error in an earlier stage of BLAST, the probability of a subsequent decision error is one. In that case the upper bound is as in (2.16) with 1 instead of 0.5 as weight of the summation.

For a general value of n_e, the error configurations with at most n_e errors before the detection of the ith stream are in the set

$$\mathcal{E}^{(i)}(n_e) = \left\{ \boldsymbol{e} : \sum_{\ell=1}^{i-1} \frac{|e_{k_\ell}|}{2\sigma_s(k_\ell)} \leq n_e \right\} . \tag{2.17}$$

The error probability for node k_i is then approximated as

$$\begin{aligned}\mathrm{P}[e_{k_i} \neq 0]|_{n_e} \;\approx\; & \sum_{\bar{\boldsymbol{e}} \in \mathcal{E}^{(i)}(n_e)} \mathrm{P}[e_{k_i} \neq 0|\boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}]\mathrm{P}[\boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}] \\ & + \frac{1}{2}\left[1 - \sum_{\bar{\boldsymbol{e}} \in \mathcal{E}^{(i)}(n_e)} \mathrm{P}[\boldsymbol{e}^{(i)} = \bar{\boldsymbol{e}}]\right] . \end{aligned} \tag{2.18}$$

The last term weighed by $1/2$ accounts for the probability of configurations with more than n_e errors. We observe that the error propagation tree can then be pruned of all nodes that have more than n_e errors, since subtrees departing from these nodes are approximated with an error probability of 0.5. Fig. 2.14 shows an example of a pruned tree for $n_e = 2$, and $N_d = 4$ total IA streams, where at most one detection error is explicitly accounted for and any configuration with more than one error is assumed to yield a correct detection probability of 0.5 in all subsequent stages (corresponding to dead leaves in the tree).

2.7.4 Computational complexity comparison

Various algorithms have been proposed for an efficient implementation of BLAST. In [37] a recursive algorithm for the matrix inversion has been proposed. In [38] a square-root based algorithm is proposed, which has been further optimized in the improved square-root (ISR) algorithm [39]. Here we consider ISR as the most efficient and reliable technique, and hence all simulations and semianalytical techniques have a common base complexity of $(2/3)N_d^3 + (7/2)N_d^2 N_A + \mathcal{O}(N_d^2 + N_d N_A)$ complex multiplications.

For the Gaussian technique, the function Q is computed N_d times, to obtain the corresponding BERs. The computation of Q can be performed by a look-up table and we bound the complexity of each function computation with the equivalent of one complex multiplication. Moreover, SINR must be computed N_d times, according to (2.7), which requires

$$\begin{aligned} C_\gamma &= (N_A{+}3{+}1) + [(1{+}1{+}N_A) + N_d(N_A{+}1) + 1] \\ &= N_d N_A + \mathcal{O}(N_d{+}N_A)\,. \end{aligned} \tag{2.19}$$

products and ratios. Hence, the Gaussian technique has a complexity

$$C_G = (C_\gamma{+}1)N_d = N_d^2 N_A + \mathcal{O}(N_d^2{+}N_A N_d)\,. \tag{2.20}$$

For the enumeration technique, we must compute N_d BER functions for each leaf of the error tree. The computation of the conditional SINRs (2.11) has the same complexity as (2.7). The complete error tree has 3^{N_d-1} leaves, leading to an overall complexity

$$C_E = 2 \cdot 3^{N_d-1} C_G\,. \tag{2.21}$$

For the pruned tree technique, the number of explored leaves is $\lambda(n_e) = \sum_{k=0}^{n_e} \binom{N_d}{k} 2^k$. Hence, the complexity of the pruned error tree algorithm is

$$C_P(n_e) = 2 \cdot \lambda(n_e) C_G\,. \tag{2.22}$$

Note that in (2.21) and (2.22) the factor 2 accounts for the double computation of Q as required by (2.12)–(2.15). We observe that the complexity of the enumeration and pruned tree techniques grows exponentially with N_d, while the Gaussian technique grows as its third power. For example, with $N_d{=}8$ the ratio between the enumeration and the Gaussian technique is $C_E/C_G = 39366$, while for $n_e{=}1$ we have $C_P/C_G = 42$. We conclude that the enumeration technique has a significantly higher complexity than the Gaussian technique and it may be even more demanding than the bit-by-bit simulation. The pruned tree technique instead involves a limited increase of complexity with respect to the Gaussian technique.

2.8 Numerical Results with Approximations

In this Section, we carry out relevant physical level as well as network level simulations that exploit the analytical framework deployed in Section 2.7, and compare results with fully detailed bit–level simulations. A very important conclusion that can be drawn from our results is that a complete and accurate physical layer modeling is crucial for both deciding how different layers interact and tuning relevant communication parameters, since oversimplified models may hamper the statistical significance of simulation results. PHY results are reported in Section 2.8.1, whereas MAC level metrics are given and commented in Section 2.8.3.

2.8.1 Bit error rate

In the following we compare the simulated performance of a generic node in the network with the analytical results derived in Section 2.7, considering channel gains having complex Gaussian statistics with zero mean and unit variance. We have compared the simulated results for different configurations of transmitted streams N_d and receive antennas N_A. The receiving node is assumed to detect all streams so that N_d is also the number of detected streams. We begin by examining this simpler setup as it corresponds to a special-case network with lower load, where all transmitters are placed at the same distance from the receiver, and each of them uses a single antenna at the full available power. Even though this setting may not be representative of a real network, it is a significant benchmark to consider before moving to more realistic scenarios.

In Fig. 2.15 we plot results of simulation and Gaussian technique for the case of a single link. The BER is averaged over all detected streams and various channel realizations and is shown as a function of the average signal to noise ratio (SNR) per antenna. The average BER obtained with the pruned tree technique is shown in Fig. 2.16. Results predicted by the Gaussian approximation technique differ from the simulated ones for two main reasons: i) the interference due to error propagation is not Gaussian, due to different received power levels and signal ordering, and ii) the interference due to imperfect ZF is also not Gaussian. Therefore, the approximation is more accurate at higher SNR, where fewer errors occurs and the error propagation phenomenon is reduced [34]. Moreover, as long as there are fewer streams than receive antennas, ZF is effective in removing all interference and analysis is in accordance with simulation. When $N_d > N_A$, ZF BLAST yields additional interference that is not Gaussian, providing a further mismatch with simulated results.

The evaluation of the average BER, not reported in a separate figure for conciseness, confirms that the enumeration technique has a better match than the pruned error tree

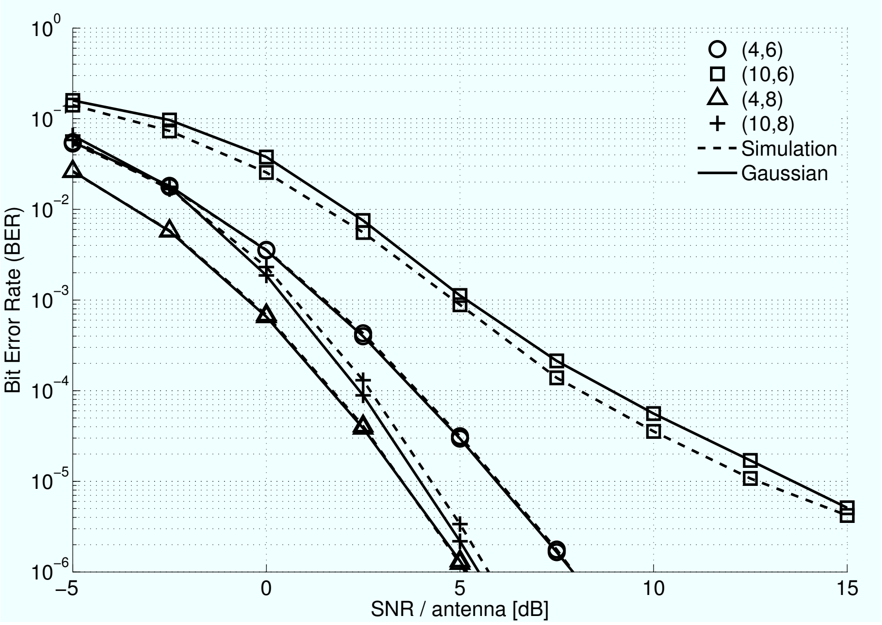


Figure 2.15. *Comparison between analytical (using the Gaussian technique) and simulated BER results for various configurations* (N_d, N_A)*, with* N_d *the detected streams (all assumed to be IA) and* N_A *the receive antennas.*

approach. For the enumeration technique, simulated and analytical BER exhibit a perfect match when $N_d \leq N_A$, while for $N_d > N_A$ a slight mismatch is present, due to the approximation of imperfect ZF as Gaussian interference.

Results reported until now are for the average BER over all detected streams. However, the BER of individual streams may differ from the average behavior, due to different level of interference and power. The per-user BER, averaged with respect to the channel statistics, is shown in Fig. 2.17 for two antenna configurations. The stream index refers to the detection order and variations in the BER among different users result from the combination of different received powers, error propagation and mutual interference. It is important to observe that the Gaussian approximation can be more accurate than the pruned error tree for the estimate of the BER of some streams. Note in fact that approximating the error probability as 0.5 after n_e errors flattens the analytical BER curve of the pruned tree approach.

2.8.2 Network simulation environment

The behavior of a network in a general arrangement can not be directly derived from the BER results given in the previous subsection. Moreover, network performance directly depends on MAC choices. Hence, BER results can not provide the full picture about the accuracy of the analytical methods, since relevant differences at the physical layer may be smoothed out or amplified by network behaviors. Before designing or optimizing protocols based on pseudo-analytical results, it is therefore very important to assess the validity of the approximate techniques in a more general networking scenario.

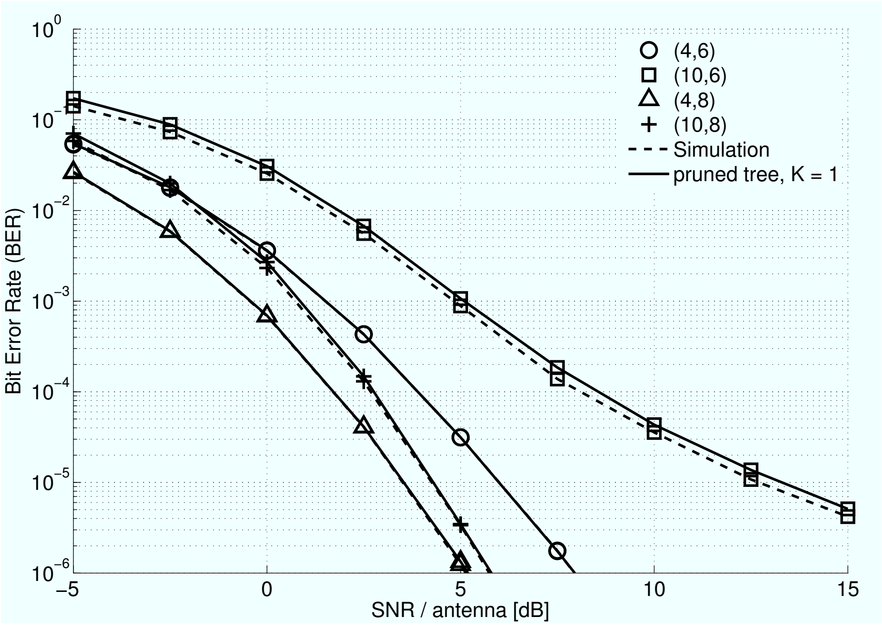


Figure 2.16. *Performance of the pruned tree technique for BLAST BER evaluation for various configurations* (N_d, N_A)*, with* N_d *the detected streams (all assumed to be IA) and* N_A *the receive antennas.*

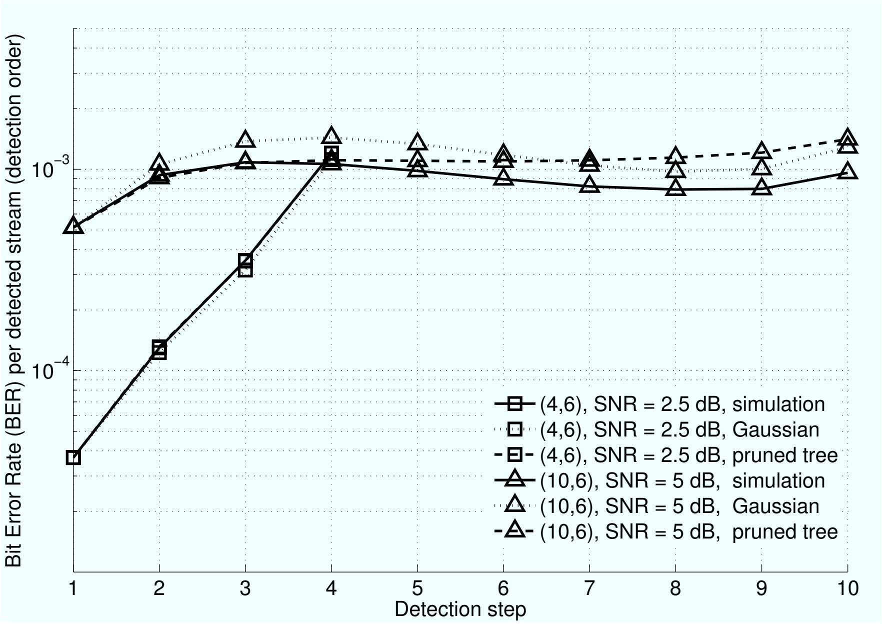


Figure 2.17. *Analytical and simulated BER* per detected stream *for antenna configurations* $(4, 6)$ *and* $(10, 6)$ *and* $\mathrm{SNR} = 2.5$ *dB and* 5 *dB, respectively. The abscissa lists the stream index as per the detection order.*

To this end, we arrange a total of 25 nodes on a grid in a (100×100) m^2 square area, such that the distance between nearest neighbors is $25\,\mathrm{m}$. With this setting, we ensure that the error probability of a transmission between the two nodes at the largest possible distance is below 1%, in the absence of interference. Assuming transmission in the 5.8 GHz band, we can obtain independent fading by packing 8 antennas per node within nearly $20\,\mathrm{cm}$, which fits on the screen of a laptop computer. Considering indoor transmission, we assumed a path loss attenuation coefficient of 4.

Packets are generated according to a Poisson process of rate λ packets per second per node. Each packet is randomly assigned a length of $k \times 1000$ bits, with k uniformly chosen in the set $\{1, 2, 3, 4\}$. Nodes keep backlogged packets in a queue that can store up to 120 1000-bit units. Nodes can track at most $N_d^{MAX} = 32$ training sequences. Transmissions are framed as explained in Section 2.4, and all nodes share the same frame synchronization. Since the frame structure is fixed, the length of signaling packets and data streams is also fixed to be 200 bits for RTSs, CTSs and ACKs and 1000 bits per used transmit antenna for SM data streams, including PHY preambles used, *e.g.*, for channel estimation. The overall frame duration is then 1600 bits, comprising RTS, CTS, data and ACK. Under the preceding assumptions, channels may be assumed to be constant over the whole duration of a frame. Recall that, in order to save overhead, RTSs and CTSs may contain transmission requests and grants to multiple nodes, respectively. Moreover, terminals confirm each stream individually, so that correctly received streams are acknowledged even if other streams belonging to the same packet are not received correctly.

All signaling packets are sent with a single antenna, since in this phase we are not interested in maximum throughput, but rather in maximum probability of success. Sending short packets with one antenna allows both to increase the transmission power and to reduce network load, which results in increased reliability. In this case, RTSs, CTSs and ACKs can travel long distances without errors, maximizing also the nodes' awareness about neighboring activity, according to the value of N_d^{MAX}. We use the distributed ad hoc MAC protocol described in Section 2.4.2and 2.5 to obtain network simulations, where signaling among nodes is limited to RTSs and CTSs and no other information is assumed.

2.8.3 Network results

A key point of our analysis is to verify whether and to what extent using a given approximation method corresponds to accurate networking results, since MAC protocols are likely to have a smoothing effect on discrepancies arising from lower level models. In order to do so, bit-by-bit simulations of the whole system, including complete physical and MAC layer modeling, are provided and compared with those obtained using the Gaussian and the pruned tree approximations. The complete tree exploration is not considered for network results, since it has an exceedingly high complexity. Note that the pruned tree approach provides a better match to simulated average BER than the Gaussian approximation technique, while the Gaussian approximation better follows BER behavior for individual streams. On the other hand, the Gaussian approximation is much less complex than the pruned tree approach.

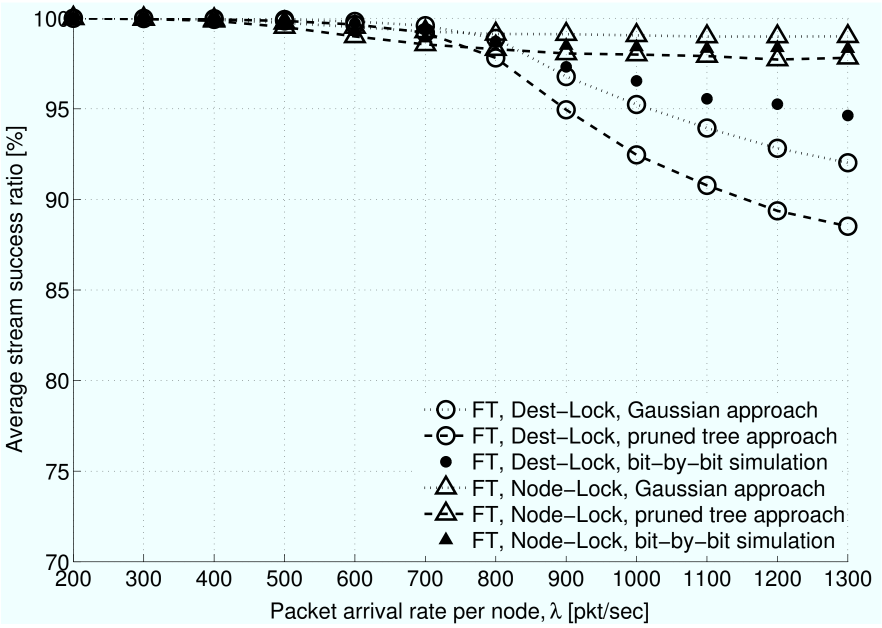


Figure 2.18. *Average 1000-bit stream transmit success ratio as a function of λ, dest–lock and node–lock.*

We use in the following the same network protocols, design and setting as in Sections 2.4.2, 2.5 and 2.6.2. We take the chance to include a further element in our discussion, that is, the backoff mechanism.

As previously discussed in this Chapter, the use of a MIMO architecture leaves considerable degrees of freedom to the protocols. In fact, it allows sources to simultaneously activate links with multiple destinations, and destinations to simultaneously receive packets from multiple sources. In this setting, random backoff is used as a sort of load control, more than as part of the access mechanism.[6] In our network, random backoff is also useful to merge the set of potential source-destination pairs to be activated and to increase the probability that a source finds its intended destination available.

Thus, backoff realizes the tradeoff between activated links and success ratio of transmitted streams, but has also an important effect on the set of links effectively activated. In the numerical results previously shown, link failure results in a random interval, whose window size depends on the number of consecutively failed transmissions, in which the source refrains from transmission. This is only one of the possible choices, that stresses the need for preventing receivers' overload. Another possibility is to lock only the source-destination pair that failed the delivery (whatever is the cause of the failure). Thus, a source can have a subset of the set of destinations of the packets in its queue that is locked by backoff, while it can try to activate links to other destinations. These two choices lead to different results in terms of throughput, latency and success rate. While assessing the accuracy of the ap-

[6] In single access networks, backoff is generally used to have the various sources rescheduling their access in different instants, so that carrier sense can work. Thus, the main goal of backoff in these networks is to decrease the collision rate.

proximation for the SINR previously presented, we provide an initial discussion, that will be prosecuted later in this chapter, on how backoff interact with the overall network performance.

In particular, we distinguish in the following between two different exponential backoff techniques: namely *destination–wise* and *transmitter–wise* backoff. In the destination–wise version, for each request whose CTS is not received, the transmitter defers communication with *that* destination for a random number of frames, randomly distributed in the back-off window $[1, B_{max}]$; in the transmitter–wise version, *all* communications originating from the failing node are deferred regardless of their recipient, as usual in many backoff schemes, *e.g.*, 802.11 DCF [8]. The backoff window is exponentially increased, following the relation $B_{max} = W \cdot 2^{N_f - 1}$ where N_f is the number of subsequently failed attempts. If a maximum preset value N_f^{MAX} is reached, the backoff window is no longer increased. Moreover, packets are assigned a maximum number of attempts, after which they are dropped. The choice of N_f^{MAX} and B_{max} is meant to ensure that a certain number of backoffs with window length B_{max} take place before packet discarding. In the following, we shall refer to destination– and transmitter–wise backoff as *dest–lock*, and *node–lock*, respectively.

The BER results for the single link are closely related to the PER and its complementary function, the network average success rate (1 − PER). In Fig. 2.18 we compare the average success rate obtained by simulation and analytical techniques, considering the average ratio between the number of correctly detected 1000-bit streams and the total number of transmitted streams per frame. We observe that the conclusions derived from the average BER results for a single link are modified by the MAC behavior and the accuracy of the analysis for each stream becomes more relevant. In particular, the pruned tree technique, which provides a good BER match, yields PER values that differ significantly from simulation results due to the limited number of branches considered. On the contrary, the Gaussian technique performs well under both dest–lock and node–lock policies and under all considered traffic levels, since as the number of streams increases, the distribution of interference becomes Gaussian by the central limit theorem.

In order to explore the accuracy of the analytical techniques we considered other network metrics, *i.e.*, average network throughput, average queue length, and delay.

Fig. 2.19 shows the average network throughput, defined as the number of correctly detected 1000-bit streams per frame, as a function of the offered traffic λ. As in Fig. 2.18, the analytical techniques are accurate for low traffic, and the pruned error tree approach exhibits a mismatch with respect to simulated results due to flattening of BER per user. Figs. 2.20 and 2.22 depict the average queue length and the delay, defined as the average time elapsed from the packet generation to the ACK reception following a successful packet transmission. We note that also in this case network parameters are well approximated by the MAC protocol evaluated by the Gaussian technique. About the latency, the smaller throughput predicted by the pruned tree translates into a greater delay in the dest–lock case. On the contrary, with node–lock, both approaches provide very close approximations. Fig. 2.21 shows the aver-

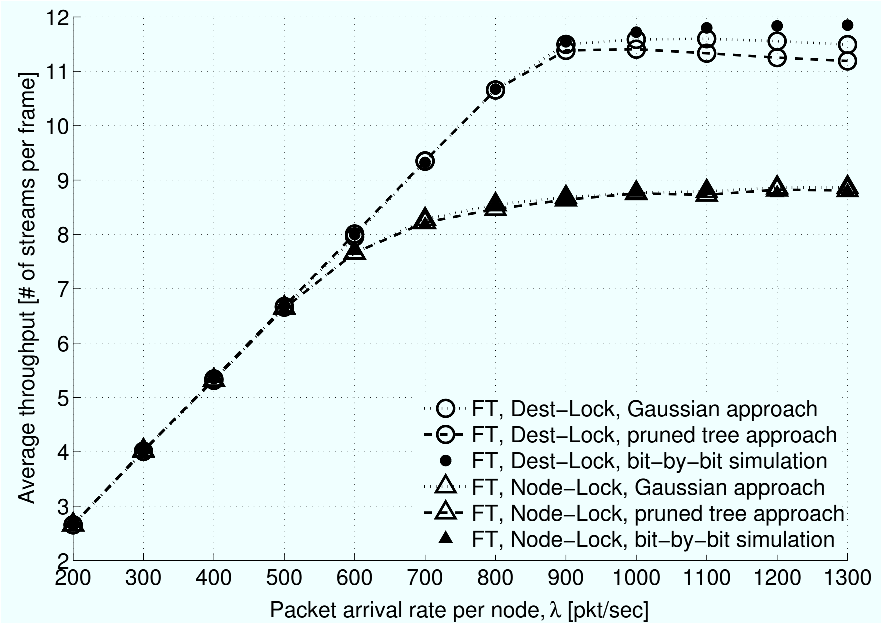


Figure 2.19. *Average network throughput as a function of* λ*, dest–lock and node–lock.*

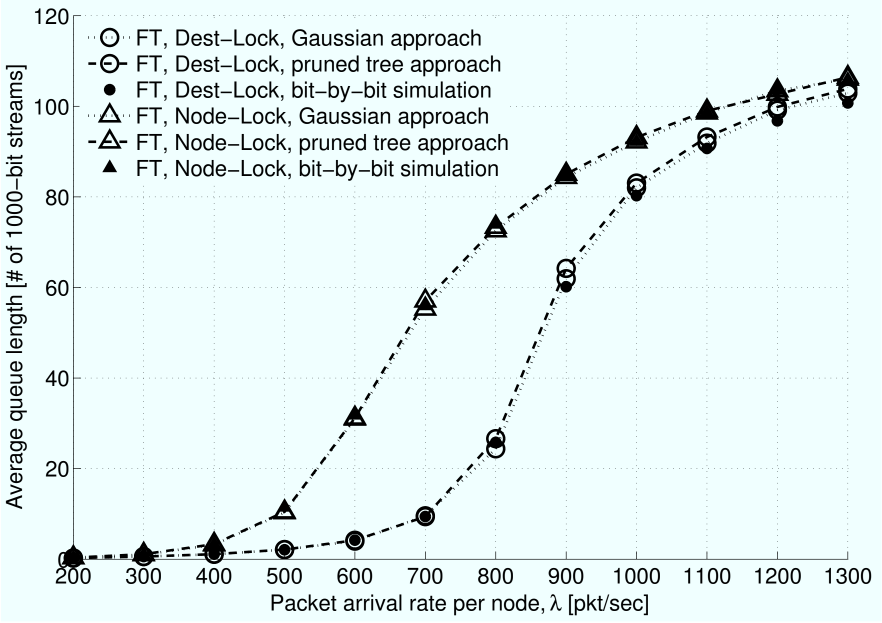


Figure 2.20. *Average queue length as a function of* λ*, dest–lock and node–lock.*

age number of transmit/receive data links that a single node activates per frame, possibly containing more than one 1000-bit stream transmission per link. Both the pruned tree and the Gaussian techniques are indeed very accurate in the dest–lock and the node–lock cases.

We conclude that the Gaussian approximation is well suited to predict network behavior for a wide range of traffic intensities and different MAC policies, and also has the advantage of a limited complexity when compared with the pruned error tree approach. It is then suitable to obtain fast results, for example to compare the two proposed MAC polices. We have also performed simulations in order to assess the dependence of throughput on the number of antennas at each node. Results, not reported here for conciseness, indicate that throughput is roughly linearly dependent on the number of antennas. The dest–lock policy favors transmissions, as it blocks communications only toward a single unavailable receiver each time a failure occurs. As more nodes transmit simultaneously, receivers must perform more cancellation stages, which results in a greater probability of detection errors. This explains the slight throughput decrease of the dest–lock policy. On the other hand, from Fig. 2.21, dest–lock generates more RTSs than node–lock, and for very high packet arrival rates dest–lock may cause more collisions than node–lock. In fact, while node–lock is conservative in backing off the entire node, dest–lock better exploits the available links, letting more nodes transmit, each with a lower number of active links.

2.9 Network Performance: Extended Investigation

The approximation of the receiver performance developed in Section 2.7 allows a considerable improvement in terms of simulation duration. In this Section, we use it to investigate some issues through extensive network simulation. The simulation environment is that considered in previously shown results.

In particular, we have obtained the results shown in Figs. 2.23 to 2.30, where we have compared the behavior and performance of node–lock (NL) and dest–lock (DL) for various values of the initial backoff window.

First, consider Fig. 2.232.24, depicting average throughput (defined as the average number of 1000-bit streams that are successfully received by their intended destinations per frame).[7] DL and NL have different behaviors for varying W. In particular, DL is a more aggressive policy. It allows nodes to send out more requests by just blocking single unavailable destinations. As a consequence, DL performs better than NL only if W is sufficiently high, such that congestion does not occur. For example, for $W = 1, 2, 4, 8$, DL is subject to a decay in throughput performance which is progressively mitigated by increasing W. This decay is mainly caused by the unsustainable amount of traffic generated due to node persistency in transmission attempts which eventually overloads the receiving stage and prevents a correct detection. Conversely, $W = 12, 16$ force longer silences on average, hence it is more likely that receivers become less loaded.

NL, on the other hand, imposes to defer any communication, having any transmitter turn into an available receiver for a given time upon any failure. Anyway, if W is too large the throughput saturates to a suboptimal value. With sufficiently low W, instead, NL outperforms the best throughput reached by DL.

[7] Note that, with 802.11, the maximum throughput attainable in a completely connected network cannot exceed 1 stream per frame.

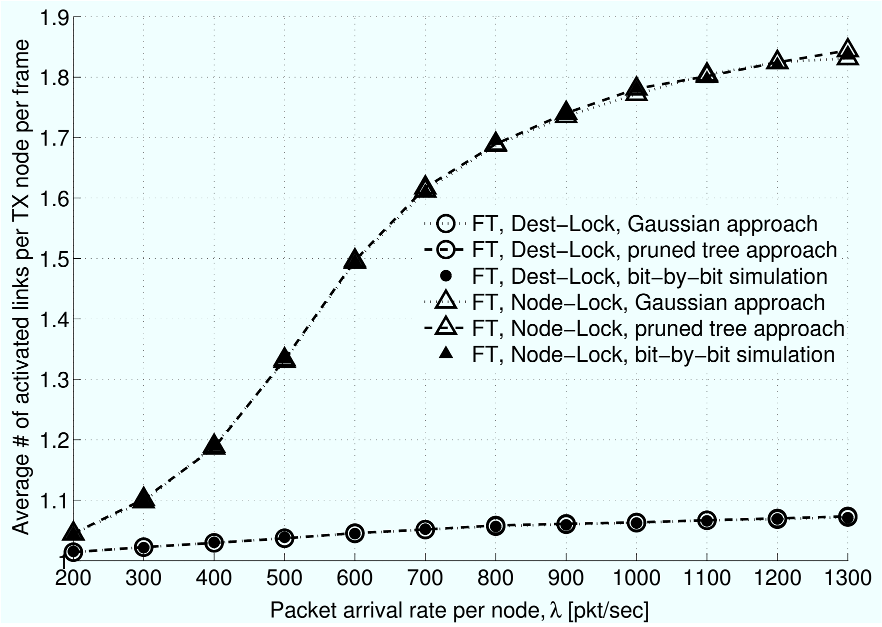


Figure 2.21. *Average number of activated links per transmitting node per frame as a function of* λ*, dest–lock and node–lock.*

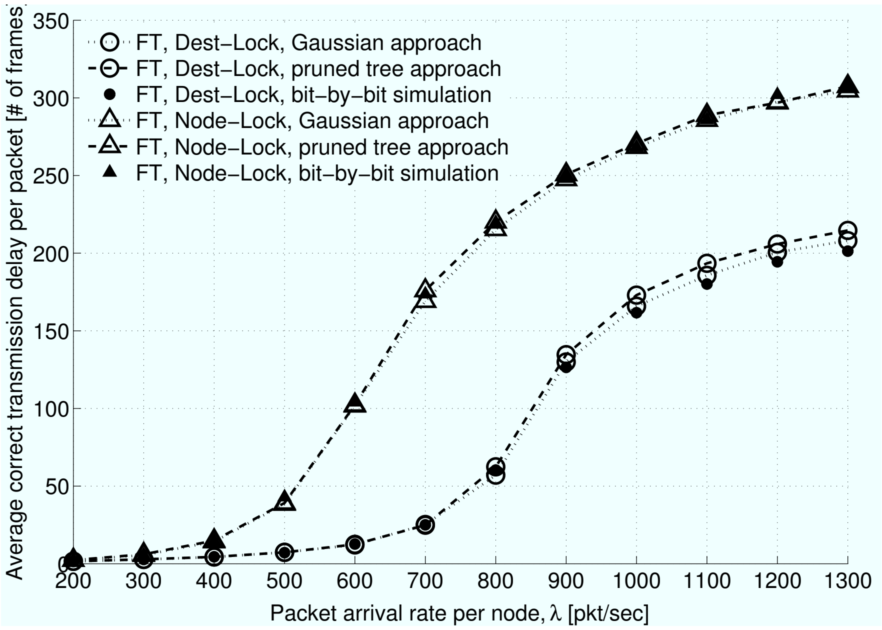
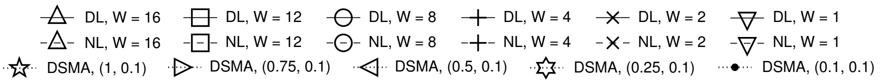


Figure 2.22. *Average correct transmission delay as a function of* λ*, dest–lock and node–lock.*



Even if outperformed by NL from a throughput point of view, DL is very useful for keeping transmission delay (defined as the number of frames from packet generation to the packet transmission that ends correctly) as low as possible. Figs. 2.25 and 2.26 details this

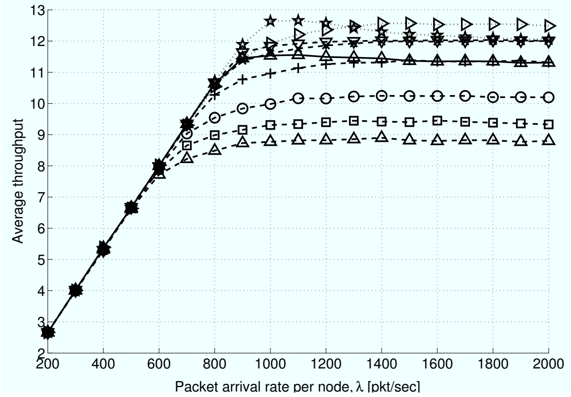


Figure 2.23. *Node–wise backoff (NL).*

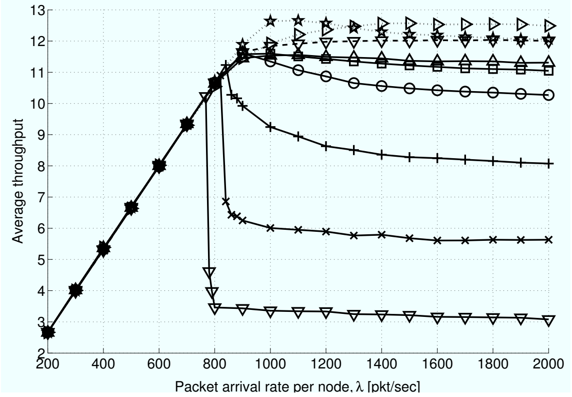


Figure 2.24. *Destination–wise backoff (DL).*

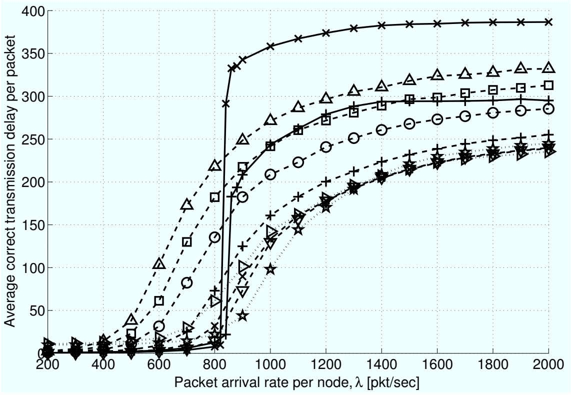


Figure 2.25. *Node–wise backoff (NL).*

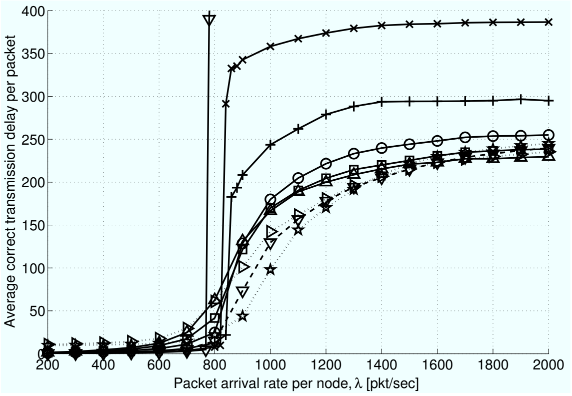


Figure 2.26. *Destination–wise backoff (DL).*

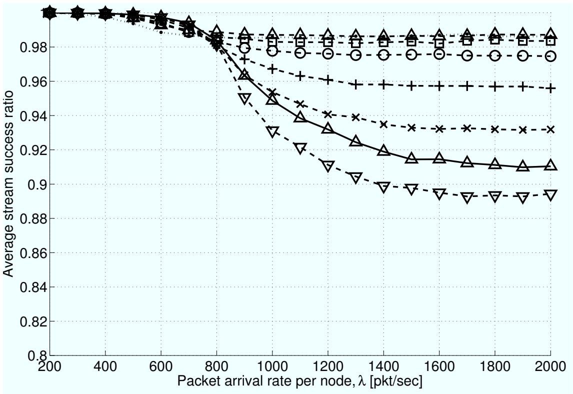


Figure 2.27. *Node–wise backoff (NL).*

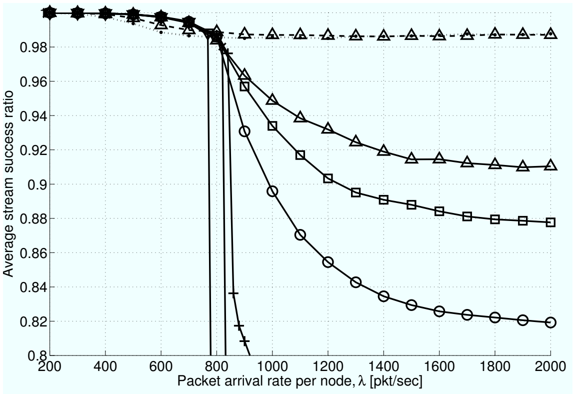


Figure 2.28. *Destination–wise backoff (DL).*

fact, which is a direct consequence of DL's aggressiveness. With DL, nodes can transmit more often, so that in low traffic scenarios they still experience a fair stream success probability with lower delay. Figs. 2.27 and 2.28, depicting the ratio of the correctly received 1000-bit streams to those sent, supports this deduction. Such considerations suggest that DL be used when low traffic is expected, while switching to NL at higher traffic and using, e.g., the average experienced delay as a measure of local network congestion for deciding

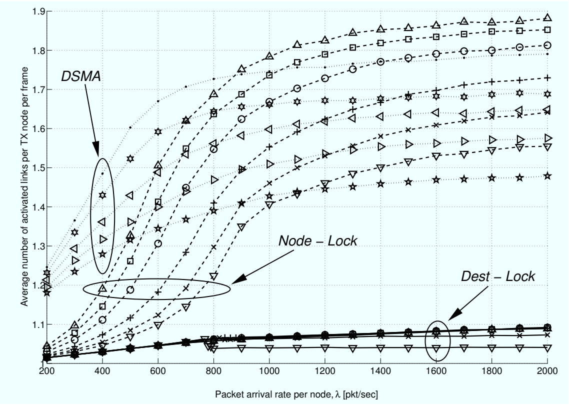


Figure 2.29. *Average no. of links per TX node per frame as a function of traffic.*

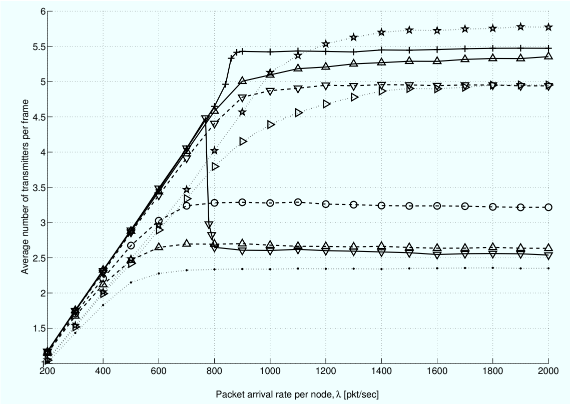


Figure 2.30. *Average no. of transmitters per frame as a function of traffic.*

when to switch from DL to NL. Studying and designing adaptive protocols is out of the scope of this paper, and is currently being addressed.

To obtain further insights on the behavior and applicability of the schemes presented, we depict in Figs. 2.29 and 2.30 the average number of links activated per node per frame, and the average number of transmitters per frame, respectively. By "link," we mean a node–to–node connection, regardless of the amount of spatial multiplexing used. In Fig. 2.30, only three curves per policy are reported. With these figures, it is possible to understand whether high throughput strategies prefer to load single connections with many streams, or to create multiple links each with smaller SM. In the former case, for example, the policy would prove to be more suited to delay–constrained connection–based networking, where it is important to convey high traffic on a given link (e.g., as part of a longer multihop path toward a final recipient). In the latter, the policy would be more applicable to information distribution scenarios, where a single source may want to address several destinations in order to spread traffic faster.

DL tends to allow more transmitters than NL (Fig. 2.30) and correspondingly more one–to–one connections (Fig. 2.29), with each connection having stronger spatial multiplexing, and thus higher throughput. Conversely, even the most permissive NL policy (for $W = 1$) enables a lower number of transmitters, each likely to connect to more than one receiver. NL thus achieves a lower data rate per link, but an overall better aggregate throughput. As a final remark, consider again Figs. 2.23 and 2.24 and 2.27 and 2.28. We note that DL and NL experience high throughput in correspondence of a success ratio near 99%, whereas the max throughput DSMA configuration $(0.75, 0.1)$ undergoes 90% success only.

More insights on backoff mechanism, and a mechanism to have an improved level of coordination among the nodes of the network, are provided in [40].

References

[1] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multiple antennas,” *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[2] H. Jafarkhani, *Space–Time Coding: Theory and Practice*. Cambridge University Press, Sep. 2005.

[3] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V–BLAST: an architecture for realizing very high data rates over the rich–scattering wireless channel,” in *Proc. of IEEE ISSSE*, Pisa, Italy, Sep. 1998, pp. 295–300.

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

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

[7] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bölcskei, “An overview of MIMO communications: a key to gigabit wireless,” *Proc. IEEE*, vol. 92, no. 2, pp. 198–218, Feb. 2004.

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

[9] 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.

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

[11] R. Ramanathan, J. Redi, C. Santivanez, D. Viggins, and S. Polit, “Ad hoc networking with directional antennas: a complete system solution,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 3, pp. 496–506, Mar. 2005.

[12] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto, “A MAC protocol for mobile ad hoc networks using directional antennas,” in *Proc. of IEEE WCNC*, vol. 2, Chicago, IL, Sep. 2000, pp. 1214–1219.

[13] T. Korakis, G. Jakllari, and L. Tassiulas, “A MAC protocol for full exploitation of directional antennas in ad hoc wireless networks,” in *Proc. of ACM MobiHoc*, Annapolis, MD, Jun. 2003.

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

[15] Q. Zhao and L. Tong, “A dynamic queue protocol for multiaccess wireless networks with multipacket reception,” *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 125–137, Feb. 2003.

[16] L. Tong, Q. Zhao, and G. Mergen, “Multipacket reception in random access wireless networks: from signal processing to optimal medium access control,” *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 108–112, Nov. 2001.

[17] H. Chen, F. Yu, H. Chan, and V. Leung, “A novel multiple access scheme over multi–packet reception channels for wireless multimedia networks,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1501–1511, Apr. 2006.

[18] Z. Wang, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves, “Closing the capacity gap in wireless ad hoc networks using multi-packet reception,” in *Proc. of ITA Workshop*, San Diego, CA, Feb. 2008.

[19] B. Chen and M. Gans, “MIMO communications in ad hoc networks,” in *Proc. of IEEE VTC 2005-Spring*, Stockholm, Sweden, May 2005, pp. 2434–2438.

[20] M. Park, S.-H. Choi, and S. M. Nettles, “Cross–layer MAC design for wireless networks using MIMO,” in *Proc. of IEEE GlobeCom*, vol. 2, St. Louis, MO, Nov. 2005, pp. 938–942.

[21] D. Vang and U. Tureli, “Cross–layer design for broadband ad hoc networks with MIMO–OFDM,” in *Proc. of Signal Processing Advances in Wireless Communications*, Jun. 2005.

[22] 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.

[23] S. N. Diggavi, N. Al-Dhahir, and A. R. Calderbank, “Algebraic properties of space–time block codes in intersymbol interference multiple-access channels,” *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2403–2414, Oct. 2003.

[24] F. Rossetto and M. Zorzi, “On gain asymmetry and broadcast efficiency in MIMO ad hoc networks,” in *Proc. of IEEE ICC*, Istanbul, Turkey, Jun. 2006.

[25] 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.

[26] S. Sfar, R. D. Murch, and K. B. Letaief, “Layered space–time multiuser detection over wireless uplink systems,” *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 653–668, Jul. 2003.

[27] 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.

[28] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1983.

[29] G. Ginis and J. M. Cioffi, “On the relation between BLAST and the GDFE,” *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 364–366, Sep. 2001.

[30] M. Biguesh and A. B. Gershman, “Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals,” *IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[31] C. Peel and A. L. Swindlehurst, “Throughput-optimal training for a time-varying multi-antenna channel,” *IEEE Trans. Wireless Commun.*, accepted. [Online]. Available: http://zeidler.ucsd.edu/~muri/pages/publications/lswindlehurst3_accepted.pdf

[32] M. Levorato, P. Casari, and M. Zorzi, “On the performance of access strategies for MIMO ad hoc networks,” in *Proc. of IEEE GlobeCom*, Nov. 2006, pp. 1–5.

[33] C. Shen, Y. Zhu, S. Zhou, and J. Jiang, “On the performance of V-BLAST with zero-forcing successive interference cancellation receiver,” in *Proc. of IEEE GlobeCom*, Dallas, TX, Nov. 2004, pp. 2818–2822.

[34] K. Liu and A. M. Sayeed, “An iterative extension of BLAST decoding algorithm for layered space-time signals,” *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1754–1761, Oct. 2005.

[35] S. Loyka and F. Gagnon, “Performance analysis of the V-BLAST algorithm: an analytical approach,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1326–1337, Jul. 2004.

[36] R. Narasimhan, “Error propagation analysis of V–BLAST with channel estimation errors,” *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 27–31, Jan. 2005.

[37] J. Benesty, Y. Huang, and J. Chen, “A fast recursive algorithm for optimum sequential signal detection in a BLAST system,” *IEEE Trans. Signal Processing*, vol. 1, no. 1, pp. 1722–1730, Jul. 2003.

[38] B. Hassibi, “An efficient square-root algorithm for BLAST,” in *Proc. of IEEE ICASSP*, Istanbul, Turkey, Jun. 2000, pp. 737–740.

[39] H. Zhu, Z. Lei, and F. P. S. Chin, “An improved square-root algorithm for BLAST,” *IEEE Signal Processing Lett.*, vol. 11, no. 9, pp. 772–775, Sep. 2004.

[40] P. Casari, M. Levorato, and M. Zorzi, “DSMA: an access method for MIMO ad hoc networks based on distributed scheduling,” in *Proc. of ACM IWCMC*, Vancouver, Canada, 3–6, 2006.

Chapter 3

Error Control

Contents

In the previous chapter, we designed a structure for the coordination and the distributed control of ad hoc networks where nodes simultaneously access the channel. We showed how coordination allows nodes to control interference in the network according to the capabilities of the receiver.

In this chapter, we investigate the issues arising when nodes can access the channel at any time. This in order to gain some understanding on the complex interaction between the various components of the system. In fact, in this scenario, the distribution of interference becomes a main issue, as it influences and is influenced by all the control mechanisms.

We focus especially on error and rate control, in a framework where interference effect is stressed by the lack for a specific access control policy.[1]

There are many interesting aspects that need an in depth investigation. Consider for instance error control. a retransmission based error control scheme may result in a higher interference unpredictability due to the higher overall birth rate. On the other hand, a

[1] As we will discuss in detail later in this chapter, nodes do not perform carrier sense and can access the channel even while other communications are active. However, we introduce random backoff to keep load under control and decorrelate the channel coefficients of consecutive delivery attempts.

coding-based error control correlates interference conditions, as it results in longer transmissions with an improved success probability. Thus, unpredictability of interference affects error probability, as it diminishes the effectiveness of the rate control mechanism. In turn, a greater error probability results in a greater average number of retransmissions, that means more interference.

Our goal is to identify these fundamental mechanisms that tie together interference and control in order to derive directions for the design of the network. In order to do that, we develop a novel analytical tool, based on recursive update of interference statistics.

In this chapter we make the following original contributions: (i) we explicitly include the use of ARQ and HARQ schemes in the considered system, and study their interactions with MUD and multiple access; (ii) we address the issue of multiple access MUD performance in an ad hoc networking setting, and develop a novel analytical framework to model the system and evaluate the metrics of interest – this framework, based on the theory of renewal and semi-Markov processes, accurately models interference relationships, and explicitly accounts for transmission overlaps and for the different statistics of the interference duration and rate of transmission due to biased sampling that results from random observations in time; (iii) as a concrete example of application, we provide specific results for a detailed system, namely a multiple access scheme in an ad hoc network based on DS-CDMA and HARQ, with MF-LSIC at the receiver. However, we remark that our analytical framework has a much wider applicability, and can be used in general to study multiple access systems with MUD and HARQ.

The reference papers for this part of the thesis are [J7ml, C17ml] (see Appendix B).

This chapter is organized as follows. We describe the system under investigation in Section 3.1. We develop the analysis in Sections 3.2 and 3.3. In Section 3.4 we presents numerical results assessing the accuracy of the analysis and highlighting the issues listed before.

3.1 System Description

We investigate the performance of an ad hoc network where source nodes have to deliver packets of fixed length L [bits] to their intended destinations. Nodes transmit with fixed power P_{t}, and the transmission rate is set according to the perceived post-processing SINR.

3.1.1 Receiver Model

In this Section we summarize the considered transmitter/receiver structure and the performance approximation derived in [1], which is used to model the output SINR throughout the paper.

In particular, we focus on single antenna direct-sequence CDMA (DS-CDMA) systems, where multiple users transmit over the channel using distinct signature waveforms. Besides the well–known conventional DS-CDMA MF receiver, several other structures have been proposed. The DS-CDMA MF linear successive interference cancellation (MF-LSIC) receiver [2] sequentially decodes and cancels the signals in descending power order using an MF receiver at each stage. The MMSE-LSIC receiver [3,4] is similar to the MF-LSIC receiver, except that an MMSE receiver is employed for signal detection. An improved version of the MMSE-LSIC receiver has been proposed in [5]. A powerful and attractive receiver structure is the joint iterative decoder [6–8], where the receiver performs an iterative algorithm exchanging soft information between the receiver components. A comparison of several DS-CDMA receivers is provided in [9].

The good performance of LSIC receivers and their relatively low complexity compared with maximum likelihood (ML) optimum receivers have generated a considerable effort in characterizing their behavior. In this paper, we focus our attention on DS-CDMA LSIC receivers, taking the results contained in [1] as the starting point of our analysis.

We consider our on DS-CDMA MF and MF LSIC receivers. For the sake of simplicity, in our analysis we assume chip–synchronous transmissions, and we refer the reader to [10] for an in–depth discussion of the performance of asynchronous systems. Let $\gamma_1, \ldots, \gamma_K$ be the received powers of K users transmitting over an AWGN channel with binary phase–shift keying (BPSK) modulation. The input of the MF bank corresponding to the jth symbol is $\boldsymbol{c}^j = \sum_{i=1}^{K} \sqrt{\gamma_i} b_i^j \boldsymbol{s}_i + \boldsymbol{n}^j$, where $\boldsymbol{s}_i = [s_i^1, \ldots, s_i^N]$ is the vector of the N chips of the spreading sequence of the ith user and b_i^j is the jth bit of the ith user. $\boldsymbol{n}^j$ is the noise vector, whose N elements $n_1^j, \ldots, n_N^j$ are modeled as uncorrelated Gaussian random variables, with zero mean and variance σ^2. $\boldsymbol{c}^j$ contains the samples corresponding to the jth transmitted symbol. The BPSK symbols b_i^j, $\forall j, i$ are assumed to be independent and identically distributed. In the following we assume that s_i^h, $h = 1, \ldots, N$ are i.i.d. random variables with $s_i^h \in \{\pm 1/\sqrt{N}\}$, and $\mathcal{P}[s_i^h = 1/\sqrt{N}] = \mathcal{P}[s_i^h = -1/\sqrt{N}] = 1/2$.

The MF LSIC receiver sequentially decodes and removes from the overall received waveform the individual received signals in decreasing power order. At each of the K stages the receiver selects the user with the strongest received power, and performs the decoding. Assuming that the users are labeled in decreasing power order, so that $\gamma_i \geq \gamma_{i+1}$, $i = 1, \ldots, K-1$, the decision variable at the mth stage is $z_m^j = \boldsymbol{s}_i^T \tilde{\boldsymbol{e}}_m^j$, where $\tilde{\boldsymbol{e}}_m$ is the input of the mth stage of the MF LSIC receiver. The estimated symbol is then $\tilde{b}_m^j = sgn(z_m^j)$, where $sgn(\cdot)$ is the signum function. The estimated symbol is then rescaled with the amplitude estimate of the MF bank, respread with $\boldsymbol{s}_m$ and subtracted from the received signal.

The mth stage input can be written as [11]

$$\tilde{\boldsymbol{e}}_m^j = \left(\boldsymbol{I} - \boldsymbol{s}_{m-1}\boldsymbol{s}_{m-1}^T\right) \tilde{\boldsymbol{e}}_{m-1}^j = \sum_{i=1}^{K} \sqrt{\gamma_i} b_i^j \boldsymbol{T}_m \boldsymbol{s}_i + \boldsymbol{T}_m \boldsymbol{n}, \tag{3.1}$$

where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{T}_m = (\boldsymbol{I} - \boldsymbol{s}_{m-1}\boldsymbol{s}_{m-1}^T) \ldots (\boldsymbol{I} - \boldsymbol{s}_1\boldsymbol{s}_1^T)$. The decision variable at stage m is $\tilde{z}_m^j = \sqrt{\gamma_m} b_m^j \tilde{\psi}_{mm} + \sum_{i \neq m} \sqrt{\gamma_i} b_i^j \tilde{\psi}_{mi} + \tilde{n}_m$, where $\tilde{\psi}_{mi} = \boldsymbol{s}_m^T \boldsymbol{T}_m \boldsymbol{s}_i$ is the effective cross-correlation and $\tilde{n}_m = \boldsymbol{s}_m^T \boldsymbol{T}_m \boldsymbol{n}$ is the effective noise component.

In [1], approximations for the residual cancellation errors, the effective noise power and the interference due to still undecoded signals are derived. The approximated output SINR for the mth decoded user for the MF LSIC receiver, given the received powers $\gamma_1, \dots, \gamma_K$, sorted in decreasing power order is

$$\mathrm{SINR}^{\mathrm{LSIC}} \approx \frac{\gamma_m \left(1-\frac{1}{N}\right)^{2(m-1)}}{\sum_{j<m} \gamma_j \left(\frac{j-1}{N^2}\right) + \sum_{j>m} \gamma_j \frac{1}{N} \left(1-\frac{1}{N}\right)^{m-1} + \sigma^2 \left(1-\frac{1}{N}\right)^{m-1}}, \tag{3.2}$$

For the MF receiver, the output SINR is

$$\mathrm{S}INR^{\mathrm{MF}} \approx \frac{\gamma_m}{\sum_{j\neq m} \frac{\gamma_j}{N} + \sigma^2}. \tag{3.3}$$

3.1.2 Communication Protocol

The protocol divides data transmission into three phases. In the first phase source and destination perform the handshake, in which the source transmits a request packet and the destination responds with a confirmation packet[2]. If the handshake succeeds, the source performs the second phase transmitting the data packet. In the third phase the destination sends out a feedback packet, in which it reports whether or not the packet has been successfully decoded. Our scheme provides that the confirmation packet contains the post–processing SINR associated with the source signal, as perceived by the destination during the reception of the request packet. Based on this value, the source sets the transmission rate of the data packet following one of the rate/error control policies listed below. Since handshake and feedback packets are generally much shorter than data packets, in the following discussion we idealize these parts of the communication. In particular, as a first step in this analysis, we assume that the handshake packet exchange and the destination feedback are error–free, that these phases are performed transmitting at fixed rate and that they do not interfere with ongoing communications[3]. We set the handshake duration to T_{H}.

Since interference and channel gain conditions may vary during a communication due to fading and to the start and end of other transmissions, we have that the post–processing SINR $S(t)$ of the intended signal is a function of the time index t, where $t=0$ is the start of the transmission. We assume that sources use a binary capacity–achieving code, so that for sufficiently long codewords the error probability vanishes when the link capacity is higher than the transmission rate. The encoded bits are then modulated and transmitted.

To characterize the outage event we adopt the integral form

$$\xi_T = \left\{ W \int_{T_{\mathrm{H}}}^{T_{\mathrm{H}}+T} \log_2\left(1+S(t)\right)\mathrm{d}t < L \right\}, \tag{3.4}$$

[2] Unlike in 802.11, this exchange is performed without resorting to carrier sensing, thanks to the MUD capabilities of the receiver.

[3] This assumption is satisfied for instance using a robust modulation/coding scheme or a dedicated control channel for the exchange of the handshake packets.

where W is the bandwidth, $T = L/R$ is the transmission duration and R [bits/s] is the transmission rate. The integral form is the limit of the sum of the capacity for parallels channels, where fragments of the same codeword are sent over different channels, and is useful to keep the framework general. However, some scenarios, such as block fading and time slotted communications, allow the classic sum–rate capacity formulation, where the integral is replaced by the sum of the capacities of time intervals in which the SINR is assumed constant.

Note that $S(0)$ corresponds to the SINR at the start of a handshake transmission, while $S(T_{\mathrm{H}})$ corresponds to the SINR at the beginning of the data packet transmission. In the following we describe the considered transmission protocols.

3.1.3 Rate and Error Control

As mentioned before, communication is set up with a handshake phase meant to check destination availability and select the transmission rate. The computation of the minimum transmission time T^* is based on the instantaneous channel conditions during the handshake. This corresponds to the maximum rate that allows correct decoding if the channel remains constant during the transmission. We define this value for the rate as $R^* = W \log_2 (1 + S(0))$. For all the described policies the maximum allowed transmission duration is equal to T_{max}.

• *ARQ Protocol*: In this protocol the source sets the value of the transmission rate to $R = R^*$. If a reception failure occurs, the source performs a further delivery attempt, including the handshake, after a random backoff interval. The process continues until the destination successfully decodes the packet or the maximum allowed number of transmissions F is reached. If a failure occurs at the F-th transmission attempt, the source dismisses the packet, leaving its recovery to the higher protocol layers. We denote as communication the whole data packet delivery attempt, including the possible retransmissions.

• *type I HARQ Protocol*: In type I HARQ schemes the packet is encoded with a rate $\rho \leq 1$ code. Assuming a capacity–achieving code this simply results in a lower transmission rate. Therefore we set the rate to $R = \rho R^*$, and consequently the duration is $T = T^*/\rho$. Thus, the smaller the value of ρ, the larger the redundancy sent. If the destination fails to decode the packet, the source starts the retransmission process as in the ARQ protocol before discarding the packet.

• *type II HARQ Protocol*: In this protocol the packet is encoded with a low rate code obtaining a long codeword and each delivery attempt is divided into two phases. In the first phase the source transmits a portion of the codeword, that corresponds to a transmission rate $R' = \eta' R^*$ for a time equal to $T' = T^*/\eta'$, then the destination sends out the feedback packet. If a decoding failure occurs, the feedback packet contains the value $S(T' + T_{\mathrm{H}})$, and the source starts the second phase. The transmission rate for the second phase is $R'' =$

$\eta'' W \log_2(1 + S(T' + T_\mathrm{H}))$, and therefore the second phase duration is

$$T'' = \frac{L - W\int_0^{T'} \log_2(1 + S(t))dt}{R''}. \tag{3.5}$$

η' and η'' are constrained only to be positive real numbers. The total data transmission time is $T = T' + T''$.

This protocol allows a greater adaptation to channel variations than type I HARQ, since the rate is computed again taking into account the perceived SINR during the first phase. While $\eta' \leq 1$ is a conservative choice, for $\eta' > 1$ the protocol is more aggressive, because the source selects a rate higher than the estimated capacity to shorten the transmission in case of good channel, relying on the second phase if the channel does not support the chosen rate.

As for the previous protocols, if the destination reports a failure at the end of the transmission the source performs a further independent delivery attempt, until a success is achieved or the maximum number of allowed transmissions F is reached.

3.2 System Analysis

In this Section we derive the interference and SINR distributions needed for the performance analysis of the following Sections. The node density is equal to μ $[\mathrm{nodes/m^2}]$ and packet arrivals are modeled as a Poisson process of intensity λ [pkts/s] per node. We assume that the maximum destination distance is set to R_max[4], and that the position of the destination node for a packet is uniformly distributed in the circular area of radius R_max.

To model the interference, we consider a circular area A of radius R_max, centered on the destination node. Therefore, given the node spatial distribution and the per node packet arrival rate λ, the overall arrival process of all transmitting sources is a Poisson process of intensity $\nu = \mu\lambda\pi R_\mathrm{max}^2$, and their positions are uniformly distributed in A. Fig. 3.1 depicts an example of source-destination pair placements, where the source and the destination of the communication we are focusing on, and the interfering sources and destinations are denoted with $\boldsymbol{S}$, $\boldsymbol{D}$, $\boldsymbol{I_k^S}$ and $\boldsymbol{I_k^D}$, respectively.

The probability that the intended signal source $\boldsymbol{S}$ is at distance $\delta \leq \delta^*$ from node $\boldsymbol{D}$ is

$$\mathcal{F}_\delta(\delta^*) = \mathcal{P}\left\{\mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) \leq \delta^*\right\} = \frac{(\delta^*)^2}{R_\mathrm{max}^2}, \tag{3.6}$$

with associated pdf $f_\delta(\delta^*) = \mathrm{d}\mathcal{F}(\delta^*)/\mathrm{d}\delta^* = 2\delta^*/R_\mathrm{max}^2$, where $\mathcal{D}(N_1, N_2)$ is the distance between nodes N_1 and N_2. Observe that in this framework the distance of the interfering transmitters $\boldsymbol{I_k^S}$ with respect to $\boldsymbol{D}$, and the distance between $\boldsymbol{I}_k^S$ and $\boldsymbol{I}_k^D$ are also distributed according to $F_\delta(\delta^*)$, i.e.,

$$\mathcal{P}\left\{\mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) \leq \delta^*\right\} = \mathcal{P}\left\{\mathcal{D}(\boldsymbol{D}, \boldsymbol{I}_k^S) \leq \delta^*\right\} = \mathcal{P}\left\{\mathcal{D}(\boldsymbol{I}_k^S, \boldsymbol{I}_k^D) \leq \delta^*\right\} = \mathcal{F}_\delta(\delta^*). \tag{3.7}$$

[4] Nodes farther away are neglected as a source of interference and are not chosen as destinations for packets generated by the nodes

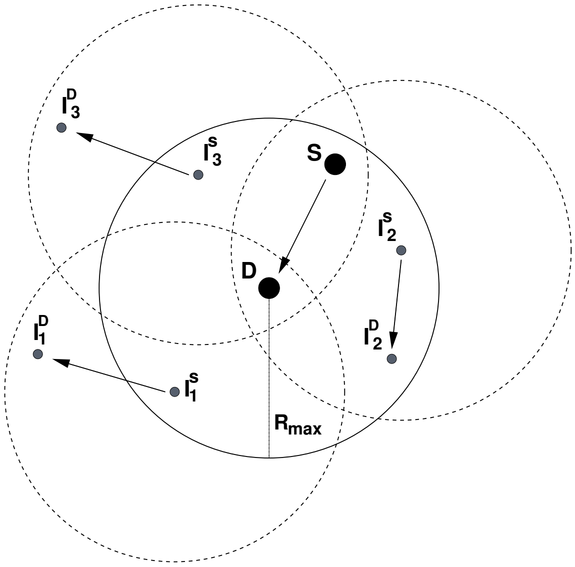


Figure 3.1. *Example of the considered scenario, the source and the destination are denoted with $\boldsymbol{S}$ and $\boldsymbol{D}$, respectively. Interfering source-destination pairs are denoted with $\boldsymbol{I}_k^S$ and $\boldsymbol{I}_k^D$*

3.2.1 Distribution of the Number of Interferers

In this Section we derive the distribution of the number of interfering transmissions during the communication between $\boldsymbol{S}$ and $\boldsymbol{D}$.

We denote with $\mathcal{G}_\delta(\tau)$ the cdf of the time duration of a generic communication where the source node and the destination node are placed at distance δ. In particular, given the distance δ, the probability that the communication duration T, without considering the hanshake, is less than or equal to τ is $\mathcal{G}_\delta(\tau) = \mathcal{P}\{T \le \tau \mid \delta\}$. Moreover, we define $\Psi_\delta = \int_0^{T_{\max}} \mathcal{P}\{\xi_\tau \mid \delta\} \,\mathrm{d}\mathcal{G}_\delta(\tau)$ as the average failure probability of a single transmission between nodes N_1 and N_2, where $\mathcal{D}(N_1, N_2) = \delta$. Thus, the average number of transmissions for the communication between nodes N_1 and N_2 is

$$\Delta_\delta = \sum_{h=1}^{F-1} h\Psi_\delta^{h-1}(1-\Psi_\delta) + F\Psi_\delta^{F-1} = \frac{1-\Psi_\delta^F}{1-\Psi_\delta}. \tag{3.8}$$

In the following we assume that $\mathcal{G}_\delta(\tau)$ and Ψ_δ are known. Section 3.3 describes the recursive process through which these distributions are computed.

It is important to observe that the retransmission process provided by the HARQ/ARQ protocols biases the distribution of the distances between the various sources and their intended destinations, while the distribution of the distance between the interfering transmitters and the other communication destinations remains $\mathcal{F}_\delta(\delta^*)$. This is one of the key points of our investigation, because any changes in the source-destination distribution may heavily affect the system interference distribution, due to the dependence between the destination distance and the communication length. In fact, although the rate control has the

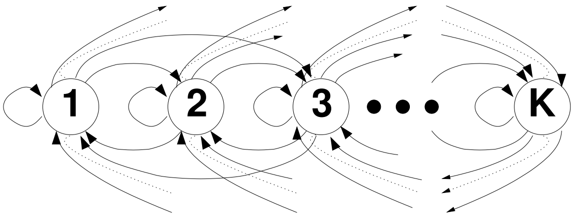


Figure 3.2. *Graphical representation of the embedded chain of the Semi–Markov process.*

aim of ensuring equal reliability to in range communications, transmissions to distant destinations may achieve worse performance due to the longer duration on average, that may reduce interference correlation, and the worse channel statistics. Moreover, the minimum rate constraint may also increase the failure probability of these communications.

Therefore transmissions to distant destinations generally suffer higher failure probability with respect to those directed to closer destinations, and the former generally incur a higher number of retransmissions. Moreover, as stated before, the higher the distance of the destination, the higher the probability that the perceived SINR is low, and consequently the source transmission has a longer duration due to the lower transmission rate. Therefore, due to the retransmission process, the number of interfering nodes and the probability that an interfering node is active at a given time t given that it was active at time $t - t^*$ are generally increased with respect to the single transmission case and the statistics of the transmission duration is biased.

Consider now a single source node that selects packet destinations with distance distribution $\mathcal{F}_\delta(\delta^*)$ and then transmits according to the communication protocol described in Section 3.1. When a packet is either delivered or discarded, the source selects another destination for the next packet and so on. We sample the process at instants t_k, $k = 1, \dots, \infty$, corresponding to the beginning of a transmission (or retransmission), and we call δ_{t_k} the distance between the source and the intended destination of the transmission starting at t_k. We are interested in the steady–state distribution of δ_{t_k}, i.e., $\lim_{k\to\infty} \mathcal{P}\left\{\delta_{t_k} \leq \delta^*\right\}$. This process can be modeled with a Semi–Markov process whose embedded chain is shown in Fig 3.2, where state i, $i = 1, 2, \dots, I$, represents a complete communication, that may include several transmissions, to a destination at distance in $((i-1)\Theta, i\Theta]$, with $I\Theta = R_{\text{max}}$. Note that the transition probability between a state i and a state j does not depend on i, due to the independence of the destinations selection[5], and is equal to

$$p_{\{i,j\}} = \int_{(j-1)\Theta}^{j\Theta} f_\delta(\delta^*)\mathrm{d}\delta^* = \mathcal{F}_\delta(j\Theta) - \mathcal{F}_\delta((j-1)\Theta). \tag{3.9}$$

[5] Note that the semi-Markov process model is not strictly necessary, due to independence of destination selections. However, it makes the discussion more intuitive and keeps the following derivation more general.

We also define the average failure probability of a single transmission given state i as

$$\Psi_i = \mathcal{P}\left\{\text{error} \mid i\right\} = \int_{(i-1)\Theta}^{(i)\Theta} \Psi_\delta \mathrm{d}\delta. \tag{3.10}$$

The average time, expressed in transmissions, that the Semi–Markov process spends in state i[6] is equal to the average number of transmissions that the source performs when communicating with a destination at distance in $((i-1)\Theta_D, i\Theta_D]$, i.e.,

$$S_i = \frac{1-\Psi_i^F}{1-\Psi_i}. \tag{3.11}$$

Observe that all rows of the transition matrix are equal to each other, and thus the steady–state probability of state i is $\pi_i = p_{\{.,i\}}$. Therefore, the average fraction of time, in transmissions, that the Semi–Markov process spends in states $i \leq i^*$ is [12]

$$B_{i^*} = \frac{\sum_{i=1}^{i^*} \pi_i S_i}{\sum_{i=1}^{I} \pi_i S_i}. \tag{3.12}$$

Let $\Theta \to 0$ with $\Theta i^* = \delta^*$ and $\Theta I = R_{\text{max}}$. Then,

$$\lim_{\Theta\to 0} \pi_{i^*} = \lim_{\Theta\to 0} \mathcal{F}_\delta((i^*+1)\Theta) - \mathcal{F}_\delta(i^*\Theta) = \mathcal{F}_\delta(\delta^*)\mathrm{d}\delta^* \qquad \lim_{\Theta\to 0} \Psi_{i^*} = \Psi_{\delta^*}, \tag{3.13}$$

and therefore, we get

$$\lim_{\Theta\to 0} B_i = \lim_{\Theta\to 0} \frac{\sum_{i=1}^{i^*} \pi_i S_i}{\sum_{i=1}^{I} \pi_i S_i} = \frac{\int_0^{\delta^*} \Delta_\delta f_\delta(\delta)\mathrm{d}\delta}{\int_0^{R_{\max}} \Delta_\delta f_\delta(\delta)\mathrm{d}\delta} = \mathcal{F}'_\delta(\delta^*). \tag{3.14}$$

$\mathcal{F}'_\delta(\delta^*)$ represents the distribution of the distance of the destination of a new transmission. A similar argument could be applied to derive the distance distribution of the destination of an ongoing communication, denoted with $\mathcal{F}''_\delta(\delta^*)$. The average duration of a transmission to a destination at distance δ is

$$\Omega_\delta = \int_0^{T_{\max}} \left(1 - \mathcal{G}_\delta(\tau)\right) \mathrm{d}\tau. \tag{3.15}$$

The process required for obtaining $\mathcal{F}''_\delta(\delta^*)$ is similar to what described for the new communications distribution, except that in this case we continuosly sample the process, so that we obtain a continuous–time Semi–Markov process. Thus, the average time, in seconds, the Semi–Markov process spends in state i is $V_i = S_i\Omega_i$, where

$$\Omega_i = \int_{(i-1)\Theta}^{i\theta} \Omega_\delta \mathrm{d}\delta. \tag{3.16}$$

[6] The described chain can be seen as the reduced version of the chain where each state i is composed of F states i_u, each representing a single transmission. The process moves from i_u to i_{u+1}, $u < F$, with probability Ψ_i, while with probability $(1-\Psi_i)p_{\{i,j\}}$ moves toward one of the states i_j, $j = 1, \ldots, I$. From states i_u it moves to j_1 with probability $p_{\{i,j\}}$. The average time spent in states $i_1, \ldots, i_F$ is s_i.

Hence, with a derivation analogous to that of $\mathcal{F}'_\delta(\delta^*)$ we get

$$\mathcal{F}''_\delta(\delta^*) = \frac{\int_0^{\delta^*} \Delta_\delta \Omega_\delta f_\delta(\delta) \mathrm{d}\delta}{\int_0^{R_{\max}} \Delta_\delta \Omega_\delta f_\delta(\delta) \mathrm{d}\delta}. \tag{3.17}$$

We also define $f'_\delta(\delta^*) = \mathrm{d}\mathcal{F}'_\delta(\delta^*)/\mathrm{d}\delta^*$, $f''_\delta(\delta^*) = \mathrm{d}\mathcal{F}''_\delta(\delta^*)/\mathrm{d}\delta^*$.

We remark that $\mathcal{F}_\delta(\delta^*)$ represents the a priori distribution of the distance of nodes selected as destination for the packets, while $\mathcal{F}'_\delta(\delta^*)$ represents the distribution of the distance between the source and the destination of a new transmission, due to the dependence between the distance and the attempt rate (longer links need more transmission attempts and are therefore more likely to occur). $\mathcal{F}''_\delta(\delta^*)$ is the distribution of the distance of an ongoing transmission, resulting from the dependence between the destination distance and the attempt rate and length (when sampling in time, it is more likely to find ongoing transmissions with longer distance, i.e., lower rate and longer duration).

We now characterize the process $Z(t)$, where $Z(t) = z$ if at time $t \geq 0$ the communication from $\boldsymbol{S}$ to $\boldsymbol{D}$ has z interfering nodes and $t = 0$ corresponds to the start of the handshake.

We denote with $R(t^*)$ the process representing the number of already active communications at time $t = 0$ that are still alive at time $t = t^*$. Ongoing communications have source–destination distance distribution $\mathcal{F}''_\delta(\delta^*)$, so that their average duration distribution and mean are

$$\mathcal{G}''(\tau) = \int_0^{R_{\max}} \mathcal{G}_\delta(\tau) f''_\delta(\delta^*) \mathrm{d}\delta^* \qquad \Omega'' = \int_0^{T_{\max}} \left[1 - \mathcal{G}''(\tau)\right] \mathrm{d}\tau. \tag{3.18}$$

Observing that $R(0) = Z(0)$, we model the number of ongoing interfering transmissions at time $t = 0$ with the long run distribution of a Poisson arrival distribution with parameter ν and lifetime distribution $\mathcal{G}''(\tau)$, that is also Poisson with parameter $\nu\Omega''$ [12], i.e.,

$$\mathcal{P}\left\{R(0) = z_0\right\} = \frac{(\nu\Omega'')^{z_0}\, e^{-\nu\Omega''}}{z_0!}, z_0 = 0, 1, 2, \ldots. \tag{3.19}$$

The probability that an active transmission at time $t = 0$ is still active at time τ is the probability that its residual life ω is greater than or equal to τ^*, i.e.,

$$\mathcal{P}\left\{\omega \geq \tau^*\right\} = 1 - \frac{1}{\Omega''} \int_0^{\tau^*} \left[1 - \mathcal{G}''(\tau)\right] \mathrm{d}\tau = 1 - \frac{\zeta''_{\tau^*} \tau^*}{\Omega''}, \tag{3.20}$$

where $\zeta''_{\tau^*} = \frac{1}{\tau^*} \int_0^{\tau^*} [1 - \mathcal{G}''(\tau)] \mathrm{d}\tau$. Observing that $R(t^*) \leq R(0)$, with $t^* > 0$, the probability that r of the $R(0)$ transmissions are still active at time t^* is then

$$\mathcal{P}\left\{R(t^*) = r \mid R(0) = z_0\right\} = \frac{z_0!}{r!(z_0 - r)!} \left(1 - \frac{\zeta''_{t^*} t^*}{\Omega''}\right)^r \left(\frac{\zeta''_{t^*} t^*}{\Omega''}\right)^{z_0 - r}. \tag{3.21}$$

However, during this transmission time, new transmissions can also start and end, contributing to the total process $Z(t)$. We denote with $N(t^*)$ the number of new transmissions started in $(0, t^*)$, and with $M(t^*)$ the process counting the number of those transmissions that are still active at time $t = t^*$. $M(t^*)$ has Poisson distribution with mean

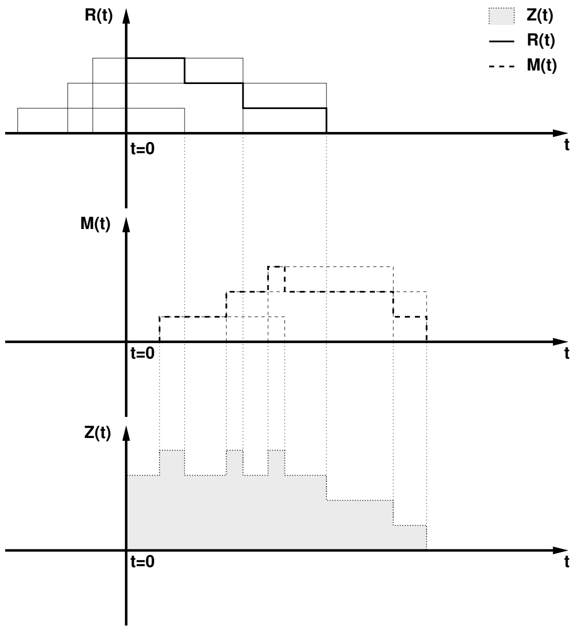


Figure 3.3. *Graphical representation of the arrival and departure process during a transmission. The processes that count the number of total, ongoing and new transmissions are denoted with dotted filled–gray, solid and dashed line, respectively.*

$\nu\zeta'_{t^*}t^* = \nu\int_0^{t^*}[1-\mathcal{G}'(\tau)]\mathrm{d}\tau$ [12], i.e.,

$$\mathcal{P}\left\{M(t^*)=m\right\} = \frac{(\nu\zeta'_{t^*}t^*)^m\, e^{-\nu\zeta'_{t^*}t^*}}{m!}, m=0,1,2,\dots \tag{3.22}$$

where $\mathcal{G}'(\tau^*) = \int_0^{R_{\max}}\mathcal{G}_\delta(\tau^*)f'_\delta(\delta^*)\mathrm{d}\delta^*$. Therefore, the distribution of the total number of active transmissions at time τ^*, conditioned on the number of ongoing communications at time $t=0$, is

$$\begin{aligned}\mathcal{P}\{Z(\tau^*)=z\,|\,Z(0)=z_0\} &= \sum_{r=0}^{\min(z_0,z)} \mathcal{P}\{R(\tau^*)=r\mid Z(0)=z_0\}\,\mathcal{P}\{M(\tau^*)=z-r\} \\ &= \sum_{r=0}^{\min(z,z_0)} \frac{z_0!}{r!(z_0-r)!}\left(1-\frac{\zeta''_{\tau^*}\tau^*}{\Omega''}\right)^r\left(\frac{\zeta''_{\tau^*}\tau^*}{\Omega''}\right)^{z_0-r}\frac{(\nu\zeta'_{\tau^*}\tau^*)^{(z-r)}e^{-\nu\zeta'_{\tau^*}\tau^*}}{(z-r)!}\end{aligned} \tag{3.23}$$

Fig. 3.3 depicts a graphical representation of the described processes. It is important to observe that the proposed analysis considers the average transmission length distributions, and it is therefore approximated in the sense that the correlation between the number of actual interfering nodes and the length distribution is ignored. In fact, given for instance a high number of interfering transmissions, their length would tend to be generally greater than in the presence of a lower number of users, due to the generally low SINR that the destinations of these communications might perceive at the start of the handshake. Our results have shown that this approximation is accurate.

3.2.2 SINR Distribution

To characterize the system performance we derive the average SINR distribution at time t, where $Z(t)=K$ is the number of transmitting users. We focus on the MF-LSIC case, since the MF case is straighforward. As in Section 3.1.1, we assume that the received powers, denoted with $\gamma_1, \ldots, \gamma_K$, are sorted in descending order.

We consider a Rayleigh block–fading channel model, so that at distance δ from a transmitter, the probability that the received power of the wanted signal at the destination, γ_s, is lower than γ^* is

$$\mathcal{J}_\delta(\gamma^*) = \mathcal{P}\left\{\gamma_s \leq \gamma^*\right\} = \int_0^{\gamma^*} \frac{1}{P_\mathrm{t}\delta^{-\alpha}} e^{-\frac{x}{P_\mathrm{t}\delta^{-\alpha}}} \mathrm{d}x = 1 - e^{\frac{\gamma^*}{P_\mathrm{t}\delta^{-\alpha}}}, \tag{3.24}$$

where α is the path–loss exponent. The received interference power of a single transmitting node distribution is

$$\mathcal{J}(\gamma^*) = \int_0^{R_\text{max}} \mathcal{J}_\delta(\gamma^*) f_\delta(\delta)\mathrm{d}\delta = 1 - \frac{2\left(\frac{\gamma^*}{P_\mathrm{t}}\right)^{-\frac{2}{\alpha}} \left(\Gamma\left(\frac{2}{\alpha}\right) - \Gamma\left(\frac{2}{\alpha}, \frac{\gamma^* R_\text{max}^\alpha}{P_\text{max}}\right)\right)}{\alpha R_\text{max}^2}, \tag{3.25}$$

where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ and $\Gamma(a,z) = \int_z^\infty t^{a-1}e^{-t}dt$ are the Gamma and the incomplete Gamma functions. We define the pdfs associated with $\mathcal{J}_\delta$ and $\mathcal{J}$ as $j_\delta(\gamma^*)=\mathrm{d}\mathcal{J}_\delta(\gamma^*)/\mathrm{d}\gamma^*$ and $j(\gamma^*) = \mathrm{d}\mathcal{J}(\gamma^*)/\mathrm{d}\gamma^*$, respectively.

The output SINR of the MF LSIC receiver is modeled with (3.2). Thus, considering a transmission from a source node $\boldsymbol{S}$ to a destination node $\boldsymbol{D}$, with $\mathcal{D}(\boldsymbol{S},\boldsymbol{D}) = \delta$, where the total number of interfering signals is K, the probability that the output SINR $S(t)$ is lower than or equal to S^*, given the received power ω_s^2 of the wanted signal, is

$$\begin{aligned}
\mathcal{X}^K(S^*,\delta) &= \int_0^\infty \sum_{m^*=1}^{K+1} \mathcal{P}\left\{S \leq S^* \mid m = m^*, \gamma_s\right\} \mathcal{P}\left\{m = m^* \mid \gamma_s\right\} j_\delta(\gamma_s)\mathrm{d}\gamma_s \\
&= \int_0^\infty \sum_{m^*=1}^{K+1} \mathcal{X}_{|\gamma_s}^{(K,m^*)}(S^*,\delta)\mathcal{P}\left\{m = m^* \mid \gamma_s\right\} j_\delta(\gamma_s)\mathrm{d}\gamma_s,
\end{aligned} \tag{3.26}$$

where m is the decoding stage. From (3.2), we get

$$\mathcal{X}_{|\gamma_s}^{(K,m)}(\psi,\delta) = \int_{\gamma_s}^\infty j(\gamma_1)\left(\int_{\gamma_s}^{\gamma_1} j(\gamma_2)\left(\ldots\int_{\gamma_s}^{\gamma_{m-2}} j(\gamma_{m-1}) Z_\psi^m(\boldsymbol{\gamma}_{m-1},\gamma_s)\mathrm{d}\gamma_{m-1}\ldots\right)\mathrm{d}\gamma_2\right)\mathrm{d}\gamma_1, \tag{3.27}$$

where $\boldsymbol{\gamma}_n$ denotes the vector $[\gamma_1,\ldots,\gamma_n]$. γ_s is equal to γ_m, since the wanted signal is decoded at stage m. $\mathcal{Z}_m^\psi(\boldsymbol{\gamma}_{m-1},\gamma_s)$ is defined as

$$\mathcal{Z}_m^\psi(\boldsymbol{\gamma}_{m-1},\gamma_s) = \int_0^{\min\left\{\gamma_s, \mathcal{S}_\psi^{m+1}(\boldsymbol{\gamma}_{m-1},\gamma_s)\right\}} j(\gamma_{m+1})\left(\ldots\int_0^{\min\left\{\gamma_{K-1}, \mathcal{S}_\psi^K(\boldsymbol{\gamma}_{K-1},\gamma_s)\right\}} j(\gamma_K)\mathrm{d}\gamma_K\ldots\right)\mathrm{d}\gamma_{m+1},$$

where

$$\mathcal{S}_{\psi}^{n}(\boldsymbol{\gamma}_{n-1}, \gamma_s) = \frac{\gamma_s}{\psi}\left(1 - \frac{1}{N}\right)^2 - \frac{\sigma^2}{N} - \sum_{j<m} \gamma_j \frac{N\left(\frac{j-1}{N^2}\right)}{\left(1 - \frac{1}{N}\right)^{m-1}} - \sum_{m<j<n} \gamma_j. \tag{3.28}$$

Now we evaluate the probability that the wanted signal is decoded at stage m^*, given the total number of transmitting nodes K and $\mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta$. Assuming that the received power of the wanted signal is γ_s, the probability that it is decoded at stage m^* is equal to

$$\mathcal{P}\left\{m = m^* \mid \gamma_s\right\} = \frac{(K-1)!}{(m^*-1)!(K-m^*)!}\left(1 - \mathcal{J}(\gamma_s)\right)^{(m^*-1)}\left(\mathcal{J}(\gamma_s)\right)^{(K-m^*)}, \tag{3.29}$$

These integrals can be computed through numerical integration.

3.3 Recursive Performance Analysis

The distributions derived in the previous Section are required for assessing the network performance. To this end we set up an algorithm that recursively computes the system interference and failure distributions.

To keep the problem tractable we divide the time axis in slots of duration T_{S}. We assume that T_{S} is within the channel coherence time, so that fading coefficients remain constant during a slot. Moreover, we assume that users can start transmissions only at slots boundaries, and that the duration of each transmission is a multiple of T_{S}, which is reasonable if transmitters have only a finite set of rates. Note that in this setting the number of interfering nodes during each slot does not change. For the sake of simplicity, also the handshake duration is set to a multiple of the slot duration. The data packet transmission duration in slots is $N = \left\lceil \frac{L}{R_x T_{\mathrm{S}}} \right\rceil$, where R_x is the rate prescribed by the used protocol, and $\lceil \cdot \rceil$ is the ceiling operator.

The recursive algorithm takes as input the estimated distributions Ψ_δ and $\mathcal{G}_\delta(\tau)$, and the per node transmission arrival rate λ. Given Ψ_δ and $\mathcal{G}_\delta(\tau)$, the algorithm computes the distribution of the number of interfering transmissions at the beginning and during the transmission as described in Section 3.2.1. Through Montecarlo trials the algorithm produces a further estimate of these distributions and collects the performance metrics described in 3.3.1. In particular, for a fixed distance δ between the source and the destination, the number of slots N the transmission lasts is a function of the initial number of interferers. Thus, in the ARQ case the distribution of the length for the next iteration is given by

$$\begin{aligned}
\tilde{\mathcal{G}}_\delta(N^* T_{\mathrm{S}}) &= \mathcal{P}\left\{N \leq N^* \mid \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\} = \\
&= \sum_{z_0=1}^{\infty} \mathcal{P}\left\{S(0) \geq 2^{\frac{L}{WN^* T_{\mathrm{S}}}} - 1 \mid \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta, Z(0) = z_0\right\} \mathcal{P}\left\{Z(0) = z_0\right\} \\
&= 1 - \sum_{z_0=1}^{\infty} \mathcal{X}^{z_0}\left(2^{\frac{L}{WN^* T_{\mathrm{S}}}} - 1, \delta\right) \frac{(\nu \Omega'')^{z_0}\, e^{-\nu \Omega''}}{z_0!},
\end{aligned} \tag{3.30}$$

where N is the number of slots where the source transmits. Note that the calculation of $\tilde{\mathcal{G}}_\delta(N^* T_{\mathrm{S}})$ is based on its estimate at the previous algorithm step. In the type I HARQ case the rate is scaled by a factor ρ. As to the failure probability, for the ARQ and type I HARQ this corresponds to the event

$$\xi = \left\{ WT_{\mathrm{S}} \sum_{u=1}^{N} \log_2 \left(1 + S(uT_{\mathrm{S}})\right) < L \right\}. \tag{3.31}$$

$\tilde{\Psi}_\delta = \mathcal{P}\left\{\xi \mid \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\}$ is evaluated through the distribution of the number of users and the SINR distribution via Montecarlo integration. In the type II HARQ case, the distribution of the transmission length is

$$\begin{aligned}\tilde{\mathcal{G}}_\delta(N^* T_{\mathrm{S}}) &= \mathcal{P}\left\{N' \le N^* \mid \bar{\xi}', \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\} \mathcal{P}\left\{\bar{\xi}' \mid \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\} \\ &+ \mathcal{P}\left\{N' + N'' \le N^* \mid \xi', \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\} \mathcal{P}\left\{\xi' \mid \mathcal{D}(\boldsymbol{S}, \boldsymbol{D}) = \delta\right\}, \end{aligned} \tag{3.32}$$

where N' and N'' are the lengths in slots of the first and second phase, and ξ' and $\bar{\xi}'$ represent the failure and success events in the first phase, respectively. As in the previous case, the various probabilities can be conditioned to the initial number of interferers and summed. Note that the distribution of the length of the second phase depends on the SINRs perceived in the first phase and the SINR of the last slot, i.e., $S(N'T_{\mathrm{S}})$.

Note also that the retransmission process changes not only the destination distance distribution of the interfering transmissions, but also the overall transmission arrival rate. The input arrival rate $\tilde{\lambda}$ for the next algorithm iteration is

$$\tilde{\lambda} = \lambda \int_0^{R_{\max}} \Delta_\delta f_\delta(\delta) \mathrm{d}\delta. \tag{3.33}$$

In the first iteration, the failure probability is set to zero. Observe that in this case $\mathcal{F}''_\delta(\delta^*) = \mathcal{F}'_\delta(\delta^*) = \mathcal{F}_\delta(\delta^*)$. The initial distribution of the source transmission length $\mathcal{G}_\delta(\tau)$ is evaluated for a number of interfering transmissions that is distributed according to a Poisson process of rate ν. With this distribution the evaluation of the initial failure probability is then performed.

3.3.1 Performance Metrics

Through the presented analysis and the distributions defined in Section 3.2.1 we obtain some metrics that are significant for characterizing the network performance.

The failure probability of the packet delivery, taking into account the various retransmissions, and conditioned to the destination distance $\delta = \delta^*$, is

$$\tilde{\Gamma}_{\delta^*} = 1 - \sum_{a=0}^{F-1} \left(\tilde{\Psi}_{\delta^*}\right)^a \left(1 - \tilde{\Psi}_{\delta^*}\right) = \left[\tilde{\Psi}_{\delta^*}\right]^F, \tag{3.34}$$

and the average number of transmissions is $\tilde{\Delta}_{\delta^*} = (1 - \tilde{\Psi}^F_{\delta^*})/(1 - \tilde{\Psi}_{\delta^*})$. We refer to their respective values averaged over the destination distance as $\tilde{\Gamma}$ and $\tilde{\Delta}$.

The overall throughput in [bps/Hz] achieved in the considered area is then

$$\mathcal{R} = \nu L \int_0^{R_{\max}} \frac{1-\tilde{\Gamma}_\delta}{\Omega_\delta \tilde{\Delta}_\delta} f_\delta(\delta) \mathrm{d}\delta. \tag{3.35}$$

We also compute the average number of active interfering transmissions per slot as

$$\mathcal{U} = \sum_{z_0=0}^{\infty} \frac{\sum_{s=0}^{N^*} \sum_{k=0}^{\infty} k \mathcal{P}\left\{Z(s\Delta_T) = k \mid Z(0) = z_0\right\}}{N^*} \mathcal{P}\left\{N = N^* \mid Z(0) = z_0\right\} \mathcal{P}\left\{Z(0) = z_0\right\}.$$

3.4 Results

In this section we present and discuss the results obtained with the system analysis developed in the previous sections. First, we compare the analytically obtained performance with the results of simulations that implement all the details of the HARQ process at both the source node and the interfering nodes. To avoid border effects, we consider a circular area of radius $QR_{\max}$, $Q \geq 1$, around the destination of which we collect the performance. This is useful to get a realistic transmission length distribution at the interfering nodes, that are in turn interfered by other transmissions. However, the interfering nodes at distance greater than $R_{\max}$ from the various destinations, including those of the interfering nodes, are ignored in the received SINR computation. Simulations are computationally much heavier than the analysis, due to the need to keep track of the status of all the ongoing communications (including those in backoff), and soon become infeasible as the number of communications increases, i.e., for high values of λ, ρ or F. In Table 3.1, the values of the parameters used in both analysis and simulations are summarized.

Figs. 3.4, 3.5 and 3.6 give some examples of comparison between analysis and simulations. Many more cases have been run, and the match was observed to be fairly good in most cases. Fig. 3.4 compares the overall analytical throughput achieved in the considered area, computed as in (3.35), with the throughput obtained through simulations for the MF and MF LSIC cases with an ARQ scheme with two retransmissions as a function of the per node packet arrival rate λ. It is possible to observe that the analysis shows a good match with the simulations, especially in the LSIC case. The MF case is more sensitive to the approximation of the interfering transmission behavior with the averaged statistics. In fact, the MF receiver has a lower resilience to interference than the LSIC receiver and then the correlation between the number and the duration of the interfering transmissions is greater.

Fig. 3.5 shows the average duration of a transmission as a function of λ for the LSIC receiver with the type I HARQ scheme for $\rho = 1, 1/2, 1/3$. As expected, the lower the coding rate, the higher the transmission length. Note that the ratio of the average durations for $\rho = 1$ and $\rho = 1/2$ is not necessarily equal to 2. In fact, besides the coding gain, the duration depends on the perceived SINR and, thus, on the interference in the network. This results from the tradeoff between the single transmission failure probability, that is the retransmission probability in the case $F = 2$, and the single transmission length. Fig. 3.6 depicts the

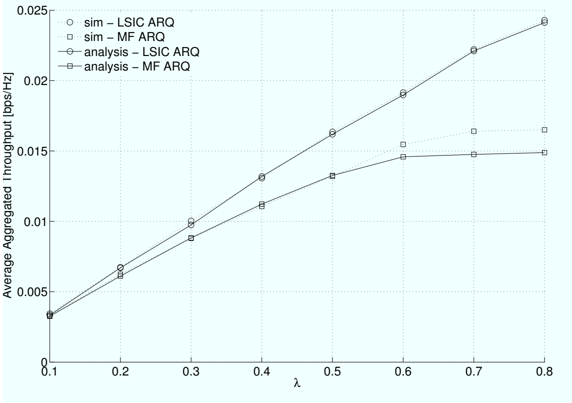


Figure 3.4. *Average throughput as a function of λ for the MF and MF LSIC cases, ARQ scheme, $F = 2$.*

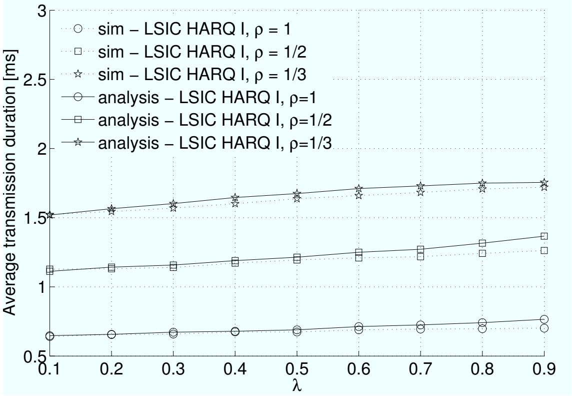


Figure 3.5. *Average transmission duration as a function of λ, type I HARQ scheme, $F = 2$.*

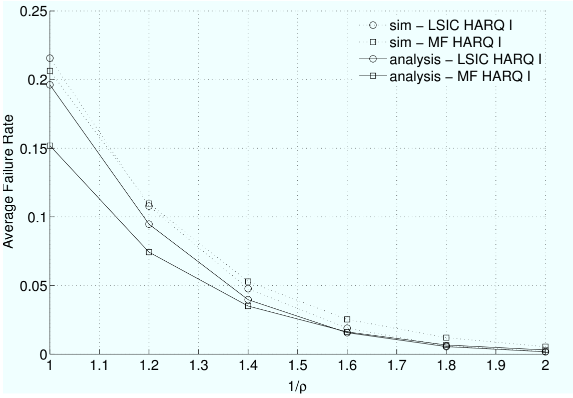


Figure 3.6. *Average failure probability as a function of $1/\rho$ for the MF and MF LSIC cases, type I HARQ scheme, $F = 2$, $\lambda = 0.4$.*

average delivery failure rate, including retransmissions, for the same cases considered in the previous plot, as a function of the coding rate ρ. The proposed analysis is slightly less accurate than for the throughput for high failure probabilities.

In the following, we present analytical results comparing the considered metrics for the various proposed schemes. In the following we set $\eta' = \eta'' = \eta$, leaving for future investigations the optimization of the performance that may come from a differentiation of the values for η' and η''. Figs. 3.7, 3.8 and 3.9 show respectively the average throughput, the number of interfering transmissions and the failure rate as a function of the node density μ achieved by the various proposed schemes for various values of F. As a first observation, for all the considered error control policies if the number of retransmissions allowed is increased the interference generated by the greater birth rate decreases the average transmission rate, and thus degrades the throughput. This effect is less noticeable in type I and II HARQ, where the higher probability that a transmission achieves a success reduces the retransmission

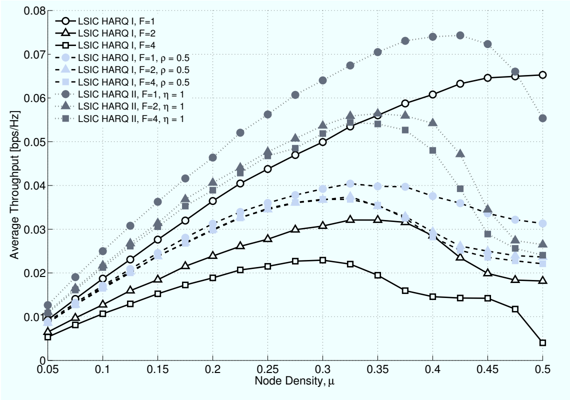


Figure 3.7. *Average throughput $\mathcal{R}$ as a function of the node density μ and the various proposed schemes for the MF LSIC case.*

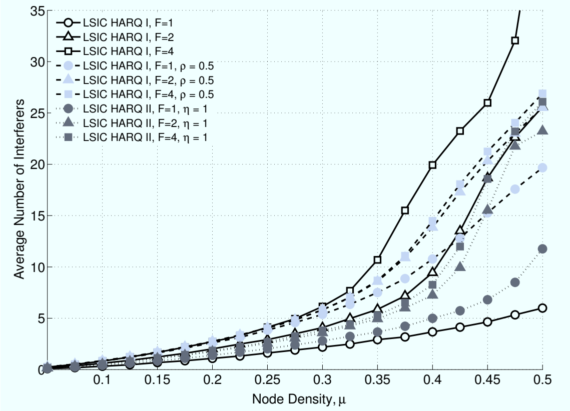


Figure 3.8. *Average number of interfering nodes $\mathcal{U}$ as a function of the node density μ and the various proposed schemes for the MF LSIC case.*

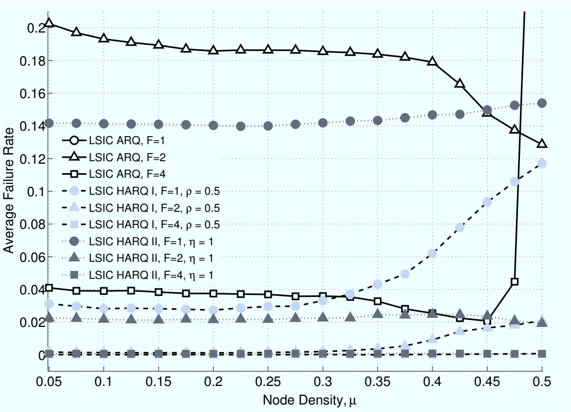


Figure 3.9. *Average communication failure rate Γ as a function of the node density μ and the various proposed schemes for the MF LSIC case, (the failure rate for the ARQ scheme with $F = 1$ is greater than 0.4).*

probability and then the increase in birth rate. Moreover, for small network load a density increase results in a throughput improvement, while above a scheme–dependent threshold the throughput decreases as the density is further increased. This is due to the tradeoff between the gain due to a higher number of simultaneously deployed transmissions, and their average duration and failure probability.

Type II HARQ appears to be the best choice for the considered density range, as it achieves a good throughput preserving communication reliability even for a low number of allowed retransmissions. In fact, while type I HARQ relies on long transmissions to keep the failure probability low, and pure HARQ incurs too many failures, type II HARQ provides a good system balance. This is due to its adaptability to highly varying channel conditions. However, as the density increases, the best choice in terms of throughput probably becomes

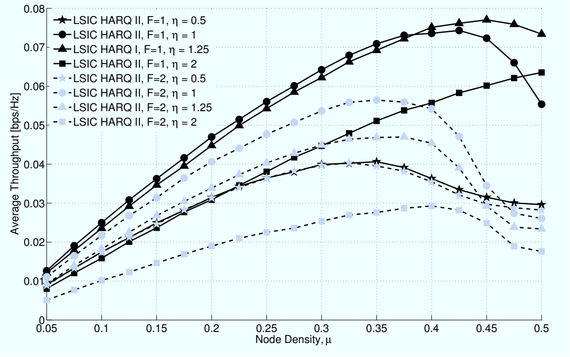


Figure 3.10. *Average throughput as a function of the node density μ for the type II HARQ scheme, $F = 2$.*

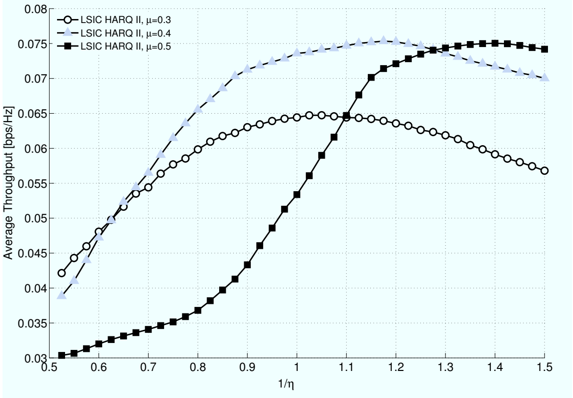


Figure 3.11. *Average throughput as a function of η for the type II HARQ scheme, $F = 1$.*

early packet discarding and short transmission to reduce receivers' load. For very high density, as interference saturates the network, strong coding, and thus longer transmission, has a lower effect on throughput, so that it can be a good solution.

An interesting observation is that that while the failure probability generally increases as the density increases for the type I and II HARQ, the pure ARQ scheme decreases slightly before the entire network collapses. This is due to the longer duration of the transmissions as the network load increases, that provides an increased channel correlation in the ARQ schemes, that have a high failure probability and hence a high birth rate. This effect is negligible in schemes with lower retransmission probability and intrinsically higher duration, such as type I and II HARQ.

Fig. 3.10 depicts the throughput for different values of the density as a function of η for a single transmission. It is possible to observe that for average values of the node density, the throughput has a maximum for values close to $\eta = 1$, and the performance quickly degrades as the coding becomes stronger. For high densities, in which the interference load is higher, a greater load due to stronger encoding heavily affects the performance in terms of throughput, so that the maximum is achieved for more aggressive choices of η.

Fig. 3.11 shows the throughput vs. node density for type II HARQ and various values of η for $F = 1$and 2. In this case it is important to observe that too aggressive or too conservative choices of the coding parameter can affect throughput. It is interesting to observe that while for $F = 1$ a too conservative choice of η, such as $\eta = 0.5$, heavily affects throughput, due to the average higher transmission duration that outweighs the improvement in terms of failure probability, for $F = 2$ and high values of the nodes density the scheme with $\eta = 0.5$ achieves the best performance. This is due to the reduced retransmission probability, that increases channel predictability. Moreover, this scheme accumulates the gain of two long transmissions, while for the other choices of η the receiver has a higher probability of discarding what was already received, relying on a further transmission. We remark that the interaction between the interference, the transmission rate and the reliability in MUD net-

Parameter	Value
Node density μ	0.1 nodes/m^2
Arrival rate λ	0.4 [pkt/s]
Maximum range $R_{\max}$	100m
Available bandwidth W	$10^8/N$ Hz
Slot duration T_S	$2 * 10^{-5}$ s
Transmission power P_t/σ^2	43dB
Path loss exponent α	2
Spreading factor N	16
Uncoded packet length L	4096bits

Table 3.1. *Table of Parameters*

works is rather complex and involved. Therefore, the scheme the system relies on for packet delivery is critical for the achieved performance. Generally, too low an encoding rate results in an increased transmission length, while too aggressive a transmission strategy incurs excessive additional retransmissions. Type II HARQ schemes, if accurately set, appear to be a good solution for both efficiently adapting the transmisson rate to the channel conditions and preserving the system reliability.

The initial results presented in this section highlight some interesting trade-offs that arise when combining HARQ, MUD, and multiple access in ad hoc networks. We believe that these behaviors, that are observed in this paper for the first time, deserve a deeper investigation, and can be expected to reveal interesting insights and to lead to strategies for the optimization of the network performance as a function of the lower layer design choices.

3.5 Conclusions

References

[1] K. Lai and J. J.Shynk, "Analysis of the Linear SIC for DS/CDMA Signals with random Spreading," *IEEE Trans. Signal Processing*, vol. 52, no. 12, pp. 3417–3428, Dec. 2004.

[2] K. Lai and J. J. Shynk, "Performance evaluation of Generalized Linear SIC for DS/CDMA signals," *IEEE Trans. Signal Processing*, vol. 51, no. 6, pp. 1604–1614, Jun. 2003.

[3] P. B. Rapajic and B. S. Vucetic, "Adaptive receiver structure for asynchronous CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 685–697, May 1994.

[4] S. L. Miller, "An adaptive direct–sequence code–division multiple–access receiver for multiuser interference rejection," *IEEE Trans. Commun.*, vol. 43, pp. 1746–1755, Feb./Mar./Apr. 1995.

[5] Y. Cho and J. H. Lee, "Analysis of an adaptive SIC for near–far resistant DS-CDMA," *IEEE Trans. Commun.*, vol. 46, no. 11, pp. 1429–1432, Nov. 1998.

[6] C. Schlegel, Z. Shi, and M. Burnashev, "Optimal power/rate allocation and code selection for iterative joint detection of coded random CDMA," *IEEE Trans. on Inform. Theory, to appear*.

[7] P. Alexander, A. Grant, and M. Reed, "Iterative detection in code-division multiple-access with error control coding," *Eur. Trans. Telecommun.*, vol. 9, no. 5, pp. 419–426, Sep. 1998.

[8] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. Schlegel, "Iterative multiuser interference reduction: Turbo CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1008–1014, Jul. 1999.

[9] P. H. Tan and L. K. Rasmussen, “Multiuser detection in CDMA – a comparison of relaxations, exact, and heuristic search methods,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1802–1809, Sep. 2004.

[10] Kiran and D. N. C. Tse, “Effective bandwidth and effective interference for linear multiuser receivers in asynchronous channels,” *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1426–1447, Jul. 2000.

[11] L. K. Rasmussen, T. J. Lim, and A.-L. Johansson, “A matrix–algebraic approach to successive interference cancellation in CDMA,” *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 145–151, Jan. 2000.

[12] S. Karlin and H. M. Taylor, *An introduction to stochastic modeling*. San Diego: Academic Press, 1998.

Chapter 4

Cooperation

Contents

4.1 Introduction

In the previous Chapter, we discussed how error control and rate control interact with interference distribution. From the presented results, it is clear how important is efficiency in asynchronous ad hoc networks with simultaneous access. In fact, more efficient communications result in lower interference at the receivers. We also highlighted the performance gain provided by adaptive error control schemes, such as hybrid ARQ. Here, we introduce cooperation in the network in order to improve adaptability and efficiency of communications.

Cooperation among nodes of a wireless ad hoc network has been recently investigated for its potential to provide spatial diversity by implementing a distributed antenna array. Early examples of cooperation are fixed relaying [1, 2] and selective relaying [3], that have been further enhanced by letting nodes cooperate only when they experience good channel conditions through *opportunistic* techniques [4]. Opportunistic routing [5] exploits the broadcast nature of the wireless channel to communicate through good links, possibly with power and rate adaptation, as in multiuser diversity forwarding (see [6, 7] and references therein). The choice of cooperative nodes then involves also scheduling issues, and various solutions have been proposed, including geographic random forwarding (GeRaF) [8] and opportunistic scheduling [9, 10].

We focus on coded cooperation schemes, that essentially realize a distributed hybrid ARQ error control policy. We believe that cooperation can considerably improve performance in our scenario. Moreover, we will show that practical implementations of cooperation schemes benefits the possibility for simultaneous communications. We divide our work on cooperative issues into three logical parts

- in the first part we present some preliminary observation of a simple coded cooperation scheme based on both analysis and simulation;
- in the second part we present an interesting analysis of interference load in the network when cooperation is enabled;
- in the last part we present the design of a complete cooperative network, carried out via analysis of a simplified scenario in order to gain some insight on the performance of the distributed HARQ scheme and extensive simulation of the entire network to discuss the interaction of cooperation with packet queueing and traffic.[1]

Technical contributions on cooperation generally focus on the analysis of simple and limited topologies and cases. We believe that the contribution of our work is to provide a wide understanding of issues arising when using cooperation in a network.

The existing literature has not thoroughly addressed the dynamic behavior of a network, where cooperation is conditioned upon the availability of nodes not involved in any other transmission, which in turn depends on the adopted medium access control (MAC) and routing techniques and on the traffic statistics. In some cases, e.g. [2], transmit and cooperative phases are disciplined by time division multiple access (TDMA), which may lead to spectrum inefficiency. Moreover, at a network level a further protocol should handle the case of a failure in both the initial and the cooperative phases. Another example of MAC for cooperation is described in [11], where code division multiple access (CDMA) allows simultaneous transmissions by multiple users, and TDMA is used to discipline cooperation, thus having the same limitations as the previous scheme. Also in this case cooperative nodes are made always available by considering full-duplex terminals, which are difficult to realize in

[1] We are extending the network protocol presented in this part to include cooperative routing and opportunistic routing.

ad hoc networks. In most works, the issues of the availability of cooperative nodes and the bandwidth cost of a cooperative action are often ignored. In addition, the analytical investigation of cooperative mechanisms is typically limited to very simple networks with only three or four nodes [12], a scenario that lends itself to theoretical approaches but may fail to reveal more interesting behaviors at the network level.

A further problem of cooperation in a complex network is node synchronization. For example, the use of space-time block codes (STBC) [13,14] for cooperation [3,15,16] typically requires symbol synchronization among cooperative nodes, which may be very expensive or even infeasible for non-infrastructured networks [16]. Moreover, STBC are designed for a specific number of transmit antennas and some signaling overhead is needed in order to coordinate or select the cooperative nodes.

This work has been done in collaboration with Stefano Tomasin, whose contribution is mainly represented by the interference model and analysis in Section 4.3 and the Markov model of a single communication of Section 4.4.

The reference papers to my work on cooperation are [J2ml, J6ml, J8ml, J09, C06ml, C10ml, C11ml, C12ml, C14ml].

4.2 Preliminary Investigation

In this Section, we present some results based on analysis and simulation useful to understand some fundamental tradeoffs between interference, resource allocationo and cooperation.

We consider a distributed Type II hybrid automatic retransmission request (HARQ) system, where nodes cooperatively send to their intended destinations fragments of the codeword obtained by encoding the original information frame. Communications are assumed to be half–duplex, i.e., a node can either transmit or receive at a given time. We investigate the performance of various HARQ schemes, both cooperative and non cooperative, in a symmetric network. Symmetry lies in the fact that the channels between every pair of nodes have the same fading statistics. Under this assumption, it is possible to gain insights on the diversity provided by the compared schemes. Moreover, in our setting a fixed number of nodes are allowed to access the channel at the same time. The aim of this part of the Chapter is to provide a thorough discussion of how to manage channel resources and interference, in order to effectively achieve diversity and coding gain. Moreover, we investigate the effect of channel correlation on the protocols performance. While the analysis is carried out under symplifying assumptions on channel correlation, in simulation we consider a model where the channel samples of adjacent blocks are correlated.

The remainder of this part of the Chapter is as follows. In Sections 4.2.1 and 4.2.2 we describe the system and the various HARQ protocols considered, respectively. In Section 4.2.3 we model the HARQ protocols with semi–Markov processes, whose transition probabilities are derived from the outage probabilities of the scheme and packet arrivals are modeled

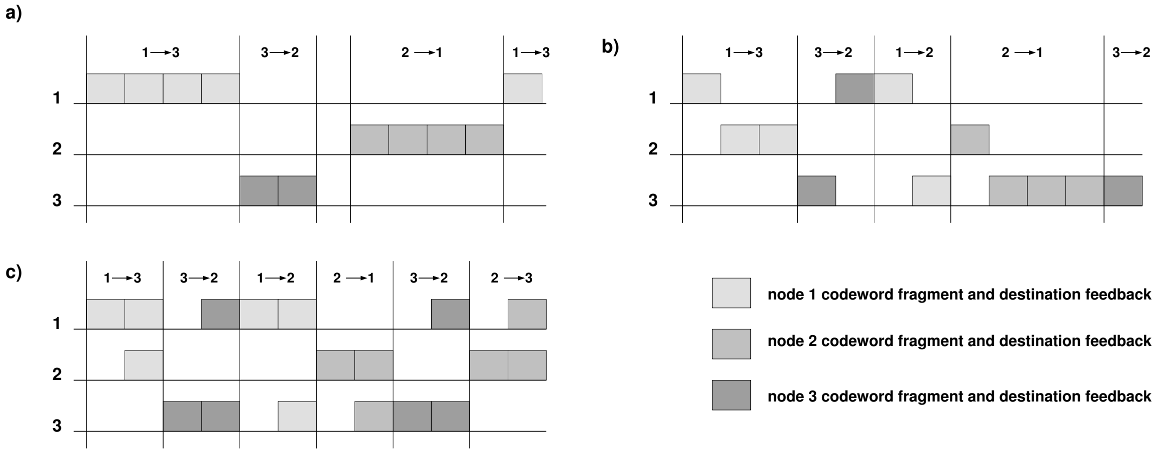

a)
1→3
3→2
2→1
1→3
1
2
3
b)
1→3
3→2
1→2
2→1
3→2
1
2
3
c)
1→3
3→2
1→2
2→1
3→2
2→3
1
2
3
node 1 codeword fragment and destination feedback
node 2 codeword fragment and destination feedback
node 3 codeword fragment and destination feedback

Figure 4.1. *Examples of communication protocols with three nodes for the single access case: a) nC b) CnS c) CS. Note that with three nodes MCnS behaves as CnS and MCS as CS.*

as a Poisson process, in order to gain fundamental insights into the system behavior. Section 4.2.4 discusses the protocol performance derived through the analytical model as well as complete system simulation.

4.2.1 System Description

We assume a network of M nodes, where each node stores the information frames in a first-in first-out (FIFO) finite queue. Frame arrivals in the node queue are modeled with a Poisson process of parameter λ frames per second per node. Each information frame is to be delivered to a destination randomly chosen among the other $M-1$ nodes. We consider half–duplex communications, i.e., for a single node it is not possible to transmit and receive simultaneously. The channels between every pair of nodes have the same fading statistics. We divide time in slots, each containing a phase, i.e., a single HARQ packet and feedback transmission.

HARQ protocol

A Type II HARQ protocol is used for transmission, and is described in general in this Section, whereas the specific details of the implementation considered are presented in Section 4.2.2.

Nodes encode each information frame with a forward error correcting code obtaining an L–bit codeword, $\boldsymbol{W}$, that is split into κ HARQ packets of L/κ bits. We consider a capacity–achieving code, i.e., for sufficiently long codewords the error probability vanishes when the link capacity is larger than the transmission rate R.

Nodes are allowed to perform a maximum of M_{tx} delivery attempts for a frame (in the following denoted as frame transmissions), each comprising at most M_{ph} phases. A single phase includes the transmission of an HARQ packet by the source, containing a fragment of $\boldsymbol{W}$, and of a feedback packet in which the destination reports whether or not it correctly decoded the frame. Upon reception of a negative feedback, a new HARQ packet is transmitted, followed by a further feedback packet. A failure reported at the M_{ph}–th phase causes the source to dismiss the attempt, and to schedule a new frame transmission after a backoff period B. The value of B is randomly chosen in the interval $[0, 2^{N_{\text{Fails}}} b_w]$, where N_{Fails} is the minimum between the number of consecutive failures and a fixed parameter Max_{Fails}, and b_w is the initial backoff window.

Note that at each phase of the same frame transmission, the destination collects and combines all the received codeword fragments, in order to achieve a higher coding gain, while it discards the received HARQ packets at the end of a frame transmission, i.e., each frame transmission is assumed to be independent with respect to the previous ones. At each frame transmission success, the value of N_{Fails} is decreased by one. The frame is discarded after M_{tx} unsuccessful frame transmissions. Please note that it is not required that all HARQ packets be transmitted by the source. In fact, through the use of cooperative protocols, the HARQ packets may be transmitted by other nodes that decoded and re-encoded the source's frame.

MAC protocol

We define various policies that rule the channel access. In the single access (SA) scheme, a single frame transmission per slot is allowed to take place. The constrained access (CA) scheme limits to K the number of users per slot that can access the channel.

Since our focus is on the transmission effectiveness of the various protocols, we assume idealized network access, so that no more than K transmissions can occur in each slot in the CA scheme, and no more than one frame transmission per slot can take place in the SA scheme. If the number of nodes that simultaneously want to access the channel is higher than the limit prescribed by the access policy, the transmitting nodes are selected at random. If the node denied access is a source, it reschedules a new access attempt with the same policy described for transmission failures. If it is a cooperator, then cooperation is dismissed. Nodes that already performed phases belonging to the same frame transmission, or that are cooperating with an ongoing frame transmission, are assumed to have higher access priority with respect to new frame transmissions. A realistic implementation of the access scheme may rely on contention slots and carrier sensing and is not addressed in this paper. Furthermore, in this initial study we do not consider errors in the feedback. Investigation of the effects of these idealized assumptions is left for future research.

4.2.2 Protocols Description

Non–Cooperative Protocol (nC)

With the non–cooperative (nC) protocol, the whole HARQ process is performed by the source node.

Fig. 4.1.a shows an example of nC communications for a three node network. Node 1 accesses the channel and transmits the first HARQ packet to Node 3, that reports a failure. Node 1 transmits the second HARQ packet, with a further negative feedback. After M_{ph}, with $M_{\text{ph}} = 4$ in the example, Node 1 reschedules a new frame transmission. In the following two slots, Node 3 sends two HARQ packets to Node 2, obtaining a negative and a positive feedback respectively. After an empty slot, due for instance to backoff or empty FIFO queues, Node 2 performs a four–phase frame transmission, followed by the second frame transmission of the initial communication between Node 1 and Node 3.

We remark that the phases belonging to the same frame transmission are performed uninterruptedly, as the access of ongoing transmissions is assumed to be prioritized with respect to the first phase of other transmissions.

Non Simultaneous Cooperative Protocol (CnS)

In the non simultaneous cooperative (CnS) protocol, after a negative feedback at one of the first $M_{\text{ph}} - 1$ phases, a node that correctly decodes the source's frame is enabled to reencode it and perform the following phases until the maximum number of phases is reached or a positive acknowledgment is sent by the destination. If none of the idle nodes correctly decodes the source frame, the following phase is performed by the source. If multiple nodes correctly decode the source frame, a node at random is selected for cooperation. Hence, in the transmission of a single information frame, at most two nodes can cooperate and they must transmit in different phases (non-simultaneous transmission).

We assume that cooperative transmissions enjoy the same increased priority as source transmissions related to the same information frame. During the cooperative phases associated with its transmission, the source remains idle and waits for the acknowledgement from the destination. For instance, in a CA scheme with $K = 2$, if a node is expected to cooperate, only another frame transmission is allowed to start. Moreover, note that the first phase of further frame transmissions is always performed by the source, so that cooperating nodes are not required to store the information frames in their queues across different frame transmissions.

An example of communication exchange under this scheme is provided in Fig. 4.1.b. In this case, after the failure at the first phase of the frame transmission from Node 1 to Node 3, Node 2, that is assumed to correctly decode the frame, transmits the following HARQ packets, achieving a success at the third phase. In the following slots, after the failure of the first phase of the frame transmission between Node 3 and Node 1, Node 2 transmits the second HARQ packet. Thanks to the success at the previous frame transmission, Node 1 is able to access again the channel, performing a frame transmission to Node 2. The following frame transmissions are carried out analogously.

Multiple Non Simultaneous Cooperative Protocol (MCnS)

The multiple non simultaneous cooperative (MCnS) protocol is a cooperative protocol similar to CnS, with the exception that the choice of the cooperating node is renewed at each phase. Thus, if a negative feedback is sent at a phase performed by a cooperating node and another idle node correctly decodes the information frame, the latter performs the following phase. With this protocol, the number of cooperating nodes for a single information frame transmission is at most equal to the overall number of phases. Still, at each time only one node can transmit an HARQ packet associated with a frame transmission. This scheme achieves a higher diversity order with respect to CnS, since the chances of getting a good channel are increased at each HARQ transmission.

Simultaneous Cooperative Protocol (CS)

The simultaneous cooperative (CS) protocol is a cooperative protocol where, upon availability of a cooperating node from the second phase, both the source and the cooperator transmit simultaneously. Still, as in CnS, only one cooperator is selected for transmission.

In the SA scheme the cooperator is always allowed to access the channel, even though the source is transmitting. As stated in the previous section, both cooperative and source retransmissions in phases belonging to ongoing frame transmissions are prioritized. Thus, the CA scheme with $K=1$ reduces to the CnS protocol. In CA with $K=2$ the presence of a cooperator prevents transmissions by other sources.

Fig. 4.1.c reports an example of operation of this protocol. Note that in this case the source and the cooperating node transmissions are simultaneous.

Multiple Simultaneous Cooperative Protocol (MCS)

The multiple simultaneous cooperative (MCS) protocol is an extension of CS where the cooperator may be changed at each phase as in MCnS, and access is managed as in CS.

4.2.3 System Analysis

In this Section we construct semi–Markov models for the various protocols, with the aim of evaluating relevant system performance metrics, such as throughput, efficiency, success probability, and average number of phases per frame transmission. In the following, we consider only the SA scheme, observing that the model for the CA scheme only requires a larger number of states and a more convoluted analysis, while from a conceptual point of view it is a straightforward extension of the proposed analytical model.

Outage Characterization

Consider a communication from node i to node j, in which i transmits part of the codeword $\boldsymbol{W}_{ij}$ to j, and let us focus on the P–th phase. During this transmission, other communications may take place simultaneously. We indicate with $\boldsymbol{I}_p$ the set of indices of nodes interfering with i during the generic p–th phase, and we define $\mathbb{I} = \{\boldsymbol{I}_p\}_{p=1,\ldots,P}$. Note that for the SA scheme $\mathbb{I}$ is empty. We also define the vector of the signal to noise ratios that characterize the links between every node of the network and the destination at the p–th phase as $\boldsymbol{C}_p^j = \left(c_p^{1j}, \ldots, c_p^{Mj}\right)$. The outage event conditioned on the interference is defined as:

$$\phi^{\{i,j\}}(P, \mathbb{I}) = \left\{ \alpha \sum_{p=1}^{P} \log_2 \left(1 + \frac{c_p^{ij}}{1 + \sum_{z \in \boldsymbol{I}_p} c_p^{zj}} \right) < R \right\}, \tag{4.1}$$

where $\alpha = 1/\kappa$ is the reciprocal of the number of total HARQ packets associated with a frame, as described is Section 4.2.1. The above equation can be generalized to the case of multiple users transmitting different parts of the same codeword to the intended destination. Define $\mathbb{T} = \{\boldsymbol{T}_p\}_{p=1,\ldots,P}$, where $\boldsymbol{T}_p$ is the set of the indices of cooperating nodes at the p–th phase, possibly including node i and with $j \notin \boldsymbol{T}_p$. Then the outage event, conditioned on the interference and the set of cooperators, is

$$\phi^{\{i,j\}}(P, \mathbb{I}, \mathbb{T}) = \left\{ \alpha \sum_{p=1}^{P} \sum_{n \in \boldsymbol{T_p}} \log \left(1 + \frac{c_p^{nj}}{1 + \sum\limits_{z \in \boldsymbol{I}_p} c_p^{zj} + \sum\limits_{t \in \boldsymbol{T}_p \setminus \{n\}} c_p^{tj}} \right) < R \right\} \tag{4.2}$$

We specialize the derivation of the outage probabilities to the case of identically distributed fading variables and to symmetric network conditions, i.e., $\{c_p^{nj}\}$ have all the same statistical description, according to the square norm of Rayleigh fading with probability density function

$$p(c) = \frac{1}{\gamma} e^{-\frac{c}{\gamma}}, \tag{4.3}$$

where γ is the average SNR for all links.

Chain Construction and Performance Analysis

For the analytical model, we make the simplifying assumptions that when a frame arrives at a node in a slot, it is either served in the next slot or randomly rescheduled, and that the Poisson arrival process rate includes the rescheduled attempts. With this assumption, we can neglect the correlation between transmission successes, transmission times, and traffic levels. Moreover, we assume that during all phases of a frame transmission the fading coefficients remain constant, i.e.,

$$\boldsymbol{C}_p^j = \boldsymbol{C}^j, \quad p \leq M_{\text{ph}}. \tag{4.4}$$

In the simulations, this assumption will be removed and more comprehensive results will be discussed.

As stated before, the various protocols are modeled with semi-Markov processes. Each protocol is characterized by a state space $\mathbb{S}$, where each state represents a different transmission condition. The number and the characterization of the states depend on the protocol features. Given that a single frame transmission is allowed to take place at a time, states are characterized by

- the phase index P;
- the number of phases characterized by a cooperating node P_c, with $P_c < P$;
- the phase indices at which the cooperating node is changed, denoted with $\boldsymbol{Q}$, with $\boldsymbol{Q} \subset \{1, \dots, P\}$, where $\boldsymbol{Q} = \emptyset$ denotes no cooperating node changes;

Observe that MCS and MCnS, although being described by analogous chains, have different outage probabilities associated with the states. Hence, the generic state is denoted by the triples $\{P, P_c, \boldsymbol{Q}\}$, where the idle state is denoted by $\{0, 0, \emptyset\}$. The set of all the allowed triple $\{P, P_c, \boldsymbol{Q}\}$ is denoted as $\mathbb{S}$.

The various protocols are also characterized by a transition matrix, whose element $p(s_1, s_2)$, $s_1, s_2 \in \mathbb{S}$, represents the probability of moving to state s_2 given that the process is in state s_1.

In order to perform the reward analysis of the semi-Markov models, we also define the reward matrix $\boldsymbol{R}$, the time cost matrix $\boldsymbol{T}$, and the transmission cost matrix $\boldsymbol{C}$, whose elements $r(s_1, s_2)$, $t(s_1, s_2)$ and $c(s_1, s_2)$, are the number of correctly decoded frames, the time cost and the transmission cost associated to the transition from state s_1 to s_2, respectively. In the following Sections we derive these matrices, and perform the protocols analysis.

Derivation of the Transition Probabilities

The transition probabilities of the various protocols are a function of the outage probabilities (conditioned on the failure in the previous phases) and of the arrival process. Note that the identical distribution of the channel statistics allows us to drop the node indices, thus

$$\phi^{\{i,j\}}\left(P, \mathbb{I}, \mathbb{T}\right) = \phi\left(P, \mathbb{I}, \mathbb{T}\right), \quad \forall i, j \in 1, 2, \dots, M. \tag{4.5}$$

Under this assumption, $\mathbb{T}$ is completely determined by the current state. Moreover, for the SA scheme $\mathbb{I}$ is empty, and thus we drop it from the notation. Therefore, we refer to the outage event associated with state $s \in \mathbb{S}$ as ϕ_s, and we denote with $\zeta\left(\phi_s\right)$ its probability.

We provide in the following a detailed description of the chain associated with protocols MCS and MCnS, that have the same structure, apart from the activity of the source node during cooperation phases. A picture of the protocol chain is shown in Fig. 4.2, where the transitions with associated probability greater than zero are represented by arrows. The chains for the simpler protocols are obtained by removing some states, and changing the transition probabilities accordingly. We define ζ_s as the conditioned outage probability associated with state $s \in \mathbb{S}$ with $P > 2$. Each state s with $P > 1$ is reachable from only one state, that we denote as $s_\uparrow$. Therefore,

$$\zeta_s = \zeta\left(\phi_s \mid \phi_{s_\uparrow}\right) = \frac{\zeta\left(\phi_s\right)}{\zeta\left(\phi_{s_\uparrow}\right)}, \tag{4.6}$$

where the second equality comes from the observation that the outage region associated with state s is always contained in the outage region associated with state $s_\uparrow$. $\zeta_{\{1,0,\emptyset\}}$ is defined as $\zeta\left(\{1,0,\emptyset\}\right)$. This probability may be represented in closed–form in the simpler cases, while it may require numerical integration in the more complex cases.

Starting from the idle state $\{0,0,\emptyset\}$ the chain moves to $\{1,0,\emptyset\}$ if a frame arrives at one of the M nodes, otherwise the process remains in the idle state, i.e.,

$$\begin{aligned} p(\{0,0,\emptyset\},\{0,0,\emptyset\}) &= a \\ p(\{0,0,\emptyset\},\{1,0,\emptyset\}) &= 1-a, \end{aligned} \tag{4.7}$$

where a denotes the probability of zero frame arrivals for one slot at all the M nodes, i.e., $a = e^{-\lambda M}$. States $\{P, P_c, \boldsymbol{Q}\}$, with $P = 1, \ldots, M_{\text{ph}}-1$, are characterized by the transition probabilities:

$$\begin{aligned} p(\{P,P_c,\boldsymbol{Q}\},\{0,0,\emptyset\}) &= a\left(1-\zeta_{\{P,P_c,\boldsymbol{Q}\}}\right), \\ p(\{P,P_c,\boldsymbol{Q}\},\{1,0,\emptyset\}) &= (1-a)\left(1-\zeta_{\{P,P_c,\boldsymbol{Q}\}}\right), \\ p(\{P,P_c,\boldsymbol{Q}\},\{P+1,P_c+1,\boldsymbol{Q}\cup P\}) &= \zeta_{\{P,P_c,\boldsymbol{Q}\}}\left[1-\left(\zeta_{\{P,P_c,\boldsymbol{Q}\}}\right)^{M-|\boldsymbol{Q}|-2}\right], \\ p(\{P,P_c,\boldsymbol{Q}\},\{P+1,P_c+(1-\delta_{P_c}),\boldsymbol{Q}\}) &= \zeta_{\{P,P_c,\boldsymbol{Q}\}}\left(\zeta_{\{P,P_c,\boldsymbol{Q}\}}\right)^{M-|\boldsymbol{Q}|-2}; \end{aligned} \tag{4.8}$$

where δ_k is one if $k=0$ and zero otherwise, and $\mid \boldsymbol{Q} \mid$ denotes the number of elements of $\boldsymbol{Q}$. In case of a correctly decoded frame, from these states, the process moves to the first phase state if a frame arrives at one of the M nodes during the current slot, and to the idle state otherwise. If a failure occurs, the process moves to $\{P+1, P_c+1, \boldsymbol{Q}\cup\{P\}\}$ if one of the $M- \mid \boldsymbol{Q} \mid -2$ nodes that are in the idle states and do not already cooperated successfully decoded the frame, otherwise the same cooperating node is maintained. Please note that in the latter case the counter of the phase with a cooperating node is increased only if $P_c > 0$. States $\{M_{\text{ph}}, P_c, Q\}$ have the following transition probabilities:

$$\begin{aligned} p(\{M_{\text{ph}},0,\emptyset\},\{0,0,\emptyset\}) &= a \\ p(\{M_{\text{ph}},0,\emptyset\},\{1,0,\emptyset\}) &= 1-a. \end{aligned} \tag{4.9}$$

In fact, after the M_{ph}–th phase the process is forced to return to the idle state or to start a new frame transmission. All other transitions never happen and have a probability equal to zero.

Performance Analysis

In this Section we define the reward, time cost and transmission cost matrices, and we derive some performance metrics associated with the MCS and MCnS protocols.

The non–zero elements of the reward matrix $\boldsymbol{R}$ are represented by the average number

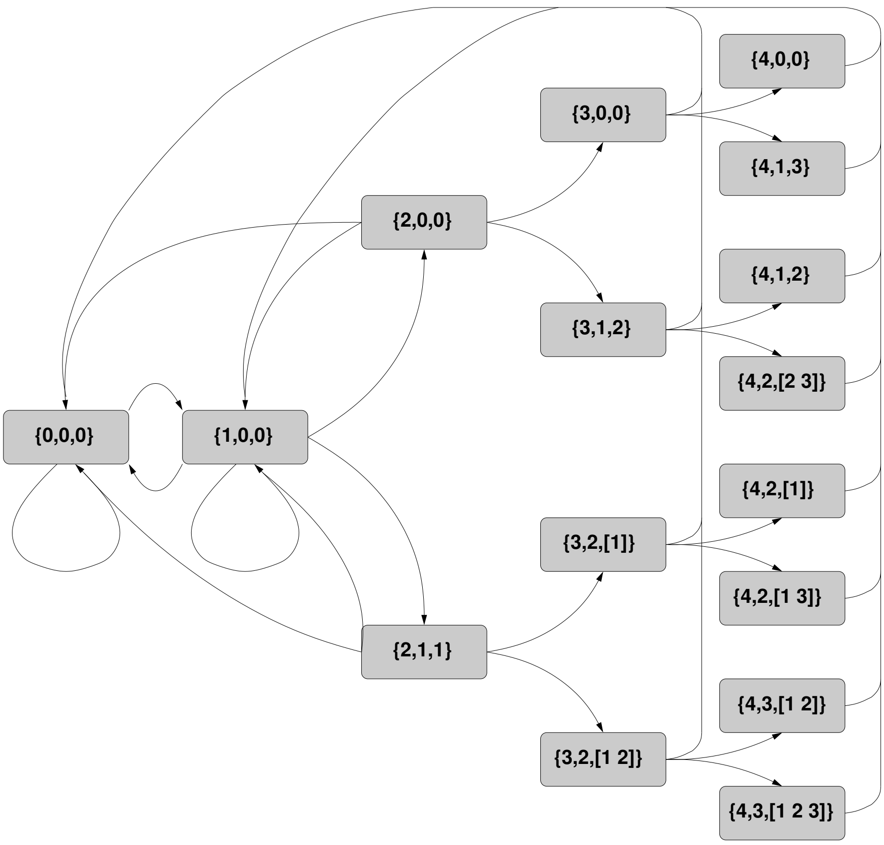


Figure 4.2. *Markov Chain for MCnS and MCS protocols with SA scheme.*

of successfully received frames associated with each transition, defined as follows:

$$
\begin{aligned}
r(\{P,P_c,\boldsymbol{Q}\},\{0,0,\emptyset\}) &= 1, && \text{if } 1 \leq P < M_{\text{ph}} \\
r(\{P,P_c,\boldsymbol{Q}\},\{1,0,\emptyset\}) &= 1, && \text{if } 1 \leq P < M_{\text{ph}} \\
r(\{M_{\text{ph}},P_c,\boldsymbol{Q}\},\{0,0,\emptyset\}) &= 1-\zeta_{\{P,P_c,\boldsymbol{Q}\}}, \\
r(\{M_{\text{ph}},P_c,\boldsymbol{Q}\},\{1,0,\emptyset\}) &= 1-\zeta_{\{P,P_c,\boldsymbol{Q}\}},
\end{aligned} \tag{4.10}
$$

with $\{P,P_c,\boldsymbol{Q}\} \in \mathbb{S}$. In fact, all states with $1 \leq P < M_{\text{ph}}$ return to $\{1,0,\emptyset\}$ or to $\{0,0,\emptyset\}$ only if a success is achieved, while states with $P = M_{\text{ph}}$ achieve an average reward equal to their success probability.

The elements of the cost matrix $\boldsymbol{C}$ for the nC, CnS and MCnS protocols associated with all the transitions from state $s = \{P,P_c,\boldsymbol{Q}\}$ are equal to P, whereas for the CS and MCS protocols they are:

$$
\begin{aligned}
c(\{P,P_c,\boldsymbol{Q}\},\{0,0,\emptyset\}) &= P + P_c \\
c(\{P,P_c,\boldsymbol{Q}\},\{1,0,\emptyset\}) &= P + P_c
\end{aligned} \tag{4.11}
$$

due to the double transmission cost in the phases with a cooperating node. The elements of the time cost matrix are all set to one, because each transmission has a duration equal to one slot.

The average reward, transmission cost, and time cost, referred to as r_{av}, c_{av} and t_{av} respectively, are computed as follows [12]:

$$\begin{aligned}\mathrm{r}_{av} &= \sum_{s\in\mathbb{S}}\pi_s\left(\sum_{d\in\mathbb{S}}p(s,d)r(s,d)\right)\\ \mathrm{c}_{av} &= \sum_{s\in\mathbb{S}}\pi_s\left(\sum_{d\in\mathbb{S}}p(s,d)c(s,d)\right),\\ \mathrm{t}_{av} &= \sum_{s\in\mathbb{S}}\pi_s\left(\sum_{d\in\mathbb{S}}p(s,d)t(s,d)\right),\end{aligned} \tag{4.12}$$

where π_s is the steady–state probability of being in state s. The average throughput, defined as the average number of correctly delivered frames per slot, is then equal to

$$\boldsymbol{Thr} = \mathrm{r}_{av}/\mathrm{t}_{av}. \tag{4.13}$$

The average efficiency is defined as the average number of successful frames per HARQ packet transmitted,

$$\boldsymbol{Eff} = \mathrm{r}_{av}/\mathrm{c}_{av}. \tag{4.14}$$

We now compute the average number of phases per frame transmission, denoted with $\boldsymbol{N_{Ph}}$. We construct a new chain starting from the protocol chains described before. In particular, we eliminate the idle state $\{0,0,\emptyset\}$, and we create a new absorbing state θ. We then modify the transition probabilities as follows:

$$\begin{aligned}p(\{P,P_c,\boldsymbol{Q}\},\{1,0,\emptyset\}) &= 0\\ p(\{P,P_c,\boldsymbol{Q}\},\{0,0,\emptyset\}) &= 0\\ p(\{P,P_c,\boldsymbol{Q}\},\theta) &= 1-\zeta_{\{P,P_c,\boldsymbol{Q}\}} \text{ if } P<M_{\mathrm{ph}}\\ p(\{P,P_c,\boldsymbol{Q}\},\theta) &= 1 \qquad\qquad \text{if } P=M_{\mathrm{ph}}\\ p(\theta,\{P,P_c,\boldsymbol{Q}\}) &= 0\\ p(\theta,\theta) &= 1,\end{aligned} \tag{4.15}$$

while the other previously defined transition probabilities remain unchanged. Let X_n be the state after n transitions. We then define the chain absorption time

$$U=\min\{n\geq 0 : X_n=\theta\}, \tag{4.16}$$

and the average time to absorption, starting from state $s\in\mathbb{S}\setminus\{0,0,\emptyset\}$,

$$\nu_s=E\left[U \mid X_0=s\right]. \tag{4.17}$$

It is simple to obtain the following expressions for ν_s, [12]:

$$\nu_s=1+\sum_d p(s,d)\,\nu_d,\ s\in\mathbb{S}\setminus\{0,0,\emptyset\}. \tag{4.18}$$

Solving (4.18) for all transient states of the chain we find ν_s. The average number of phases per frame transmission is then $\boldsymbol{N_{Ph}}=\nu_{\{1,0,\emptyset\}}$.

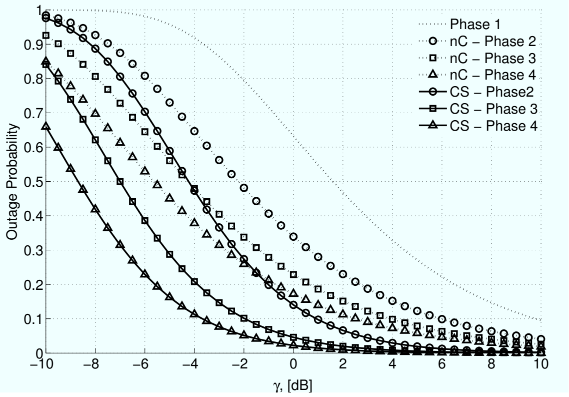


Figure 4.3. *Outage probabilities for the nC and CS protocols as a function of the average SNR γ. $R\!=\!0.1$, $\alpha\!=\!0.1$.*

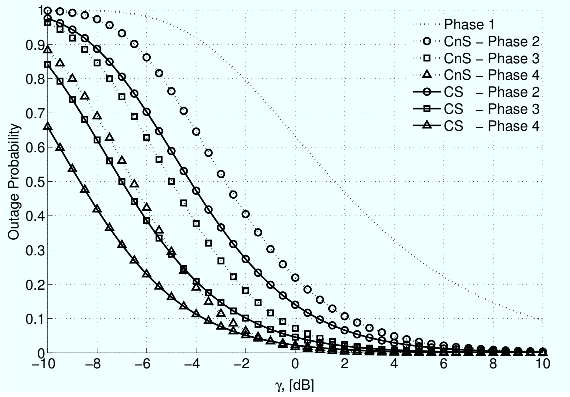


Figure 4.4. *Outage probabilities for the CS and CnS protocols as a function of the average SNR γ. $R\!=\!0.1$, $\alpha\!=\!0.1$.*

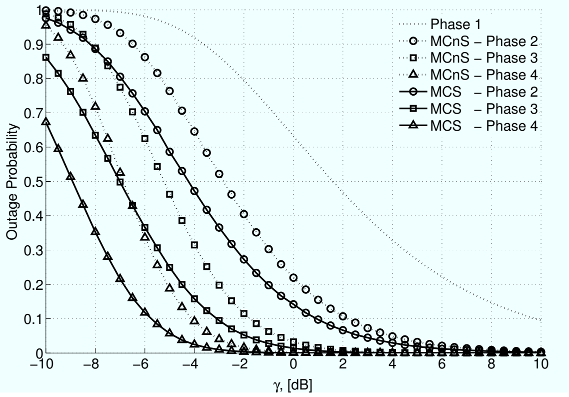


Figure 4.5. *Outage probabilities for the MCS and MCnS protocols as a function of the average SNR γ. $R\!=\!0.1$, $\alpha\!=\!0.1$.*

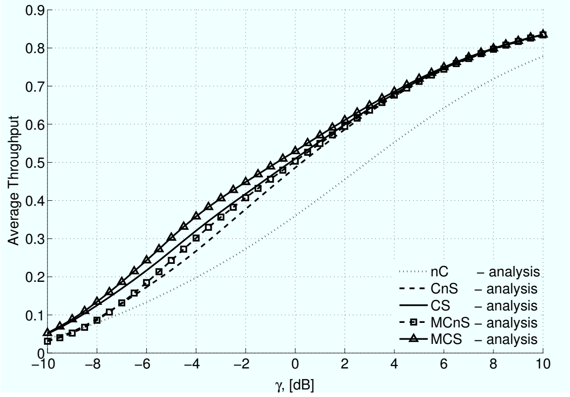


Figure 4.6. *Throughput for the nC, CS, CnS, MCS and MCnS protocols with SA scheme as a function of the average SNR γ. $R\!=\!0.1$, $\alpha\!=\!0.1$.*

4.2.4 Results

Analytical Results

In this Section we present some results derived from the analytical model of Section 4.2.3. A summary of the parameters used in both analysis and simulation is provided in Table 4.1.

Figs. 4.3, 4.4 and 4.5 show the outage probabilities at the various phases for the different protocols, evaluated under the assumption that a cooperating node is always available and for static channel as in Eq. (4.4). Is it possible to observe the different effect of the static channel assumption on the various protocols. Despite the coding gain offered by the transmission of a new part of the codeword, in the nC protocol the outage probability does not decrease very rapidly as the phase index is increased. In fact, for the nC protocol, where the transmitter of the various HARQ packets is the source, the probability of correct decoding at the receiver, conditioned on the outage at the previous phases, decreases or does not increase significantly. Conversely, when a further HARQ packet is sent by a new transmitter, the outage probability is highly reduced. Under the assumption that a cooperating node is

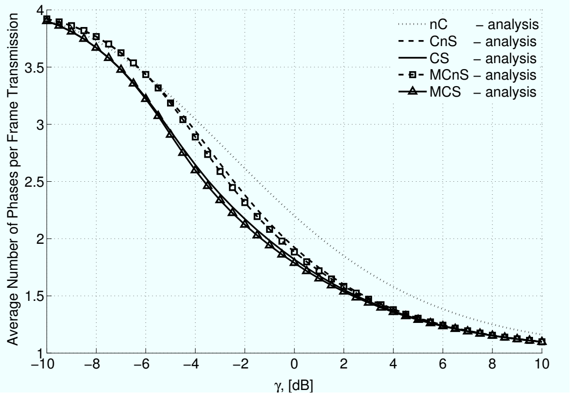


Figure 4.7. *Average number of phases per frame transmission for the nC, CS, CnS, MCS and MCnS protocols with SA scheme as a function of the average SNR γ. $R=0.1$, $\alpha=0.1$.*

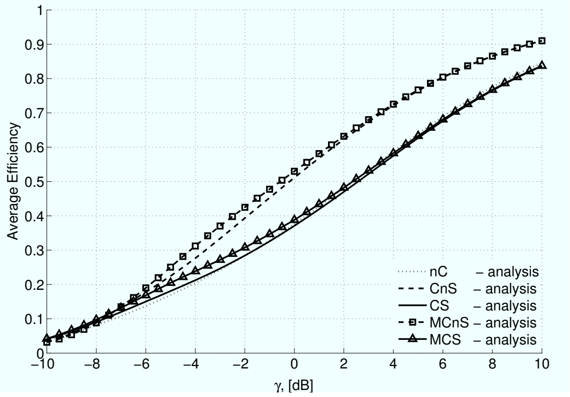


Figure 4.8. *Efficiency for the nC, CS, CnS, MCS and MCnS protocols with SA scheme as a function of the average SNR γ. $R=0.1$, $\alpha=0.1$.*

available, the CnS protocol provides a new transmitter at the second phase, while in case of further failures the cooperating node remains the same at the third and fourth phases. Thus, the outage probability substantially diminishes at the second phase, while the static channel conditions prevent any considerable improvements at subsequent phases. The CS protocol behaves similarly, but provides better performance than CnS, thanks to the higher coding gain achieved in the cooperative phases, that, for the considered transmission rate and encoding rate, outweighs the effects of the mutual interference between the cooperating node and the source transmission. The MCnS and MCS protocols, that change cooperating node at each phase, show significantly improved performance in terms of correct decoding probability even for the third and fourth phases. Figs. 4.6, 4.7 and 4.8 show respectively the average throughput, the average number of phases and the average efficiency for the SA scheme, evaluated through the proposed analytical model. The cooperative protocols generally outperform the nC protocol in terms of throughput, and CS and MCS achieve a higher throughput than their non simultaneous versions CnS and MCnS. At high average SNR, all the cooperative protocols achieve a similar throughput. In fact, in the SA scheme at high λ the throughput is mostly governed by the outage probabilities and all the cooperative protocols abate the failure probability. Note that CS and CnS perform closely to MCS and CnS. In fact, the former two protocols differ from the other two only at the third and fourth phases, that are generally not entered very often. Therefore, in the following only the multiple cooperating node protocols are considered for the sake of clarity. Similar observations apply to the average number of phases per frame transmission. The larger differencies between the cooperative protocols for the SA scheme are visible in the efficiency plot. In fact, the simultaneous schemes suffer a double transmission cost, without a significant gain in terms of fewer phases per frame transmission. Thus the efficiency of CS and MCS is lower than the efficiency achieved by CnS and MCnS, and is comparable with that of nC.

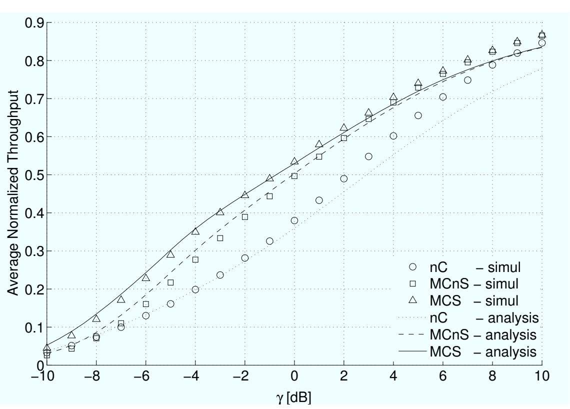


Figure 4.9. *Comparison between analytical and simulation throughput results for the nC, MCS and MCnS protocols with SA scheme as a function of* γ. $R=0.1$, $\alpha=0.1$, $\rho=0.9$.

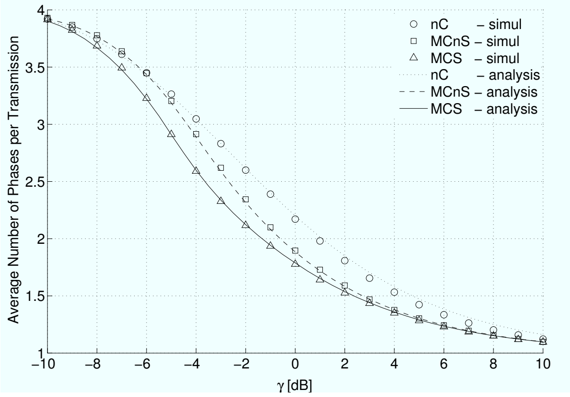


Figure 4.10. *Comparison between analytical and simulation average number of phases per frame transmission for the nC, MCS and MCnS protocols with SA scheme as a function of* γ. $R=0.1$, $\alpha=0.1$, $\rho=0.9$.

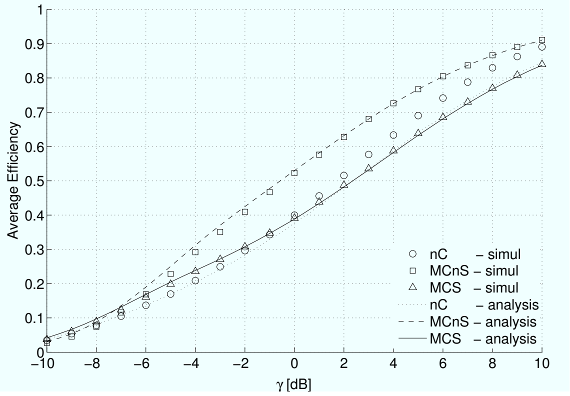


Figure 4.11. *Comparison between analytical and simulation efficiency for the nC, MCS and MCnS protocols with SA scheme as a function of* γ. $R=0.1$, $\alpha=0.1$, $\rho=0.9$.

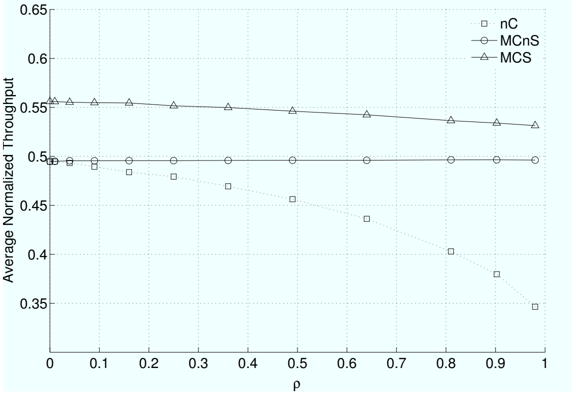


Figure 4.12. *Throughput for the nC, MCS and MCnS protocols with SA scheme as a function of* ρ, $R=0.1$, $\alpha=0.1$, $\gamma=0$.

Simulation Results

In addition to the analysis presented in the previous Section, we also performed detailed simulations of the protocols and MAC described in Sections 4.2.1 and 4.2.2 in order to characterize all the performance metrics in a more complete framework and to assess the performance of the CA scheme. In the simulations we relax the simplifying assumptions used for the analytical model, while still considering the idealized MAC schemes described in Section 4.2.1.

Differently from the analysis, where the channel coefficients are constant during an entire frame transmission and different frame transmissions have independent channel conditions, in simulations the channel is still assumed to be constant for the slot duration but fading coefficients in adjacent slots are correlated with correlation coefficient ρ. More specifically, an autoregressive moving average (ARMA) model is used, and the complex fading coefficient

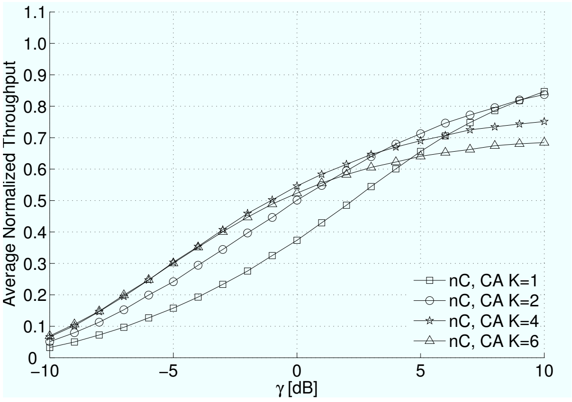


Figure 4.13. *Throughput vs.* γ *for the nC protocol with CA scheme and various values of* K. $R\!=\!0.1$, $\alpha\!=\!0.1$, $\rho\!=\!0.9$.

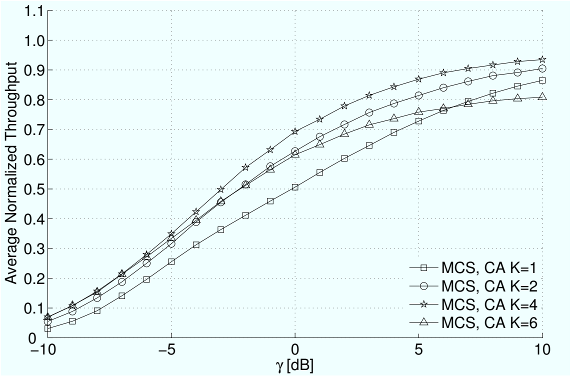


Figure 4.14. *Throughput vs.* γ *for the MCS protocol with CA scheme and various values of* K. $R\!=\!0.1$, $\alpha\!=\!0.1$, $\rho\!=\!0.9$.

between node i and node j for slot t is

$$\phi_{i,j}(t)\!=\!\rho\phi_{i,j}(t-1)\!+\!(1\!-\!\rho)\,\xi(t), \tag{4.19}$$

where $\xi(t)$ is the square norm of an i.i.d. complex Gaussian process with zero mean and unit variance. Moreover, simulations accurately implement FIFO queue management, retransmission mechanism, and backoff policy, in order to obtain a detailed picture of how they affect the performance of the HARQ protocols.

In Figs. 4.9, 4.10 and 4.11 the previously discussed metrics are compared with those evaluated through simulation for the nC, MCS, and MCnS protocols. The analysis provides an accurate prediction for both cooperative protocols, while it is slightly less precise for the nC protocol. This is due to how the analytical assumptions affect the different protocols under different parameters. The analytical derivation is carried out under the assumption that the channel remains constant during a single frame transmission, while it is completely independent across transmissions. The simulation provides a more realistic framework, where the channel is slowly changed slot by slot ($\rho\!=\!0.9$). Thus, for the nC protocol, that suffers in static channel conditions, the simulation results are better than those obtained by analysis, due to the improved channel diversity during a frame transmission. The performance of the cooperative protocols, being less sensitive to channel correlation, is better predicted by the analysis. Note that this effect is more pronounced at high SNRs, where further transmissions of the same frame are less likely. When retransmissions become more frequent, the analysis predicts better performance than the simulation, due to the assumed channel independence between successive transmissions.

In Fig. 4.12 the throughput achieved by the nC, MCs and MCnS protocols with the SA scheme as a function of the channel correlation ρ is shown for $\gamma = 0$ dB. The nC protocol throughput is significantly reduced as the channel correlation increases. As expected, the MCnS protocol is insensitive to the channel correlation, as the cooperating node changes in each phase. Note that the MCS protocol suffers a slight throughput loss as ρ is increased,

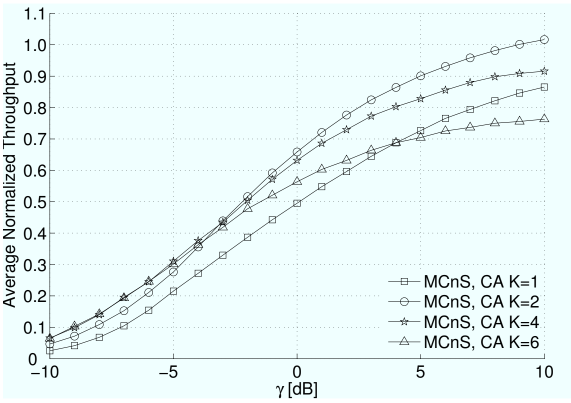


Figure 4.15. *Throughput vs. γ for the MCnS protocol with CA scheme and various values of K. $R\!=\!0.1$, $\alpha\!=\!0.1$, $\rho\!=\!0.9$.*

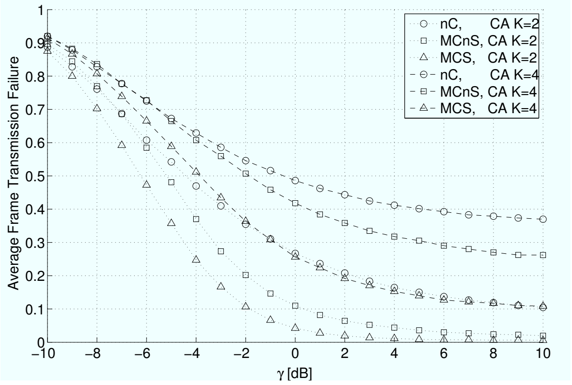


Figure 4.16. *Frame transmission failure rate vs. the average SNR γ for nC, MCS and MCnS protocols with CA scheme, $K=2,\ 4$, $R=0.1$, $\alpha=0.1$, $\rho\!=\!0.9$.*

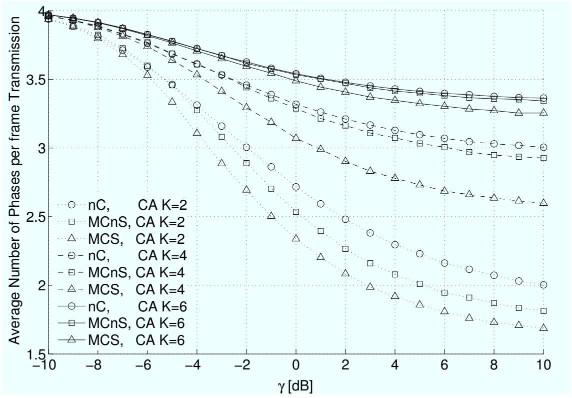


Figure 4.17. *Average number of phases per frame transmission vs. the average SNR γ for nC, MCS and MCnS protocols with CA scheme, $K\!=\!2,\ 4,\ 6$, $R\!=\!0.1$, $\alpha\!=\!0.1$, $\rho\!=\!0.9$.*

due to the reduced effectiveness of the source transmission after a failure at the first phase under highly correlated channel conditions. It is important to observe that nC and MCnS are exactly the same protocol in the current framework, apart from instantaneous queue conditions, due to the symmetry assumption and the uniform node parameters. In fact, if the channel coefficients in two consecutive slots for the same pair of nodes are independent, then letting the same node transmit in the further phases or changing it in each phase is equivalent in terms of diversity.

In the following, we discuss the performance of the various protocols with the CA scheme. Figs. 4.13, 4.14 and 4.15 show the throughput of the nC, MCnS and MCS protocols for various values of the maximum number of nodes that are simultaneously allowed to access the channel, K. This access scheme enables a comparison between cooperation and competition in a system with constrained resources. Note that in the MCS protocol, coop-

General parameters	
Transmission rate R	0.1 bit/s/Hz
Per node arrival rate λ	0.1 frame/s
Number of nodes M	24
M_{tx}	8
M_{ph}	4
α	0.1
Simulation only parameters	
Maximum queue length	32 frames
Backoff Parameter b_w	2
Max_{fails}	4
Simulation slots	200000

Table 4.1. *System Parameters.*

erating nodes, transmitting simultaneously with the original source, decrease the number of frame transmissions in the same slot. Moreover, in low traffic environments, i.e., when the access resource is not completely used, MCS may increase the number of interfering nodes per slot. In the MCnS protocol, cooperating nodes may be forced to defer their frame transmissions for cooperation purposes.

In Fig. 4.13, it is possible to observe that at high SNR CA schemes with greater K achieve a better throughput, while on the other hand at low SNR better performance is obtained by the schemes with strong access control. In fact, at high SNR the system is limited by interference, and the failure rate rapidly degrades with the number of simultaneously transmitting nodes, as shown in Fig. 4.16, and the larger number of simultaneous transmissions is not sufficient to balance the increased failure rate. On the other hand, at low SNR the system is dominated by noise, thus the interference between users has a diminished relevance, while a higher number of simultaneous frame transmissions increases the probability that a source–destination pair has a sufficiently high channel coefficient. With the MCS protocol the increased coding gain provided by the simultaneous transmission of multiple HARQ packets for the same frame transmission makes the interference vs. failure rate tradeoff slightly different than for the nC protocol. In the MCS case, the system can support higher values of K, even at high SNR, without incurring any throughput degradation. It is possible to observe that generally MCS achieves higher throughput than nC. In fact, although the number of simultaneously ongoing frame transmissions is reduced by the transmissions of cooperating nodes, their duration is on average shorter and their success rate is higher, see Figs. 4.16 and 4.17, so that the overall throughput is improved. In the MCnS case a single HARQ packet can be transmitted per frame transmission per slot, and thus the system is affected by interference, similarly to the nC protocol. Nevertheless, the higher diversity provided by MCnS ensures a lower frame transmission failure rate and generally a lower number of phases per frame transmission. Thus, the transmission rate in the system is reduced, due to the lower number of retransmissions and to their shorter average duration, and thus the interference is lower on average. In conclusion, cooperation, providing diversity, seems capable of increasing the capacity of the system, and, furthermore, to effectively deal with the HARQ mechanism, even in systems where the cooperative transmissions compete with frame transmissions.

4.3 Outage Analysis of the Network

In the previous Section, we discussed a simple scenario. In this Section, we introduce issues related to interference in an asynchronous network where nodes make use of a cooperative hybrid ARQ.

We consider an ad hoc network with half duplex nodes transmitting simultaneously on the same channel and using a HARQ cooperative protocol. Differently from most of prior work analyzing cooperative schemes where orthogonality is assumed, we analyze the performance of the interfering network, taking into account that cooperation may increase the interference in the network as both the source and the relay may transmit simultaneously. In particular, we assume that when a node is able to decode the first transmission and the intended destination failed to decode, the cooperator starts transmitting redundancy bits simultaneously with the source, thus implementing a distributed multiple input single output (MISO) scheme. However, in order to have a simple protocol without the symbol synchronization requirement of space-time coded transmission [17], transmissions from source and relay overlap and the receiver detects the two transmissions with mutual interference [18]. For comparison purposes we also consider two versions of the DF protocol with random and opportunistic node selection based on the distance among nodes, respectively.

Our main contribution of this Section is the analysis of the network performance of the considered cooperative HARQ in the interference-limited network. In particular, we first compute the outage probability of various cooperative HARQ protocols under interference conditions. While in the case of orthogonal transmissions, network throughput is easily derived by the behavior of a simple network with three nodes, in our scenario network performance is determined by the interference caused by other nodes. In fact, the number of HARQ retransmissions depends on the interference level, which in turns depends on the node activity, i.e., the HARQ retransmissions of surrounding nodes. Therefore, we derive the steady state condition of the network, where the average interference generated by each node is compatible with the average duration of HARQ retransmissions.

The rest of this part of the Chapter is organized as follows. In Section 4.3.1 we describe the model of the network, including physical and medium access control (MAC) layers and the cooperative protocol. The outage probability for a single node transmission in the considered scenario is derived in Section 4.3.2. The steady state analysis of the network is carried out in Section 4.3.3. Numerical results on the network performance for the considered cooperative protocols are presented in Section 4.3.4.

4.3.1 Network model

In this paper we consider an ad hoc network where nodes can communicate directly with each other. Key characteristics of the network are: a) HARQ for error control, b) cooperation among nodes to increase efficiency, and c) simultaneous transmissions by many nodes in the network, to provide multiplexing.

MAC protocol

For the description of the MAC protocol, consider a source node S that attempts to transmit a data packet to a destination node D. A third cooperating node C may support this communication, as detailed in the following.

As in a conventional HARQ protocol, time is divided into frames, each comprising a data and an acknowledgment slot. In the data slot, coded data are transmitted by node S and/or node C, while in the acknowledgment slot, the correct or wrong reception is reported by node D. The transmission of a single data packet lasts at most $(N_{\max} + 1)$ frames, according to HARQ, where $N_{\max}$ is the maximum number of retransmissions. All frames have the same duration T and the transmission bandwidth is B.

Coded cooperation is implemented at node S by encoding the data packet and transmitting different portions of the coded packet during the various frames. For good channel conditions, node D is able to decode the data packet using only the first frame, which already contains all the information. However, for bad channel conditions, node D combines the information received in successive frames and attempts to decode the packet.

By using a cyclic redundancy check (CRC), node C determines whether it has correctly decoded the first frame and, if so, it cooperates in the subsequent frames by acting as a second source, re-encoding the packet and transmitting various sections of the coded packet.

In particular, we consider two configurations for cooperation:

- **Decode and forward** (DF): the first transmission is performed by node S. If needed, retransmissions in forthcoming frames are performed by node C if it decoded the packet correctly, and by node S otherwise.

- **Multiple input-single output cooperation** (MISO): the first transmission is performed by node S. If needed, retransmissions in forthcoming frames are performed by both node S and node C if C decoded the packet correctly, and only by node S otherwise.

The MISO configuration benefits from both the retransmissions of node S and the diversity of node C. Still, as we assume that nodes do not know the channel on which they are transmitting, signals coming from S and C are not received coherently and in general interfere. Indeed, they are detected as two separate signals and contain different redundancy bits relative to the same data packet.

The choice of the cooperator is relevant for the system performance. Indeed, if node C has a good link toward D it can provide a significant throughput increase, while in a DF configuration, if link C-D is poor, cooperation may provide even worse performance than no cooperation. Hence, beyond DF, where cooperation is implemented regardless of cooperator conditions, we consider also opportunistic DF (O-DF) [19], where cooperation is activated only if the signal to noise plus interference ratio (SNIR) of the C-D link is better than that of the S-D link. In this paper we do not explicitly address the issue of how the nodes estimate instantaneous SNIR's and distances with respect to other nodes, but rather focus on the comparison of cooperative techniques.

Channel and interference power description

We consider two phenomena of wireless propagation: path-loss and fading. In a transmission between nodes at distance d, the received power can be written as

$$P(d) = P_r \left(\frac{d}{d_0}\right)^{-\kappa} x \tag{4.20}$$

where P_r is the average received power at reference distance d_0, κ is the path-loss exponent (e.g., 4), and x is an exponential random variable that represents fading. We assume that all nodes transmit at the same power level so that P_r is the same for all nodes and for the sake of a simpler notation, we set $d_0 = 1$ in the following. The probability density function (pdf) of the received power assuming that a node transmits from distance d is therefore

$$f_{P|d}(a|d) = \frac{1}{P_r d^{-\kappa}} e^{-\frac{a}{P_r d^{-\kappa}}}\,, \quad a \geq 0\,. \tag{4.21}$$

The average received power is $P_r d^{-\kappa}$.

The power of interference coming from other nodes in the network depends on the number of active nodes, as well as on their spatial distribution. Assuming that interfering signals are independent and identically distributed, we can approximate the total interference power through the one-sided form of the central limit theorem as a χ^2 distribution with order equal to the number of interferers $n_{\mathtt{I}}$, [20]. The pdf of the interference power can be approximated as [21, pg. 235]

$$f_{P_{\text{tot}}}(a) \approx \frac{a^{n_{\mathtt{I}}\mu^2/\sigma^2 - 1} e^{-a\mu/\sigma^2}}{\left(\frac{\sigma^2}{\mu}\right)^{n_{\mathtt{I}}\mu^2/\sigma^2} \Gamma\left(\frac{n_{\mathtt{I}}\mu^2}{\sigma^2}\right)}\,, \quad a \geq 0 \tag{4.22}$$

and the characteristic function of the interference can be approximated as

$$\psi_{P_{\text{tot}}}(\omega) = \mathrm{E}\left[e^{j\omega P_{\text{tot}}}\right] \approx \left(1 - j\frac{\sigma^2}{\mu}\omega\right)^{-\frac{n_{\mathtt{I}}\mu^2}{\sigma^2}}\,, \tag{4.23}$$

where μ and σ are the mean and standard deviation of the interference power generated by a single node.

We assume that the channel and the interference power do not change for the entire duration of a packet transmission, which may last up to $(N_{\max} + 1)$ frames. We also assume that each packet transmission has independent fading and interference characteristics (block fading channel).

4.3.2 Outage probability

In this section we derive the expression of the outage probability for the cooperative scenario described above. We assume that the error protection codes are ideal, i.e., when the number of bits per frame goes to infinity, we can reach the normalized Shannon capacity $\log_2(1+\gamma)$, with γ the signal to noise plus interference ratio (SNIR) between the transmitter and the receiver. Note that the suboptimal performance of realistic codes and finite frame lengths can be accounted for by a constant SNIR gap Γ_{gap}, and in this case the achievable rate is $\log_2(1+\gamma/\Gamma_{\text{gap}})$. In the following, for the sake of a simpler notation, we assume ideal coding, i.e., $\Gamma_{\text{gap}} = 1$. While some literature has considered multiuser detection at node $\mathtt{D}$ [22], as a realistic assumption for a low-complexity node we assume that node $\mathtt{D}$ performs single-user decoding of the signals coming from nodes $\mathtt{S}$ and $\mathtt{C}$. Hence, no interference cancellation or any other multiuser detection is performed at node $\mathtt{D}$.

We focus on the transmission of a single packet of N_S symbols. We indicate with $\gamma_{\mathtt{S},\mathtt{D}}(n)$, $\gamma_{\mathtt{C},\mathtt{D}}(n)$ and $\gamma_{\mathtt{S},\mathtt{C}}(n)$ the SNIR at frame n between nodes $\mathtt{S}$ and $\mathtt{D}$, nodes $\mathtt{C}$ and $\mathtt{D}$, and nodes $\mathtt{S}$ and $\mathtt{C}$, respectively, with $n = 0, 1, \ldots, N_{\max}$. Note that, although we assume a constant interference power for the entire packet duration, the SNIRs $\gamma_{\mathtt{S},\mathtt{D}}(n)$ and $\gamma_{\mathtt{C},\mathtt{D}}(n)$ are in general dependent on the frame index n, as interference may arise in the case of simultaneous transmissions from node $\mathtt{S}$ and node $\mathtt{C}$.

Let the maximum transmission time for a single packet be $T_{\text{tot}} = (N_{\max}+1)T$.

Since nodes $\mathtt{S}$ and $\mathtt{C}$ are not always active, depending on the MAC protocol and on whether node $\mathtt{C}$ has decoded the first frame, we indicate with $\delta_{\mathtt{S}}(n)$ and $\delta_{\mathtt{C}}(n)$ the activity of $\mathtt{S}$ and $\mathtt{C}$ during frame n. In particular,

$$\delta_{\mathtt{S}}(n) = \begin{cases} 1 & \text{if } \mathtt{S} \text{ is active in frame } n \\ 0 & \text{otherwise}\,; \end{cases} \tag{4.24}$$

$$\delta_{\mathtt{C}}(n) = \begin{cases} 1 & \text{if } \mathtt{C} \text{ is active in frame } n \\ 0 & \text{otherwise}\,. \end{cases} \tag{4.25}$$

Since we assume that transmit nodes do not know the channel and are not symbol-synchronous, $\mathtt{S}$ and $\mathtt{C}$ transmit different redundancy bits relative to the same data packet and the two signals are detected separately and mutually interfere. Let R [bit/s/Hz] be the rate of the first frame, normalized with respect to the bandwidth, and $R/(N+1)$ the rate after N retransmissions. Then the outage event after $(N+1)$ frames is defined as

$$\begin{aligned} C = &\sum_{n=0}^{N} \delta_{\mathtt{S}}(n) \log_2[1+\gamma_{\mathtt{S},\mathtt{D}}(n)] \\ &+ \sum_{n=0}^{N} \delta_{\mathtt{C}}(n) \log_2[1+\gamma_{\mathtt{C},\mathtt{D}}(n)] < R\,. \end{aligned} \tag{4.26}$$

Note that (4.26) holds true under the assumption that $\mathtt{D}$ performs separate decoding for signals coming from $\mathtt{S}$ and $\mathtt{C}$ with no interference cancellation.

In both (O-)DF and MISO cases, node C is active if it is able to decode the first frame from S and it satisfies the criterion for cooperator selection. Let $\mathcal{E}$ be the event of node C being active, and let $\bar{\mathcal{E}}$ be the complementary event.

In the following we compute the outage probability for given a) distances among nodes S, C and D, b) number of frames $N+1$, c) number of interferers $n_{\mathtt{I}}$, averaged over noise, channel fading and interferers positions.

The probability of being in outage can be conditioned on $\mathcal{E}$ and written as

$$\mathrm{P}[\text{out}] = \mathrm{P}\left[C < R|\mathcal{E}\right]\mathrm{P}[\mathcal{E}] + \mathrm{P}\left[C < R|\bar{\mathcal{E}}\right](1 - \mathrm{P}[\mathcal{E}]). \tag{4.27}$$

From (4.26) and (4.27) by defining

$$I_{\mathcal{E}} = \mathrm{P}[\mathcal{E}] \tag{4.28}$$

$$I_{\mathcal{C}} = \mathrm{P}\left[\prod_{n=0}^{N}(1+\gamma_{\mathtt{S},\mathtt{D}}(n))^{\delta_{\mathtt{S}}(n)}\prod_{n=0}^{N}(1+\gamma_{\mathtt{C},\mathtt{D}}(n))^{\delta_{\mathtt{C}}(n)} < 2^R \middle| \mathcal{E}\right] \tag{4.29}$$

$$I_{\mathcal{N}} = \mathrm{P}\left[\prod_{n=0}^{N}(1+\gamma_{\mathtt{S},\mathtt{D}}(n))^{\delta_{\mathtt{S}}(n)} < 2^R \middle| \bar{\mathcal{E}}\right] \tag{4.30}$$

we obtain

$$\mathrm{P}[\text{out}] = I_{\mathcal{C}} I_{\mathcal{E}} + I_{\mathcal{N}}(1 - I_{\mathcal{E}})\,. \tag{4.31}$$

Cooperation probability for non opportunistic schemes. For DF and MISO schemes, no constraint is set on the quality of the C-D link and the event that node C is active is

$$\mathcal{E} = \{\log_2[1+\gamma_{\mathtt{S},\mathtt{C}}(0)] \geq R\}\ . \tag{4.32}$$

Cooperation probability for O-DF. For O-DF instead, node C is active when both it is able to decode the first transmission and the SNIR between C and D is higher than the SNIR between S and D. In this case, $\mathcal{E}$ can be written as

$$\mathcal{E}_{\mathrm{O-DF}} = \{\log_2[1+\gamma_{\mathtt{S},\mathtt{C}}(0)] \geq R\,\text{and}\,\gamma_{\mathtt{S},\mathtt{D}}(0) \leq \gamma_{\mathtt{C},\mathtt{D}}(0)\}\ . \tag{4.33}$$

Note that C is a random variable, depending on the random values of $\gamma_{\mathtt{S},\mathtt{D}}(n)$ and $\gamma_{\mathtt{C},\mathtt{D}}(n)$.

Decode and forward

For the (O-)DF protocol, only one node is transmitting to node D at any time, which may be either node S or node C. In particular, for node S we have

$$\delta_{\mathtt{S}}(n) = \begin{cases} 1 & n = 0 \\ 1 & n > 0 \text{ and C not active} \\ 0 & \text{otherwise}\,. \end{cases} \tag{4.34}$$

For node C we have

$$\delta_{\mathtt{C}}(n) = \begin{cases} 1 & n > 0 \text{ and C active} \\ 0 & \text{otherwise}\,. \end{cases} \tag{4.35}$$

Let P_i be the interference power seen by node $\mathtt{D}$. Then SNIR's can be written as

$$\gamma_{\mathtt{S},\mathtt{D}}(n) = \begin{cases} \frac{P(d_{\mathtt{S},\mathtt{D}})}{N_0+P_i} & n=0 \\ 0 & n>0 \text{ and } \mathtt{C} \text{ active} \\ \frac{P(d_{\mathtt{S},\mathtt{D}})}{N_0+P_i} & \text{otherwise} \end{cases} \tag{4.36}$$

$$\gamma_{\mathtt{C},\mathtt{D}}(n) = \begin{cases} \frac{P(d_{\mathtt{C},\mathtt{D}})}{N_0+P_i} & n>0 \text{ and } \mathtt{C} \text{ active} \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

where N_0 is the noise power. Let $\bar{\gamma}_{\mathtt{C},\mathtt{D}} = \gamma_{\mathtt{C},\mathtt{D}}(1)$ when $\mathtt{C}$ is active and $\gamma_{\mathtt{S},\mathtt{D}} = \gamma_{\mathtt{S},\mathtt{D}}(0)$.

For DF, cooperation takes place if node $\mathtt{C}$ decodes the first transmission and, as shown in the Appendix, $\mathcal{E}$ becomes

$$I_{\mathcal{E},\mathrm{DF}} = e^{-\frac{(2^R-1)N_0}{P_r d_{\mathtt{S},\mathtt{C}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left(j\frac{(2^R-1)}{P_r d_{\mathtt{S},\mathtt{C}}^{-\kappa}}\right) \tag{4.38}$$

while $I_{\mathcal{C}}$ and $I_{\mathcal{N}}$ become

$$\begin{aligned} I_{\mathcal{C},\mathrm{DF}} =& 1 - e^{-\frac{A' N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left[j\frac{A'}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right] \\ & - \frac{N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{b=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(b) N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{b N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left[j\left(\frac{b'_{\mathrm{DF}}(b)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{b}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right] \mathrm{d}b \\ & - \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{b=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(b) N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{b N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi'\left[j\left(\frac{b'_{\mathrm{DF}}(b)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{b}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right] \mathrm{d}b \end{aligned} \tag{4.39}$$

$$I_{\mathcal{N},\mathrm{DF}} = 1 - e^{-\frac{(2^{R/(N+1)}-1)N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left(j\frac{(2^{R/(N+1)}-1)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right), \tag{4.40}$$

where

$$b'_{\mathrm{DF}}(x) = \left[\left(\frac{2^R}{(1+x)}\right)^{1/N} - 1\right], \tag{4.41}$$

$$A' = 2^R - 1, \tag{4.42}$$

and

$$\psi'(\omega) = \mu n_{\mathtt{I}} \left(1 - j\frac{\sigma^2}{\mu}\omega\right)^{-\left(\frac{n_{\mathtt{I}}\mu^2}{\sigma^2}+1\right)}. \tag{4.43}$$

From these results we obtain the outage probability for given distances among nodes $\mathrm{P}[\mathrm{out}|\mathrm{DF}, d_{\mathtt{S},\mathtt{D}}, d_{\mathtt{C},\mathtt{D}}, d_{\mathtt{S},\mathtt{C}}, n_{\mathtt{I}}$ from (4.31) using (4.38), (4.39) and (4.40).

For O-DF, beyond decoding of the first transmission, it is also required that $\gamma_{\mathtt{S},\mathtt{D}} \leq \gamma_{\mathtt{C},\mathtt{D}}$. As shown in the Appendix, by defining

$$I_{\mathcal{S}} = \frac{d_{\mathtt{C},\mathtt{D}}^{-\kappa}}{d_{\mathtt{S},\mathtt{D}}^{-\kappa} + d_{\mathtt{C},\mathtt{D}}^{-\kappa}} \tag{4.44}$$

$I_{\mathcal{E}}$, $I_{\mathcal{C}}$ and $I_{\mathcal{N}}$ become

$$I_{\mathcal{E},\mathrm{O-DF}} = I_{\mathcal{S}} e^{-\frac{(2^R-1)N_0}{P_r d_{\mathtt{S,C}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}} \left(j \frac{(2^R-1)}{P_r d_{\mathtt{S,C}}^{-\kappa}} \right) , \tag{4.45}$$

$$\begin{aligned} I_{\mathcal{C},\mathrm{O-DF}} =& 1 - e^{-A'_{\mathrm{O}} N_0 \left(\frac{1}{P_r d_{\mathtt{S,D}}^{-\kappa}} + \frac{1}{P_r d_{\mathtt{C,D}}^{-\kappa}} \right)} \psi_{P_{\mathrm{tot}}} \left[j A'_{\mathrm{O}} \left(\frac{1}{P_r d_{\mathtt{S,D}}^{-\kappa}} + \frac{1}{P_r d_{\mathtt{C,D}}^{-\kappa}} \right) \right] \\ & - \frac{N_0}{I_{\mathcal{S}} P_r d_{\mathtt{S,D}}^{-\kappa}} \int_{x=0}^{A'_{\mathrm{O}}} e^{-\frac{b'_{\mathrm{DF}}(x) N_0}{P_r d_{\mathtt{C,D}}^{-\kappa}}} e^{-\frac{x N_0}{P_r d_{\mathtt{S,D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}} \left[j \left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C,D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S,D}}^{-\kappa}} \right) \right] \mathrm{d}x \\ & - \frac{1}{I_{\mathcal{S}} P_r d_{\mathtt{S,D}}^{-\kappa}} \int_{x=0}^{A'_{\mathrm{O}}} e^{-\frac{b'_{\mathrm{DF}}(x) N_0}{P_r d_{\mathtt{C,D}}^{-\kappa}}} e^{-\frac{x N_0}{P_r d_{\mathtt{S,D}}^{-\kappa}}} \psi' \left[j \left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C,D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S,D}}^{-\kappa}} \right) \right] \mathrm{d}x \end{aligned} \tag{4.46}$$

$$I_{\mathcal{N},\mathrm{O-DF}} = I_{\mathcal{N},\mathrm{DF}} , \tag{4.47}$$

where $b'_{\mathrm{DF}}(x)$ is as in (4.41) and

$$A'_{\mathrm{O}} = 2^{R/(N+1)} - 1 . \tag{4.48}$$

In this case, the outage probability is given by (4.31) using (4.45), (4.46) and (4.47).

Multiple input-single output cooperation (MISO)

For the MISO case, when node $\mathtt{C}$ is able to decode the first frame transmitted by node $\mathtt{S}$, it starts transmitting simultaneously to node $\mathtt{S}$ from the second frame. Hence we have $\delta_{\mathtt{S}}(n) = 1$, $\forall n$ and $\delta_{\mathtt{C}}(n)$ as in (4.35). The SNIR's now account for the mutual interference of transmissions of nodes $\mathtt{S}$ and $\mathtt{C}$, when $\mathtt{C}$ is active. Note that in a MISO configuration, the total transmit power is doubled with respect to DF, since both $\mathtt{S}$ and $\mathtt{C}$ transmit at maximum power. This is a distinctive feature of cooperative systems with respect to a single link with nodes equipped with multiple antennas, where the power must be distributed among the antennas. In particular, we obtain

$$\gamma_{\mathtt{S,D}}(n) = \begin{cases} \frac{P(d_{\mathtt{S,D}})}{N_0+P_i} & n = 0 \\ \frac{P(d_{\mathtt{S,D}})}{N_0+P_i+P(d_{\mathtt{C,D}})} & n > 0 \text{ and } \mathtt{C} \text{ active} \\ \frac{P(d_{\mathtt{S,D}})}{N_0+P_i} & n > 0 \text{ and } \mathtt{C} \text{ not active,} \end{cases} \tag{4.49}$$

$$\gamma_{\mathtt{C,D}}(n) = \begin{cases} \frac{P(d_{\mathtt{C,D}})}{N_0+P_i+P(d_{\mathtt{S,D}})} & n > 0 \text{ and } \mathtt{C} \text{ active} \\ 0 & n = 0 . \end{cases} \tag{4.50}$$

If nodes use CDMA, interference of simultaneous transmissions is reduced by the use of spreading codes and when $\mathtt{C}$ is active, equations (4.49) and (4.50) become $\frac{P(d_{\mathtt{S,D}})}{N_0+\rho P_i+\rho P(d_{\mathtt{C,D}})}$ and $\frac{P(d_{\mathtt{C,D}})}{N_0+\rho P_i+\rho P(d_{\mathtt{S,D}})}$, respectively, where ρ is the correlation among spreading codes.

The probability of event $\mathcal{E}$ is the same for both MISO and DF cases, since decoding of the first transmission by node $\mathtt{C}$ is independent of the cooperation scheme. Moreover, if cooperation does not take place, DF and MISO have the same outage probability. Therefore

$$I_{\mathcal{E},\mathrm{MISO}} = I_{\mathcal{E},\mathrm{DF}} \tag{4.51}$$

$$I_{\mathcal{N},\mathrm{MISO}} = I_{\mathcal{N},\mathrm{DF}}\,. \tag{4.52}$$

Event $\mathcal{C}$ instead is affected by MISO, and by letting $\bar{\gamma}_{\mathtt{C},\mathtt{D}} = \gamma_{\mathtt{C},\mathtt{D}}(1)$ when $\mathtt{C}$ is active, $\gamma_{\mathtt{S},\mathtt{D}} = \gamma_{\mathtt{S},\mathtt{D}}(0)$ and $\bar{\gamma}_{\mathtt{S},\mathtt{D}} = \gamma_{\mathtt{S},\mathtt{D}}(1)$, we have

$$I_{\mathcal{C},\mathrm{MISO}} = \mathrm{P}\left[(1+\gamma_{\mathtt{S},\mathtt{D}})(1+\bar{\gamma}_{\mathtt{C},\mathtt{D}})^{N}\,(1+\bar{\gamma}_{\mathtt{S},\mathtt{D}})^{N} < 2^{R}\right]\,. \tag{4.53}$$

Let us define

$$\xi'(x) = 2^{R/N}\,(1+x)^{-1/N} - 1\,, \tag{4.54}$$

$$b'_{\mathrm{MISO}}(x) = \frac{1}{2}(1+x)\left[-(1-\xi'(x)) + \sqrt{(1-\xi'(x))^2 - \frac{4[x-\xi'(x)]}{(1+x)}}\right]^{+}\,, \tag{4.55}$$

where $[y]^{+} = 0$ for $y < 0$ or y complex, and $[y]^{+} = y$ otherwise. As shown in the Appendix, we obtain

$$\begin{aligned} I_{\mathcal{C},\mathrm{MISO}} =& 1 - e^{-\frac{A' N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}}\,\psi_{P_{\mathrm{tot}}}\left[j\frac{A'}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right] \\ & - \frac{N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{MISO}}(x) N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}}\, e^{-\frac{x N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}}\,\psi_{P_{\mathrm{tot}}}\left[j\left(\frac{b'_{\mathrm{MISO}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x \\ & - \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{MISO}}(x) N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}}\, e^{-\frac{x N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}}\,\psi'\left[j\left(\frac{b'_{\mathrm{MISO}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x \end{aligned} \tag{4.56}$$

Lastly, the outage probability can be written as (4.31) using (4.38), (4.40) and (4.56).

4.3.3 Average network behavior

In Section 4.3.2 we have derived the outage probability for a given placement of source, destination and cooperator nodes and for a given number of interfering nodes. However, the number of interfering nodes itself is related to the traffic intensity, the number of nodes and the transmission duration. While the first two parameters define the network scenario, the transmission duration is related to the selected protocol and therefore can not be set a priori but must be derived from the outage probability. In this section we perform a steady state analysis of the network behavior, to derive the average transmission duration, the average per node throughput, and other relevant parameters.

Network scenario

As reference scenario we consider ν nodes uniformly randomly located in a ring around node $\mathtt{D}$, having inner radius R_{min} and outer radius R_{max}. The transmitting nodes, except node $\mathtt{S}$, are interfering and we assume that on average $\nu_{\mathtt{I}} < \nu$ nodes are interfering. By assuming a uniform distribution of interfering nodes on the plane and assuming that transmissions of interfering nodes are independent events, the number of interfering nodes $n_{\mathtt{I}}$ has a binomial distribution with average $\nu_{\mathtt{I}}$. Note that nodes may interfere because they are operating either as source or as cooperators for other transmissions.

The average interference power due to a single interferer can be written as

$$\mu = \int_{R_{\min}}^{R_{\max}} \frac{P_r}{\pi(R_{\max}^2 - R_{\min}^2)} 2\pi \left(\frac{d}{d_0}\right)^{-\kappa} d \,\mathrm{d}d = \frac{2P_r}{(R_{\max}^2 - R_{\min}^2) d_0^{-\kappa}(2-\kappa)} (R_{\max}^{2-\kappa} - R_{\min}^{2-\kappa}) \,, \quad (4.57)$$

while its variance is

$$\sigma^2 = \frac{2P_r^2}{(R_{\max}^2 - R_{\min}^2) R_{\min}^{-2\kappa}(2-2\kappa)} (R_{\max}^{2-2\kappa} - R_{\min}^{2-2\kappa}) - \mu^2 \,. \quad (4.58)$$

We consider as *available cooperators* all the nodes that do not operate as source, destination or cooperators for other transmissions. The average number of available cooperators is ν_{aC} and at most only one of them cooperates with node $\mathtt{S}$. We consider the following options for cooperator selection, both performed before transmission of the data packet:

- **Random Selection** (RS). A node is selected at random before transmission;

- **Outage Probability Selection** (OPS). The cooperator being placed at the minimum distance from the destination is selected, [23]. Note that in making this choice we are not considering the instantaneous SNIR, as instead proposed in [24].

Each node attempts to transmit data with an average traffic intensity of λ packets per seconds.

Average transmission duration

Assuming that on average there are $\nu_{\mathtt{I}}$ interfering nodes, the average duration of a communication is

$$\tau = \sum_{n=0}^{N_{\max}} (n+1)T\mathrm{P}[\mathrm{succ}\, N = n|\nu, \nu_{\mathtt{I}}] + \left(1 - \sum_{n=0}^{N_{\max}} \mathrm{P}[\mathrm{succ}\, N = n|\nu, \nu_{\mathtt{I}}]\right) (N_{\max}+1)T \,, \quad (4.59)$$

where $\mathrm{P}[\mathrm{succ}\, N = n|\nu, \nu_{\mathtt{I}}]$ is the average probability of having a success in exactly n retransmissions, i.e., for $n > 0$

$$\mathrm{P}[\mathrm{succ}\, N = n|\nu, \nu_{\mathtt{I}}] = (1 - \mathrm{P}[\mathrm{out}|\nu, \nu_{\mathtt{I}}, N = n])\mathrm{P}[\mathrm{out}|\nu, \nu_{\mathtt{I}}, N = n-1] \,, \quad (4.60)$$

while for $n = 0$ $\mathrm{P}[\mathrm{succ}\, N = 0] = 1 - P[\mathrm{out}|\nu, \nu_{\mathtt{I}}, N = 0]$.

The success probability as a function of the average number of nodes $\nu_{\mathtt{I}}$ can be obtained by Monte Carlo method, by randomly generating the number of interfering nodes according to the binomial distribution with parameters chosen according to the selected transmission scheme.

(O-)DF protocol. With DF, for each transmission one node is transmitting at any given time and therefore the number of interfering nodes $n_{\mathtt{I}}$ is a binomial random variable in the range $[0,(\nu-2)/2]$ with average $\nu_{\mathtt{I}}$. Given that $n_{\mathtt{I}}$ nodes are interfering, for DF we have that the number of nodes available for cooperation is

$$n_{\mathtt{aC}} = \nu - 2n_{\mathtt{I}} - 2\,, \tag{4.61}$$

where for each interfering node we have considered its corresponding destination node as well.

MISO protocol. When the MISO protocol is considered, for each transmission one or two nodes may be transmitting, depending on whether or not cooperation has been activated. In this case $\nu_{\mathtt{I}}$ comprises both the number of source nodes and the number of nodes cooperating in other transmissions than the one originated by S. The number of interfering nodes $n_{\mathtt{I}}$ is a binomial random variable in the range $[0,\frac{\nu-2}{2}]$ with average $\nu_{\mathtt{I}}$. On the other hand, the number of cooperating nodes $n_{\mathtt{C}}$ is a binomial random variable in the range $[0,n_{\mathtt{I}}]$ with average

$$\nu_{\mathtt{C}|n_{\mathtt{I}}} = \frac{n_{\mathtt{I}}}{2}\left(\frac{\tau-T}{\tau}\right)p_{\mathtt{C}}^{(MISO)}\,, \tag{4.62}$$

where $p_{\mathtt{C}}^{(MISO)}$ is the average probability of cooperation, i.e., the average of $I_{\mathcal{E}}$ over all possible sets of interfering nodes and their positions. The number of nodes available for cooperation is therefore

$$n_{\mathtt{aC}} = \nu - 2n_{\mathtt{I}} - 2 - n_{\mathtt{C}}\,. \tag{4.63}$$

Once the number of nodes available for cooperation is generated, the cooperating node is selected. For the RS protocol, one of the nodes available for cooperation is selected at random. With the OPS protocol, we randomly generate $n_{\mathtt{aC}}$ node positions and determine the corresponding distances and outage probabilities $\mathrm{P}[\mathrm{out}\,i]$, $i=1,2,\ldots,n_{\mathtt{aC}}$. We then select the minimum, i.e., $\mathrm{P}[\mathrm{out}] = \min_i\{\mathrm{P}[\mathrm{out}\,i]\}$.

Steady state analysis

The steady state search algorithm is based on the computation of the average transmission duration. From (4.59) we obtain the average transmission duration assuming an average number of interfering nodes $\nu_{\mathtt{I}}$ and a cooperation probability $p_{\mathtt{C}}^{(MISO)}$. On the other hand, the average number of interfering nodes depends on the average transmission duration as follows.

Assuming an equal distribution of packet arrivals at all nodes, in steady state the average number of interfering nodes for the DF and MISO protocols are

$$\bar{\nu}_{\mathtt{I}}^{(DF)} = \lambda\tau^{(DF)} \tag{4.64a}$$

$$\bar{\nu}_{\mathtt{I}}^{(O-DF)} = \lambda\tau^{(O-DF)} \tag{4.64b}$$

$$\bar{\nu}_{\mathtt{I}}^{(MISO)} = \lambda\left[p_{\mathtt{C}}^{(MISO)}2(\tau^{(MISO)}-T) + (1-p_{\mathtt{C}}^{(MISO)})(\tau^{(MISO)}-T) + T\right]\,. \tag{4.64c}$$

```
1.  Set $\nu_{\min} = 0$, $\nu_{\max} = \frac{\nu-2}{2}$.
2.  Set $\nu_{\text{I}} = \frac{\nu_{\max}-\nu_{\min}}{2}$
3.  Compute the average duration by Monte Carlo method for the nodes distribution and using (4.59).
4.  Compute $p_{\text{C}}^{(MISO)}$ by averaging (4.38) over nodes positions.
5.  Obtain the average number of interfering nodes $\nu'_{\text{I}}$ from (4.65).
6.  If $(\nu_{\text{I}} > \nu'_{\text{I}})$
7.      $\nu_{\max} = \nu'_{\text{I}}$.
8.  Endif
9.  If $(\nu_{\text{I}} < \nu'_{\text{I}})$
10.     $\nu_{\min} = \nu'_{\text{I}}$.
11. Endif
12. If $\nu_{\text{I}} = \nu'_{\text{I}}$
13.     End
14. Else
15.     Goto 2
16. Endif
```

Table 4.2. *Dichotomic algorithm for the steady state analysis*

However, for some values of the network parameters, (4.64) may provide values greater than 1, which is an infeasible solution. This happens when the network is not able to support the offered traffic and some packets must be dropped. Therefore, the average number of interferers can be written as

$$\nu_{\text{I}} = \min\{1, \bar{\nu}_{\text{I}}\}\frac{\nu - 2}{2}\,. \tag{4.65}$$

In order to compute the steady state solution we use (4.59) to compute τ as a function of ν_{I} and (4.65) to compute ν_{I} as a function of τ. Then, we apply a dichotomic search algorithm that matches the average number of interfering nodes in (4.59) and (4.65). Note that $p_{\text{C}}^{(MISO)}$ must be computed at each iteration as it depends on the number of interferers and the positions of nodes that may cooperate, i.e., that do not operate either as sources or as destinations for other communications. The value of $p_{\text{C}}^{(MISO)}$ can be computed by averaging (4.38) over the positions of nodes available for cooperation.

The dichotomic algorithm is reported in Table 4.3.3.

4.3.4 Numerical results

The parameters of the network are reported in Table 4.3.4. All distances are normalized with respect to d_0. The average signal to noise ratio in the absence of interference at distance $d = d_0$ is 30 dB. Fig. 4.18 shows the contour lines of the average outage probability after frame $N = 30$ for fixed positions of nodes $\texttt{D}$ and $\texttt{S}$, as a function of the cooperator node position for the DF, O-DF and MISO techniques. We considered $n_{\text{I}} = 4$ interferers. The position of node $\texttt{S}$ corresponds to an average outage probability for the non cooperative

$N_{\max}$	30
$R_{\max}$	3
$R_{\min}$	1
κ	4
P_r	1
B	2 MHz
Γ (@ $d = 1$)	30 dB
T	1 ms
R	10 bit/s/Hz

Table 4.3. *Network parameters*

(NC) scheme $\mathrm{P}[\mathrm{out}|\mathrm{NC}] = 0.18$. Lines show results obtained with the analysis of this paper, while dots show simulation results, assuming interference with chi-square statistics. We first observe that analysis accurately matches simulations. We also observe that the outage probability is significantly affected by the position of node C, which for a wide area gives no advantage for cooperation. For DF, the best positions for the cooperator are near D as the success probability for forthcoming cooperative retransmissions is very high. Note that when the cooperator is further away from D than S, the outage probability is even higher than that of the NC system. O-DF and MISO techniques instead show a smoother behavior and the best location for the cooperator is around S as the outage probability is in this case dominated by the successful decoding of the first transmission.

The performance advantage of O-DF comes at the cost of additional signalling between nodes S and C in order to assess the link quality. On the other hand, MISO does not require additional signalling but yields an increased energy consumption as two nodes transmit simultaneously. Hence, from Fig. 4.18 it could be concluded that MISO is the best solution in terms of performance and signalling. However, the simultaneous activity of two nodes increases interference and reduces the nodes available for cooperation with other transmissions, thus affecting the overall network performance. We conclude therefore that a single link analysis does not provide enough information to assess the validity of the cooperation schemes. We therefore resort to the steady state analysis to obtain more comprehensive results. In the following we report the performance obtained for a network where source, cooperative and interfering nodes are spread on a ring centered on the destination node, with inner radius $R_{\min}$ and outer radius $R_{\max}$, as reported in Table 4.3.4.

In Fig. 4.19 we compare the average packet transmission duration τ for RS and OPS and NC methods as a function of the traffic intensity λ, with $\nu = 50$ nodes in the network. As expected, based upon considerations on Fig. 4.18, the RS protocol has an almost negligible advantage over the NC scheme, since in most cases the node selected for cooperation does not provide a lower outage probability than the S-D link. Therefore in the following we will consider only the OPS protocol, and compare it with the NC technique. For comparison

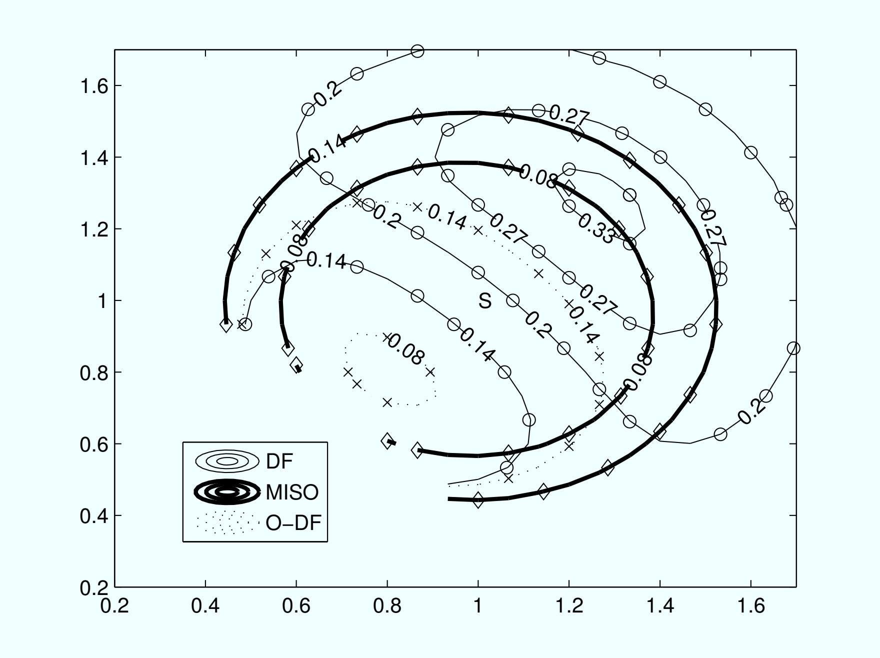


Figure 4.18. *Contour lines of the average outage probability as a function of the cooperator node position for DF, O-DF and MISO techniques. $n_{\mathrm{I}} = 4$. Node* S *is located at (1,1), while node* D *is at (0,0). $N = 30$. Lines: analysis results. Markers: simulation results.*

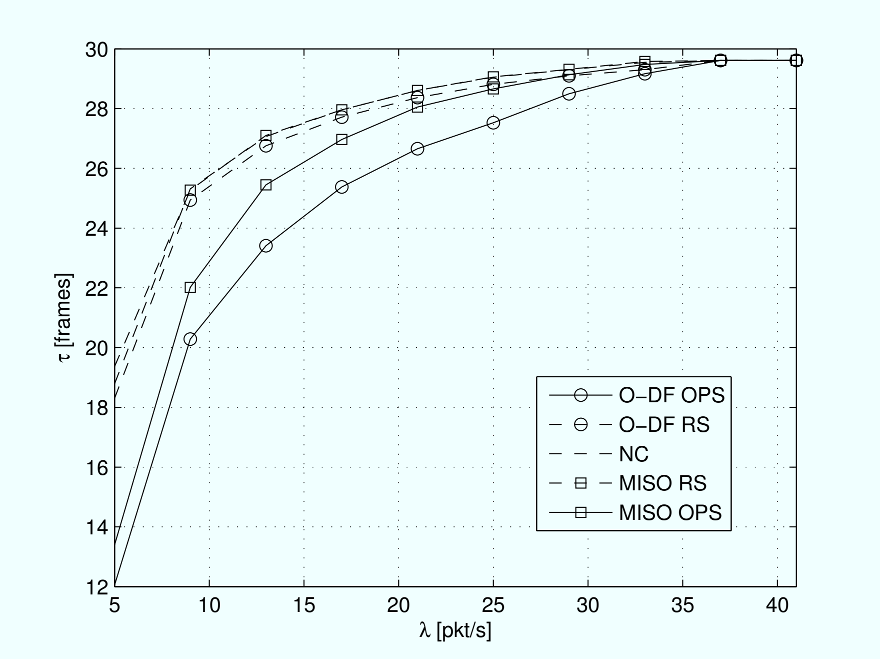


Figure 4.19. *Average packet transmission duration τ as a function of the traffic intensity λ. $\nu = 50$.*

purposes, we consider the effective throughput per node, defined as

$$T_{\mathrm{node}} = R \sum_{n=0}^{N_{\max}} \frac{1}{(n+1)} \mathrm{P}[\mathrm{succ}\, N = n] \,, \tag{4.66}$$

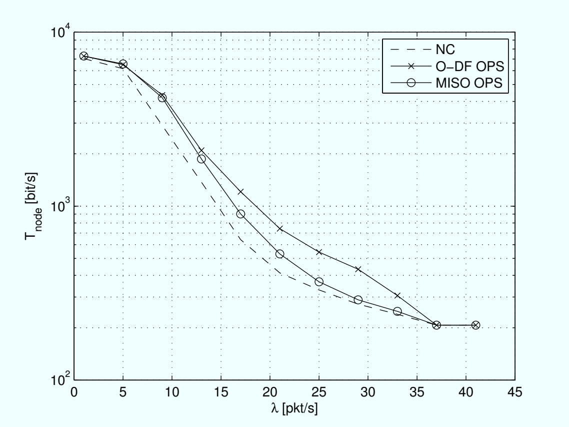


Figure 4.20. *Effective per node throughput as a function of the traffic intensity* λ. $\nu = 20$.

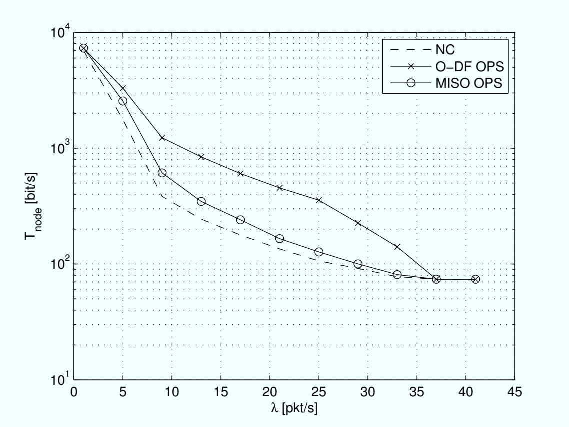


Figure 4.21. *Effective per node throughput as a function of the traffic intensity* λ. $\nu = 50$.

where $\mathrm{P}[\mathrm{succ}\, N = n]$ is the probability of successfully decoding the packet in exactly n frames.

Figs. 4.20, 4.21 and 4.22 show T_{node}, for various protocols, as a function of the traffic intensity λ, for networks with $\nu = 20$, $\nu = 50$ and $\nu = 100$ nodes, respectively. For $\nu = 50$, we observe that the O-DF-OPS protocol allows a dramatic increase of the effective throughput, while the MISO OPS protocol has a slightly better performance than the NC system. The good performance of O-DF OPS is due to the cooperator selection that allows to best exploit at network conditions. Indeed, as shown in Fig. 4.23, the O-DF OPS protocol achieves a very high success probability at the first retransmission, while the other schemes have a smoother behavior. On the other hand, MISO OPS suffers from both a suboptimal detection and a less efficient node usage as two nodes transmit simultaneously, thus reducing the number of available cooperators. We also note that the advantage of the cooperative schemes increases with the number of nodes in the network since more nodes provide an increased chance of finding a cooperator in a good position. On the other hand, as the traffic intensity increases, the average per node throughput decreases for all schemes. Asymptotically, for $\lambda \to \infty$, all schemes converge to the NC behavior, since no nodes are available for cooperation. Note also that in this case, if the offered traffic is more than the network can handle, packets are discarded as queues are not included in the analysis. Fig. 4.24 shows the effective per node throughput as a function of the number of nodes in the network, for an average traffic intensity $\lambda = 25$ pkt/s. The O-DF OPS protocol is able to achieve more than ten times the effective throughput of a NC system exploiting network conditions at best, as O-DF is largely insensitive to the number of network nodes. In fact, an increased number of nodes in the network yields an increased traffic but at the same time also an increased number of nodes available for cooperation, thus yielding faster transmissions.

Lastly, Fig. 4.25 shows the effective per node throughput as a function of the maximum number of retransmitted frames per packet, N_{max}, for a network with $\nu = 50$ nodes and a traffic intensity of $\lambda = 25$ pkt/s. We observe that both NC and MISO OPS protocols benefit

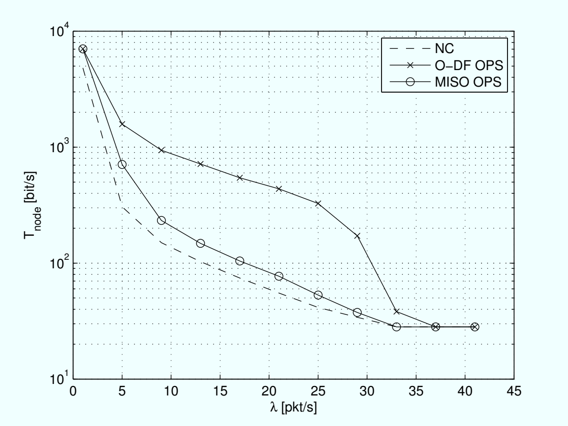


Figure 4.22. *Effective per node throughput as a function of the traffic intensity* λ. $\nu = 100$.

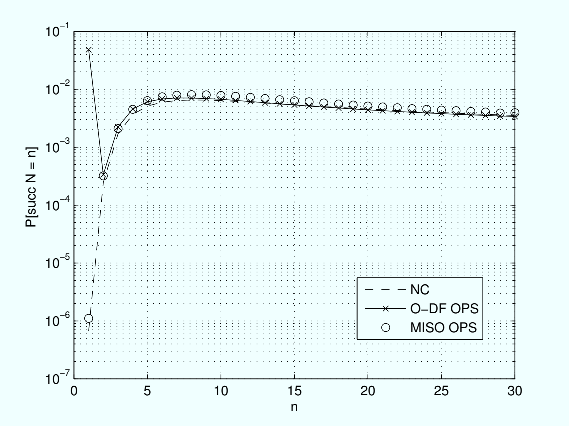


Figure 4.23. *Average success probability as a function of the number of retransmissions* n*, with* $\lambda = 25$ *[pkt/s] and* $\nu = 50$.

from a higher number of frames as the first retransmission may not always be successful. The O-DF OPS technique instead is characterized by a high probability of success at the first retransmission, due to the accurate choice of the cooperator, whereas subsequent frames provide a negligible advantage to the achieved per node throughput. We therefore conclude that for the O-DF OPS protocol the number of maximum retransmissions $N_{\max}$ can be limited to a smaller number than for NC and MISO protocols, as this would both decrease the average delay and increase the number of available cooperators, at the expense of a negligible decrease of the network throughput.

4.4 A Complete Network Framework

In this Section, we design a complete network using cooperation to improve overall reliability, throughput and efficiency.

In our view, the investigation of issues related to packet arrivals and queueing, and node activity is of fundamental importance. In fact, dynamic of interference in the network, and issues such as availability of nodes to cooperation, are of fundamental importance when considering cooperation for practical networking.

As in the previous Sections, we consider an hybrid ARQ, as its intrinsic efficiency can significantly improve performance in our scenario. In particular, we design an adaptive HARQ scheme, able to effectively counteract channel variations due to the start and end of interfering transmissions.

It is interesting to see how the possibility for simultaneous transmission makes faster control messages exchange and reduces the latency due to cooperation set up.

The rest of this part of the Chapter is organized as follows. In Section II we describe PHY, MAC and data link layers of the proposed network. Section III provides an analysis of the performance of LCCS for a simple network of three nodes. Numerical results obtained

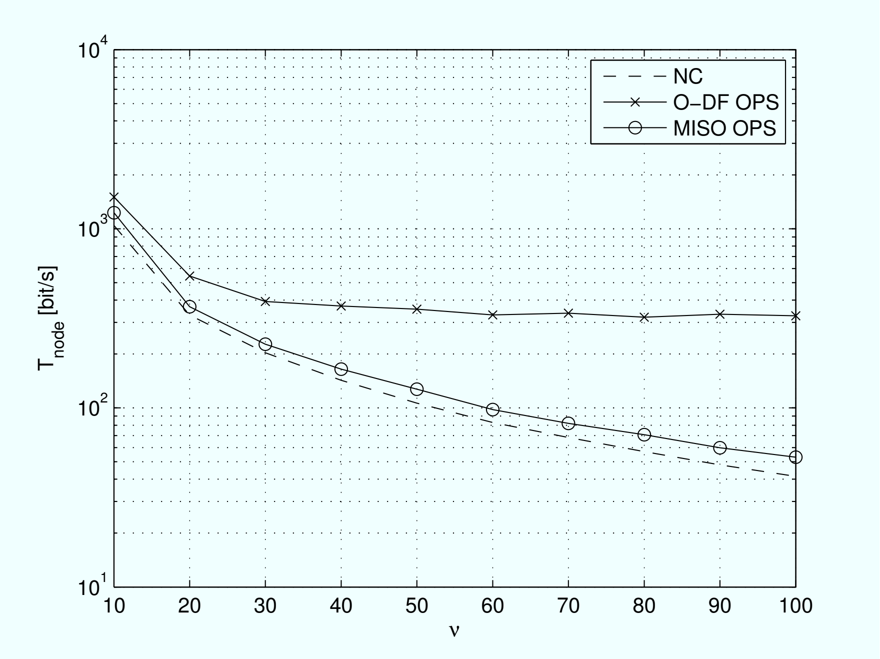


Figure 4.24. *Effective per node throughput as a function of the number of nodes in the network* ν*.* $\lambda = 25$ *pkt/s.*

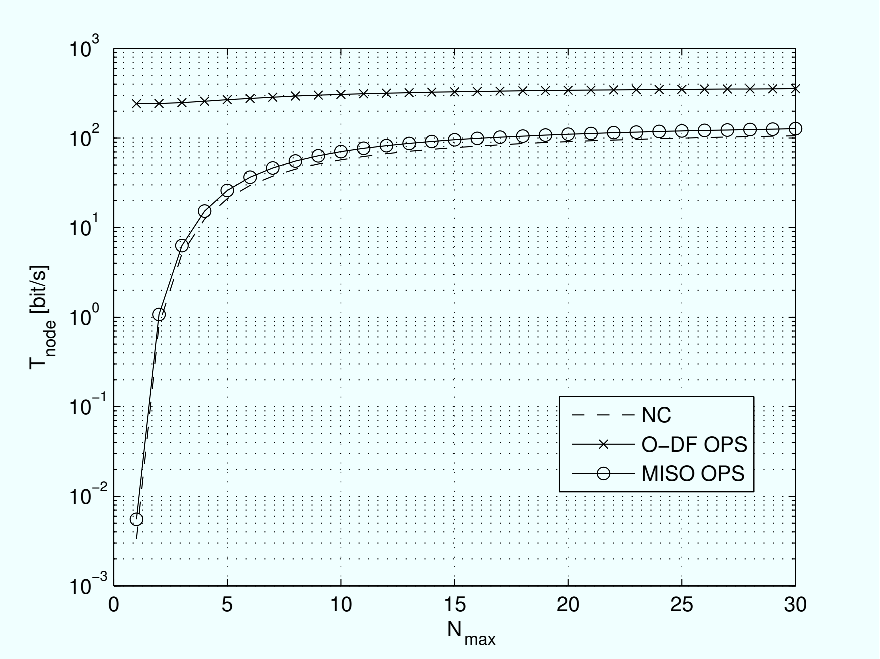


Figure 4.25. *Effective per node throughput as a function of the maximum number of frames per packet* $N_{\max}$*.* $\lambda = 25$ *[pkt/s] and* $\nu = 50$*.*

from this analysis, as well as from extensive simulations of a complete cooperative network with tens of nodes, are presented in Section IV.

4.4.1 System model

Spatial multiplexing and layered receiver

Each node has N antennas and is assigned a pseudo-random spreading sequence for CDMA. The bits to be transmitted are split into N streams, one for each antenna. Then, each stream is spread with the node-specific spreading sequence of N_S chips and all the antennas transmit simultaneously their spread stream on the channel.

The simultaneous transmission on all the antennas has two major consequences: a) the transmit power P_{TOT} must be split among the antennas, and b) the receiver sees the superposition of the signals coming from all the transmit antennas. As for the first point, we assume that P_{TOT} is uniformly divided among the N antennas so that each antenna transmits at a lower power than in the single antenna case, while the use of multiple transmit antennas yields an increased data rate. For the second issue of signal superposition, a LASTMUD receiver [25] is used. For the sake of a simple analysis, we consider a narrowband transmission, while the extension to a wideband scenario can be easily obtained by using, for example, orthogonal frequency division multiplexing (OFDM) with spreading in the time axis (see [26] for an overview).

Let us suppose that K nodes, denoted with indices $k = 1, 2, \dots, K$, are transmitting. By letting $\alpha^{(k,m)}$ be the power gain due to the path-loss from node k to node m, the complex channel gain from antenna i of node k to antenna j of node m can be written as $h_{i,j}^{(k,m)} = \sqrt{\alpha^{(k,m)}}g_{i,j}^{(k,m)}$, where $g_{i,j}^{(k,m)}$ is a random variable accounting for fading.

Let $\boldsymbol{S}$ be the $N_S \times K$ matrix collecting in its kth column the spreading code of node k. Let $\boldsymbol{H}_j^{(k,m)} = [h_{1,j}^{(k,m)}, h_{2,j}^{(k,m)}, \dots, h_{N,j}^{(k,m)}]$ be the N-size row vector of the complex channel gains from the N antennas of node k to antenna j of node m. We collect all channel vectors for antenna j of receive node m in the diagonal matrix $\boldsymbol{C}_j^{(m)}$, i.e. $\text{diag}\{\boldsymbol{C}_j^{(m)}\} = [\boldsymbol{H}_j^{(1,m)}, \boldsymbol{H}_j^{(2,m)}, \dots, \boldsymbol{H}_j^{(K,m)}]$. Let $\boldsymbol{d}$ be the column vector of the data symbols transmitted simultaneously by all N antennas of each of the K nodes and let us define the $N_S \times NK$ matrix $\boldsymbol{S}' = \boldsymbol{S} \otimes \boldsymbol{1}_N$, where $\otimes$ is the Kroneker product and $\boldsymbol{1}_N$ is a row vector of all ones and size N. The set of N_S chips received by node m on antenna j can be written as [25]

$$\boldsymbol{r}_j^{(m)} = \boldsymbol{S}'\boldsymbol{C}_j^{(m)}\boldsymbol{d} + \boldsymbol{\omega}_j^{(m)} = \boldsymbol{R}_j^{(m)}\boldsymbol{d} + \boldsymbol{\omega}_j^{(m)} , \tag{4.67}$$

where $\boldsymbol{\omega}_j^{(m)}$ is an N_S column vector of complex Gaussian noise samples with zero mean and power σ^2. A key component of our layered coded cooperative system (LCCS) is the LASTMUD receiver. The receive node m extracts a sufficient statistics for decoding by applying a matrix filter matched to the channel and summing the contributions of all the antennas. Following the derivations of [25], by defining the $KN \times KN$ correlation matrix

$$\tilde{\boldsymbol{R}}^{(m)} = \sum_{j=1}^{N} \boldsymbol{R}_j^{(m)H}\boldsymbol{R}_j^{(m)} , \tag{4.68}$$

the output of the matched receiver is the KN-size vector

$$\tilde{\boldsymbol{r}}^{(m)} = \tilde{\boldsymbol{R}}^{(m)}\boldsymbol{d} + \boldsymbol{n}^{(m)}\,, \tag{4.69}$$

where $\boldsymbol{n}^{(m)}$ is the filtered noise and ^{H} is the Hermitian operator. Then $\tilde{\boldsymbol{r}}^{(m)}$ is processed in KN stages. At each stage, decoding of an antenna signal is performed and its interference contribution on the received signal is generated. Before decoding a new antenna, the contributions of all previously decoded antennas are removed from the received signal, in order to reduce interference. Note that a receive node may decode packets even when they are intended for other receivers, in order to reduce interference on its own packets. Further details on the LASTMUD receiver can be found in [25].

Medium Access Control protocol

Traditional access protocols for ad hoc networks try to avoid collisions, i.e., simultaneous transmissions in the same neighborhood that would interfere and result in data loss and waste of resources. In LCCS, instead, simultaneous transmissions are possible thanks to CDMA and SM and thus we design the MAC protocol with the aim of increasing the network throughput.

Transmission is organized in *time slots* and each data packet may span multiple slots. Each slot comprises a) a short training sequence used by receive nodes to estimate the channel, b) a header identifying the source and destination nodes, and c) control or data bits.

Before data transmission, a handshake phase is established between the source and destination nodes. First, the source sends a Request Packet (RP), which contains the identifiers of source, destination and packet, as well as the duration of the requested transmission. If the destination node is not already involved in another communication and successfully decodes RP, it responds with a Grant Packet (GP) in the following slot. Note that a node may receive several requests in a single slot and many granting policies could be implemented. For instance, selecting the request with the maximum received power is likely the best for link reliability, but may be unfair. However, the exploration of the best granting policy is beyond the scope of this paper, and we consider a random choice among the received requests. The handshake avoids transmission of long data packets to unreachable destinations. For the same purpose, nodes keep an *occupancy table* with the expected number of slots before a neighbor becomes idle and delay RP transmissions accordingly. Information on nodes' activity is extracted from any detected packet, including packets detected for interference cancellation purposes.

After the handshake, the source transmits the packet as described in Section 4.4.1. The receiver reports the success or failure of the packet decoding through Feedback Packets (FP) of acknowledge (ACK) or non-acknowledge (NACK). We assume that transmission of control and data packets is performed in time-adjacent slots and that the delays due to processing and propagation are negligible.

We also assume that each control packet (RP, GP and FP) fits the duration of a single slot and therefore has a limited impact on communication efficiency. Still, their reception is of paramount importance for network performance, thus we assume that they are transmitted using only one antenna at full power P_{TOT}, and are protected by a convolutional code of rate $1/2$ with generator polynomial given in Table 4.4.

Note that we exploit SM and CDMA for multiple access so that nodes do not perform carrier sensing before accessing the channel. For the same reason, unlike in the IEEE 802.11 distributed coordination function, in our protocol the reception of a handshake packet does not prevent nodes from accessing the channel for the duration of the forthcoming data transmission. Still, since the number of simultaneous transmissions is limited by the spreading factor and by the spatial properties of the channel, we include in the protocol a backoff mechanism that limits repeated transmission attempts. Link failures (i.e., GP or FP is not received, or a NACK is reported at the end of the transmission) force a source node to defer any other transmission for a number of slots B, randomly chosen in the exponentially increasing window $[1, 2^{N_{\text{fail}}} \times W]$, where W is the initial window value and N_{fail} is the minimum between the number of consecutive failures and $Max_{N_{\text{fail}}}$. At each correct detection, N_{fail} is decreased by one unlike in conventional backoff in which it is reset to zero. Preliminary simulations show that this window policy achieves a better performance and is more adaptive to network conditions than the standard technique. A detailed analysis of this issue is left for future study.

HARQ protocol

The performance of decoding algorithms is highly dependent on the received power, the number of incoming signals and the channel conditions. All these quantities may change in each slot and the estimation of the channel coding rate needed to successfully transmit cannot be performed in the handshake phase. In order to counteract variations in link conditions, we include in our protocol an adaptive HARQ error control scheme that combines the benefits of forward error correction (FEC) and ARQ providing *incremental redundancy*. We want to stress that the proposed protocol admits several different implementations. In this and the following subsections we outline the general characteristics of both HARQ and cooperative HARQ, whereas in Section 4.4.1 a specific implementation is proposed and discussed.

Each data packet is encoded with an error correcting code and transmission is performed in several FEC phases, each providing a different portion of the coded packet. In particular, at FEC phase p, a subset $\mathcal{F}_p$ of the coded bits is transmitted using one or more slots. We will discuss the choice of $\mathcal{F}_p$ and its size in Section 4.4.1. Together with $\mathcal{F}_p$, in each phase the source node transmits checksum bits that allow the destination to understand whether or not that subset has been received correctly. In the next slot, the destination replies with an ACK or NACK, accordingly. An ACK causes the source to stop FEC phases. After a NACK

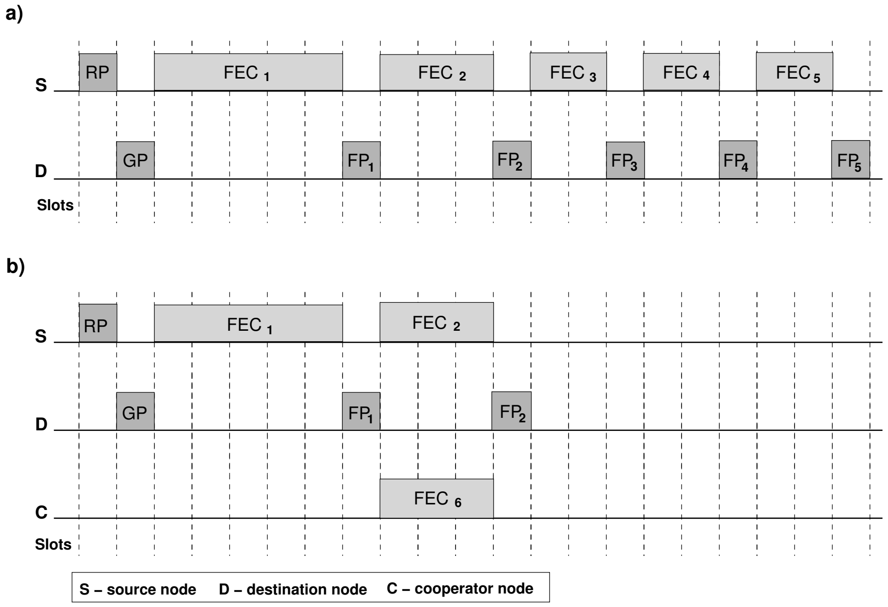


Figure 4.26. *Example of transmission for the non-cooperative (a) and the cooperative (b) protocol.*

at the pth FEC phase, the source transmits $\mathcal{F}_{p+1}$. If M_{fec} FEC phases have been performed and the destination still reports a failure, the source defers the transmission of the packet by B slots. After M_{tx} failed packet transmission attempts, the packet is dropped.

Fig. 4.26 (a) shows an example of operation of the proposed adaptive HARQ scheme, where the source performs five FEC phases. The proposed HARQ scheme effectively adapts the coding rate to the channel conditions, providing incremental redundancy when the destination fails to decode the packet with the previously received parts of the codeword. The source reschedules a new delivery attempt after the failure of M_{fec} FEC phases in order to diminish the load of the network, refraining from further redundancy transmissions under bad channel conditions.

Cooperative HARQ protocol

Cooperation is well known to provide advantages due to spatial diversity obtained by relaying data through cooperative nodes. In our protocol, the destination node already decodes multiple streams coming from the antennas of the source as well as interfering nodes. Therefore, the proposed PHY and MAC layers can be easily adapted for cooperation by letting cooperative nodes transmit incremental redundancy simultaneously with the source. In particular, idle nodes that correctly decode the packet at the first FEC phase and receive a NACK from the destination, may cooperate by re-encoding the packet and transmitting redundancy to the destination node. Cooperation is dismissed at the reception of an ACK or after M_{fec} FEC phases. Observe that cooperative and source transmissions are simultaneous and have the same duration, although the transmitted subsets $\mathcal{F}_p$ may be different for the two transmitting nodes.

In Fig. 4.26 (b), an example of cooperative transmission is shown. It is important to remark that the channels of source and cooperative nodes are independent. Thus, in slow fading environments, the cooperative nodes may resolve the delivery more efficiently than the source. Note also that cooperation becomes active only upon a NACK, thereby avoiding unnecessary FEC phases.

Cooperator choice. In order to optimize cooperation, we let nodes cooperate only when they have a better chance than the source of being correctly decoded by the destination. In particular, each candidate node compares the signal to noise plus interference ratio (SNIR) of its own link to the destination ($SNIR_{CD}$) with the SNIR of the source-destination link ($SNIR_{SD}$) and cooperates only if $SNIR_{CD} > SNIR_{SD}$. For this purpose, the destination node includes an estimate of $SNIR_{SD}$ in each GP while $SNIR_{CD}$ is estimated through FP.

Multiple cooperative nodes. Our cooperation scheme is completely transparent to the source, and does not require additional signaling or negotiation, since the receiver is able to identify useful sub-packets by header inspection. Several nodes may cooperate for the same transmission with no need for coordination, although too many packets may overload the receivers.

It is important to observe that our protocol enables nodes to establish new links while other communications are active in their neighborhood. Thus, a drawback of our protocol is that cooperative nodes are deaf to incoming RP packets when transmitting, since we consider half-duplex terminals. Moreover, transmission of packets originated in cooperative nodes is delayed for the duration of the cooperation. This may result in throughput loss and delay degradation at the cooperative nodes. On the other hand, by providing channel diversity, cooperation yields an improved efficiency and, consequently, a lower average interference. Simulation results will show that the balance between drawbacks and advantages is in favor of cooperation for the considered scenario.

In the following, the non-cooperative protocol is referred to as layered coded system (LCS), while the cooperative protocol is referred to as LCCS.

Implementation by packet coding

In this Section we propose an implementation of the HARQ protocol through linear erasure codes (LEC), specifically designed for error control in packet networks. Examples of LEC are the codes of [27] and the burst erasure correction codes with low decoding delay of [28], both of which have been recently studied in a networking context in [29]. Packet coding is particularly useful in transmissions affected by bursty interference, as in our system where several nodes transmit simultaneously and interference affects entire packets.

A LEC of rate r_c is defined by the triple $(U, r_c, \boldsymbol{G})$, where U is an integer and $\boldsymbol{G}$ is a $(U/r_c) \times U$ *generator matrix*, with elements taken from the Galois field GF(2^b). For the encoding of a data packet of L_{pkt} bits, the bits are first grouped into symbols of b bits each and then split into U blocks $\boldsymbol{a}_u$, $u = 1, 2, \ldots, U$ of $M = L_{pkt}/(bU)$ symbols each. The LEC encoder generates $L = U/r_c$ coded blocks as follows

$$\boldsymbol{b}_\ell = [\boldsymbol{G}]_{\ell,1}\boldsymbol{a}_1 \oplus [\boldsymbol{G}]_{\ell,2}\boldsymbol{a}_2 \oplus \ldots \oplus [\boldsymbol{G}]_{\ell,U}\boldsymbol{a}_U \,, \tag{4.70}$$

where $\ell = 1, 2, \ldots, L$ and $\oplus$ denotes the element-wise sum in GF(2^b) of vectors. In the following, the coded blocks $\boldsymbol{b}_\ell$ will be denoted as *sub-packets* of the data packet. Without restriction, we consider a systematic LEC for which $\boldsymbol{b}_\ell = \boldsymbol{a}_\ell$ with $\ell = 1, 2, \ldots, U$.

An interesting property of any LEC having full-rank generating matrix is that if any U out of the L sub-packets $\boldsymbol{b}_\ell$ are correctly decoded, then the entire packet can be recovered [27]. Let $\mathcal{C} = \{\kappa_1, \kappa_2, \ldots, \kappa_U\}$ be the indices of the correctly decoded sub-packets and let $\bar{\boldsymbol{G}}_{\mathcal{C}}$ be the matrix containing the columns of $\boldsymbol{G}$ with index in $\mathcal{C}$. Since $\boldsymbol{G}$ is full-rank, $\bar{\boldsymbol{G}}_{\mathcal{C}}$ is also full rank and can be inverted in GF(2^b), to obtain $\bar{\boldsymbol{G}}_{\mathcal{C}}^{-1}$. By combining the decoded sub-packets with $\bar{\boldsymbol{G}}_{\mathcal{C}}^{-1}$, we obtain the original U data blocks

$$\boldsymbol{a}_u = [\bar{\boldsymbol{G}}_{\mathcal{C}}^{-1}]_{u,1}\boldsymbol{b}_{\kappa_1} \oplus [\bar{\boldsymbol{G}}_{\mathcal{C}}^{-1}]_{u,2}\boldsymbol{b}_{\kappa_2} \oplus \ldots \oplus [\bar{\boldsymbol{G}}_{\mathcal{C}}^{-1}]_{u,U}\boldsymbol{b}_{\kappa_U} \,, \tag{4.71}$$

where $u = 1, 2, \ldots, U$.

According to the general description of the previous Section, we consider that each subset $\mathcal{F}_p$ is an ensemble of sub-packets $\boldsymbol{b}_\ell$. In particular, $\mathcal{F}_1 = \{\boldsymbol{b}_\ell\}_{\ell=1,2,\ldots,U}$. Then $\mathcal{F}_p$, $p = 2, \ldots, M_{\text{fec}}$, contains a variable number of sub-packets, according to the number of sub-packets that have been correctly decoded by the destination in the previous FEC phases. In particular, if the destination at FEC phase p has decoded ν_p sub-packets, with $\nu_p < U$, the NACK contains the number of missing sub-packets $H_p = U - \nu_p$, and the indices of the correctly decoded sub-packets. At FEC phase $p+1$, $\mathcal{F}_{p+1}$ contains H_p randomly selected sub-packets, excluding those correctly decoded. Note that if the NACK does not include the indices of the correctly decoded sub-packets, the same sub-packet may be re-transmitted, even if it has already been correctly decoded by the destination. Therefore, the signaling overhead is reduced at the expense of a slight coding gain reduction.

While both the source and the cooperative nodes follow the same rules for the choice of the length of the FEC sets, the indices of the transmitted sub-packets is in general different. Still, since the transmissions from the source and the cooperative node are not coordinated, the destination may receive multiple copies of the same sub-packet. In this case the protocol incurs a slight loss of efficiency, that could be avoided by devising more sophisticated solutions, left for future study.

4.4.2 Analysis of LCCS

In order to gain some fundamental insight into the behavior of LCCS we analyze a simple network with a source node S, a destination node D, and a possible cooperative node C. This kind of analysis has been commonly used in the literature for assessing the performance of cooperative protocols [1, 2, 12]. Node C may be in one of three states for the duration of an entire session of FEC phases between S and D: a) *idle state*, when it does not cooperate

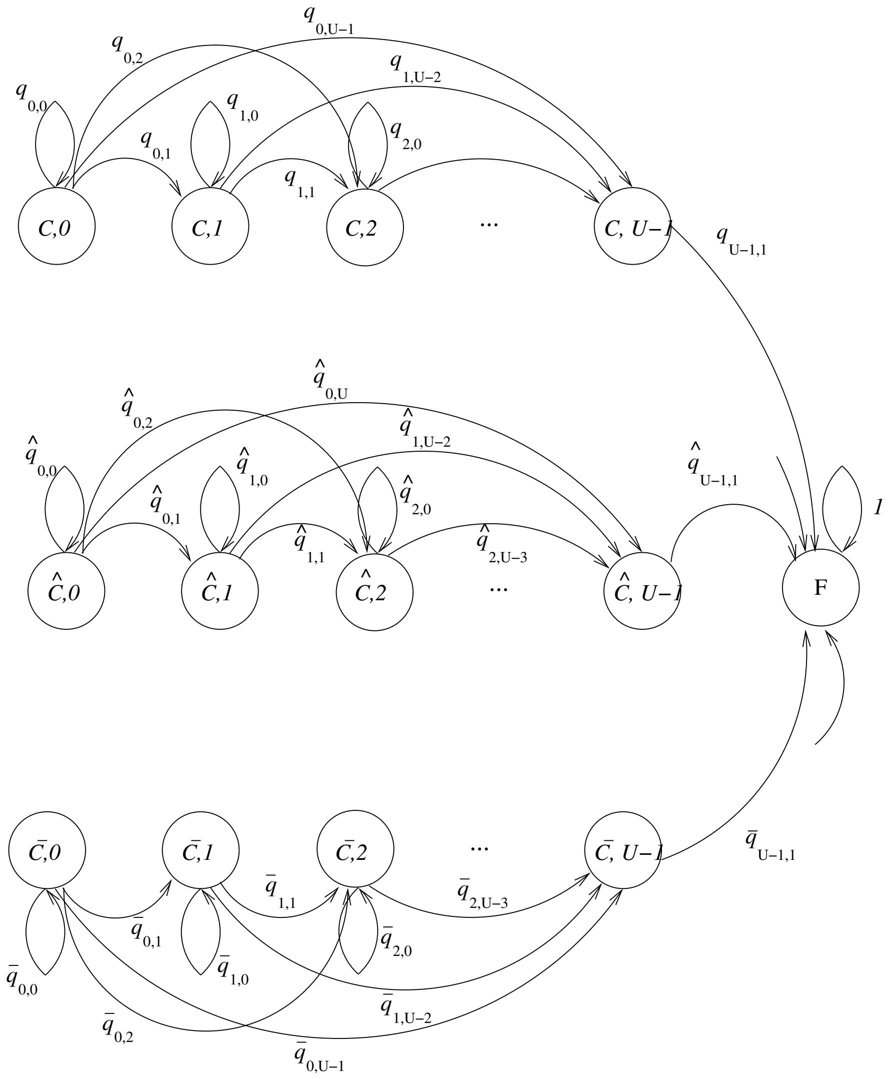


Figure 4.27. *Example of Markov chain for LCCS.*

because it failed to decode the first FEC phase and it does not transmit to any other node; *b*) *cooperation state*, when it cooperates with S; and *c*) *busy state*, when it is transmitting to a node other than D. The busy state occurs with probability π_{B} and in this case node C interferes with the transmission between S and D. We do not consider here the effect of the presence of other nodes (e.g., nodes communicating with C when it is busy).

The behavior of this simple LCCS network using LEC for HARQ can be described by the Markov chain of Fig. 4.27, where states identify both the number of correctly decoded sub-packets ν at the end of each FEC phase and the status of node C, where (C, ν), $(\bar{C}, \nu)$ and $(\hat{C}, \nu)$ refer to cooperation, idle and busy status, respectively. The Markov chain has also a state F, which corresponds to the correct decoding of U sub-packets, i.e., the packet is correctly received. The first FEC phase sets the initial state of the Markov chain. Let p_ν, $\bar{p}_\nu$ and $\hat{p}_\nu$ be the probabilities of being in state (C, ν), $(\bar{C}, \nu)$ and $(\hat{C}, \nu)$ at the end of the first FEC phase, respectively. In particular, let z_ν, be the probability that node D has correctly decoded ν sub-packets in the first FEC phase, conditioned on the fact that C is not busy.

As we mentioned above, we assume that node C cooperates only if it has decoded U sub-packets, i.e., it is able to decode the entire packet. Let π_{C} be the probability that node C decodes all U sub-packets. The probability of being in state (C, ν) at the end of the first FEC phase is $p_\nu = z_\nu \pi_{\mathrm{C}}(1-\pi_{\mathrm{B}})$, $\nu = 0, 1, \ldots, U-1$. On the other hand, the probability that node D has correctly decoded ν sub-packets and node C is neither collaborating nor interfering is $\bar{p}_\nu = z_\nu(1-\pi_{\mathrm{B}})(1-\pi_{\mathrm{C}})$, $\nu = 0, 1, \ldots, U-1$.

Given that node C is busy, let $\hat{z}_\nu$, $\nu = 0, 1, \ldots, U$ be the conditional probability that node D correctly decodes ν sub-packets at the end of the first FEC phase, then the probability of being in state $(\hat{C}, \nu)$ at the end of the first FEC phase is $\hat{p}_\nu = \hat{z}_\nu \pi_{\mathrm{B}}$, $\nu = 0, 1, \ldots, U-1$. The probability that node D decodes all U sub-packets in the first FEC phase is $\tilde{p} = (1-\pi_{\mathrm{B}})z_U + \pi_{\mathrm{B}}\hat{z}_U$.

The next FEC phases correspond to transitions in the Markov chain of Fig. 4.27 and the state is updated according to the number of new sub-packets decoded by D in each phase. For a cooperative transmission, let $q_{\nu,\mu}$ be the probability that μ sub-packets are decoded out of $U-\nu$ sub-packets transmitted by node S and $U-\nu$ sub-packets transmitted by node C. Note that $q_{\nu,\mu}$ has a different statistical description from p_ν, since in the first FEC phase only S transmits, while $q_{\nu,\mu}$ accounts for simultaneous transmissions from both S and C. When node C is idle, $\bar{q}_{\nu,\mu}$ is the transition probability from state $(\bar{C}, \nu)$ to state $(\bar{C}, \nu+\mu)$. When node C is busy with another transmission, the transition probabilities from state $(\hat{C}, \nu)$ to state $(\hat{C}, \nu+\mu)$ are indicated as $\hat{q}_{\nu,\mu}$. Lastly, transition probabilities from states (C, ν), $(\bar{C}, \nu)$, $(\hat{C}, \nu)$ to state F are collected into the U-size column vectors $\boldsymbol{t} = [q_{0,U}, q_{1,U-1}, \ldots, q_{U-1,1}]^T$, $\bar{\boldsymbol{t}} = [\bar{q}_{0,U}, \bar{q}_{1,U-1}, \ldots, \bar{q}_{U-1,1}]^T$ and $\hat{\boldsymbol{t}} = [\hat{q}_{0,U}, \hat{q}_{1,U-1}, \ldots, \hat{q}_{U-1,1}]^T$, respectively. ^{T} denotes the transpose operator.

From the Markov chain of Fig. 4.27 we capture two distinctive features of LCCS, namely a) the cooperative behavior, which is reflected into the sub-chain (C, ν), and b) HARQ, which is reflected into the U states of each sub-chain. Moreover, in LCCS there may be simultaneous transmissions to different destinations and this is reflected by the probability that node C is busy.

Let $\boldsymbol{T}$ be a $U \times U$ matrix with entries

$$[\boldsymbol{T}]_{\ell_1,\ell_2} = \begin{cases} q_{\ell_1-1,\ell_2-\ell_1} & \ell_1, \ell_2 = 1, 2, \ldots, U,\ \ell_1 \geq \ell_2 \\ 0 & \text{otherwise} \end{cases} \tag{4.72}$$

and let $\bar{\boldsymbol{T}}$ and $\hat{\boldsymbol{T}}$ be defined analogously to $\boldsymbol{T}$ with $\bar{q}_{\ell_1-1,\ell_2-\ell_1}$ and $\hat{q}_{\ell_1-1,\ell_2-\ell_1}$ instead of $q_{\ell_1-1,\ell_2-\ell_1}$

The evolution of the Markov chain is governed by the transition probability matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0}_{U\times U} & \boldsymbol{0}_{U\times U} & \boldsymbol{t} \\ \boldsymbol{0}_{U\times U} & \bar{\boldsymbol{T}} & \boldsymbol{0}_{U\times U} & \bar{\boldsymbol{t}} \\ \boldsymbol{0}_{U\times U} & \boldsymbol{0}_{U\times U} & \hat{\boldsymbol{T}} & \hat{\boldsymbol{t}} \\ \boldsymbol{0}_{1\times U} & \boldsymbol{0}_{1\times U} & \boldsymbol{0}_{1\times U} & 1 \end{bmatrix}. \tag{4.73}$$

Let also $\boldsymbol{s}(u) = [\boldsymbol{s}_C(u)\,, \boldsymbol{s}_{\bar{C}}(u)\,, \boldsymbol{s}_{\hat{C}}(u)\,, s_F(u)]$ be the $3U+1$ row vector of state probabilities at the uth FEC phase, where $\boldsymbol{s}_C(u)$ is the U row vector containing the probabilities of states $(C,0),(C,1),\ldots,(C,U-1)$. $\boldsymbol{s}_{\bar{C}}(u)$ and $\boldsymbol{s}_{\hat{C}}(u)$ are defined analogously to $\boldsymbol{s}_C(u)$ with states $(\bar{C},\nu)$ and $(\hat{C},\nu)$, respectively, instead of (C,ν). Lastly, $s_F(u)$ is the probability of being in state F. At the end of the first FEC phase, the state probabilities are

$$\boldsymbol{s}(1) = [p_0,\ldots,p_{U-1},\hat{p}_0,\hat{p}_1,\ldots,\hat{p}_{U-1},\bar{p}_0,\ldots,\bar{p}_{U-1},\tilde{p}]\,, \tag{4.74}$$

while at end of FEC phase u the state probabilities are $\boldsymbol{s}(u) = \boldsymbol{s}(u-1)\boldsymbol{P}$, $u = 2,3,\ldots,M_{\text{fec}}$.

Since at each FEC phase node S (and C) retransmits in sequence a number of sub-packets equivalent to those that have not been decoded, the average number of sub-packets transmitted by the source after u FEC phases is

$$\tau(u) = U + \sum_{v=1}^{u-1}\sum_{\ell=1}^{U}(U-\ell+1)\{[\boldsymbol{s}_C(v)]_\ell + [\boldsymbol{s}_{\bar{C}}(v)]_\ell + [\boldsymbol{s}_{\hat{C}}(v)]_\ell\}\,,$$

where $\tau(1) = U$. Hence, the average throughput after u FEC phases is

$$T(u) = b_p \frac{s_F(u)}{T_s \cdot S_s \cdot \tau(u)}\,, \tag{4.75}$$

where T_s is the symbol period, S_s is the number of symbols per sub-packet and b_p is the number of data bits of one packet.

Transition probabilities computation

We derive the transition probabilities of the LCCS Markov chain under the following assumptions: a) error propagation is neglected; b) Rayleigh fading is assumed with unit power; c) channels do not change within a sub-packet transmission and are independent for each sub-packet (*block fading*) and d) BPSK modulation is used.

We assume that signals coming from different nodes are almost orthogonal, due to CDMA, so that the pseudo-inverse of (4.68) is equal to the pseudo-inverse of the correlation matrix obtained without detection of the interfering node. Still, we model the residual interference as a random vector of size NN_S. This vector is projected on the vector obtained by matched filtering (4.69) and despreading [25]. Assuming the spreading factor is large enough, the resulting variable is complex Gaussian distributed with zero mean and variance $1/(NN_S)$. Its squared magnitude has probability density function (pdf) $f_I(y) = NN_S e^{-NN_S y}$, $y \geq 0$. Suppose that node D attempts to decode signals coming from a node with path loss α, under the interference from a node with path loss β. The probability of correctly decoding a block of M/N BPSK symbols at the LASTMUD stage θ is [30]

$$\zeta(\theta,\alpha,\beta) = \int_0^\infty \int_0^\infty \left[1 - Q\left(\sqrt{\frac{x\alpha^2}{y\beta^2+\sigma^2}}\right)\right]^{M/N} f_H(x,\theta) f_I(y)\,dx\,dy \tag{4.76}$$

where $\theta = 1, 2, \ldots, N$, $Q(\cdot)$ is the complementary normalized Gaussian distribution function and $f_H(x, \theta)$ is the pdf of the power gain x relative to the antenna decoded at stage θ of LASTMUD. It has been shown in [30] that x is chi-square distributed with 2θ degrees of freedom, i.e.

$$f_H(x,\theta) = \left[2^{\theta}\Gamma\left(\theta\right)\right]^{-1} x^{\theta-1}e^{-x/2}\,, \quad x \geq 0\,. \tag{4.77}$$

The probability of decoding μ sub-packets out of the ℓ transmitted is

$$\begin{aligned}\varphi(\alpha,\beta,\ell,\mu) =& \binom{\ell}{\mu}\left[\prod_{\theta=1}^{N}\zeta(\theta,\alpha,\beta)\right]^{\mu}\\ &\left[1-\prod_{\theta=1}^{N}\zeta(\theta,\alpha,\beta)\right]^{\ell-\mu}\,.\end{aligned} \tag{4.78}$$

Let $\alpha^{(S,D)}$ be the path loss between S and D and $\alpha^{(C,D)}$ the path loss between node C and D. Let $\alpha^{(S,C)}$ be the path loss between S and C. We also assume that when C cooperates, node D first decodes signals coming from C and then those coming from S. In this case, the reception of sub-packets from C is interfered by S, while, assuming perfect cancellation in the LASTMUD receiver, the reception of sub-packets from S has no interference.

First FEC phase. In the first FEC phase U sub-packets are transmitted by the source and we have $z_\nu = \varphi(\alpha^{(S,D)}, 0, U, \nu)$ and $\pi_{\mathrm{C}} = \varphi(\alpha^{(S,C)}, 0, U, U)$. If node C is busy, we have $\hat{z}_\nu = \varphi(\alpha^{(S,D)}, \alpha^{(C,D)}, U, \nu)$.

Non-cooperative FEC phases. Suppose that ν sub-packets have been correctly received, assuming that the source transmits $U - \nu$ sub-packets, the transition probability from state $(\bar{C}, \nu)$ to state $(\bar{C}, \nu + \mu)$ is $\bar{q}_{\nu,\mu} = \varphi(\alpha^{(S,D)}, 0, U - \nu, \mu)$, where $\mu = 0, 1, \ldots, U - \nu - 1$. If C is busy, we obtain $\hat{q}_{\nu,\mu} = \varphi(\alpha^{(S,D)}, \alpha^{(C,D)}, U - \nu, \mu)$, where $\mu = 0, 1, \ldots, U - \nu - 1$.

Cooperative FEC phases. If C cooperates, both S and C transmit $U - \nu$ sub-packets. Assuming that D first decodes signals coming from C and then those coming from S, the transition probabilities can be written as

$$\begin{aligned}q_{\nu,\mu} =& \sum_{\ell=0}^{\mu}\varphi(\alpha^{(C,D)}, \alpha^{(S,D)}, U-\nu, \ell)\\ &\varphi(\alpha^{(S,D)}, 0, U-\nu, \mu-\ell)\,,\end{aligned} \tag{4.79}$$

where $\mu = 0, 1, \ldots, U - \nu - 1$.

4.4.3 Numerical results

In this Section we present and discuss the performance of the system proposed in Section 4.4.1. Table 4.4 summarizes the values of all the parameters. We consider a transmission in the 5 GHz ISM frequency band and BPSK modulation with 99% in–band power, so that packet transmission may afford up to $7.5/N_S$ Mbps per used tx antenna and the block channel variation assumption is reasonable.

PHY Parameters	value
Modulation	BPSK
Antennas per node N	2
Spreading factor N_S	16
Bit-rate per antenna B_w	468.75 kbps
Operating band	5.8 GHz ISM
P_{TOT}	0.25W
Data packet length (L_{pkt})	4096 bits
Erasure code parameters (U,r_c)	(8, 1/3)
Block length ($M = L_{pkt}/U$)	512 bits
Noise power σ^2	-170 dBm
Signaling FEC polynomial (rate 1/2)	$133_8, 171_8$

HARQ parameters	value
M_{tx}	8
M_{fec}	10
$Max_{N_{\mathrm{fail}}}$	6

Simulation parameters	value
Number of nodes (K_{tot})	36
Network topology	150 × 150 m square grid
Channel correlation ρ	0.9
Number of simulated slots	400000

Queue and backoff parameters	value
Queue timeout Q_{TO}	4096 slots
Queue length	Q_{pkt} pkt
W	4

Table 4.4. *Parameters for simulation*

Analytical results

The performance of the network of Section 4.4.2 is evaluated in terms of throughput (4.75) as a function of the average SNR between S and D, $SNR^{(S,D)} = \alpha^{(S,D)}/\sigma^2$, and the average SNR between C and D, $SNR^{(C,D)} = \alpha^{(C,D)}/\sigma^2$, while we assume that the average SNR between S and C is fixed $(SNR^{(S,C)})_{\text{dB}} = 10\log_{10}[\alpha^{(S,C)}/\sigma^2] = 20$ dB. The channel is assumed block-fading with independent Rayleigh fading coefficients.

The results are obtained with PHY layer and HARQ parameters as indicated in Table 4.4. In the throughput computation the overhead due to sub–packet header and checksum is neglected. We compare the performance of LCCS and LCS. Fig. 4.28 shows the isometric lines of the average throughput in kbps as a function of $SNR^{(S,D)}$ and $SNR^{(C,D)}$ for $\pi_{\text{B}} = 0.2$. We plot the performance of LCCS (solid lines) and LCS (dashed lines) and we note that LCCS gains about 1 dB in $SNR^{(S,D)}$ when $SNR^{(C,D)}$ is in the range 5 to 15 dB. For a given $SNR^{(S,D)}$ and for an increasing $SNR^{(C,D)}$ we observe that initially the throughput increases, thanks to the advantage provided by cooperation. However, for very high $SNR^{(C,D)}$ we incur a throughput degradation due to the interference caused by node C when it is busy. Lastly, note that LCS sees a performance degradation as $SNR^{(C,D)}$ increases, even if it is not cooperating. This is due to the fact that when C is busy in another transmission, it interferes with the communication between S and D, and the interference power increases with the channel gain between C and D, i.e., with $SNR^{(C,D)}$. Fig. 4.29 shows the average throughput as a function of the busy probability π_{B} for $SNR^{(S,D)} = 6$ dB, $SNR^{(C,D)} = 10$ dB and $SNR^{(S,C)} = 20$ dB. We observe that as the busy probability increases the network becomes less cooperative.

Simulation results

A main goal of this paper is to design a cooperative system that works effectively in a simultaneous access network environment. Most of the literature, with few exceptions such as [31], shows performance for small networks with very few nodes. A distinctive feature of this work is that performance is assessed through the simulation of MAC/PHY layers of a network with tens of nodes, as reported in Table 4.4. We set up a one–hop fully connected $L \times L$ sized grid network with K_{tot} nodes, providing a maximum distance between nodes equal to the coverage range measured without interference. The packet arrival for each node is a Poisson process of rate λ [pkt/s]. Packets are served with a First In–First Out (FIFO) policy and are discarded after a maximum of Q_{TO} slots spent in the queue. The queue stores at most Q_{pkt} packets. Note that each of the K_{tot} nodes may cooperate or operate as source or destination, and accesses the channel without any restriction but the backoff mechanism. This scenario allows to assess the protocol efficiency in balancing the resources since nodes have the option to either cooperate or transmit their own packets.

The PHY layer and the access protocol are those considered in Sections 4.4.1 and 4.4.1, respectively. By preliminary simulations we have selected the spreading factor in order to

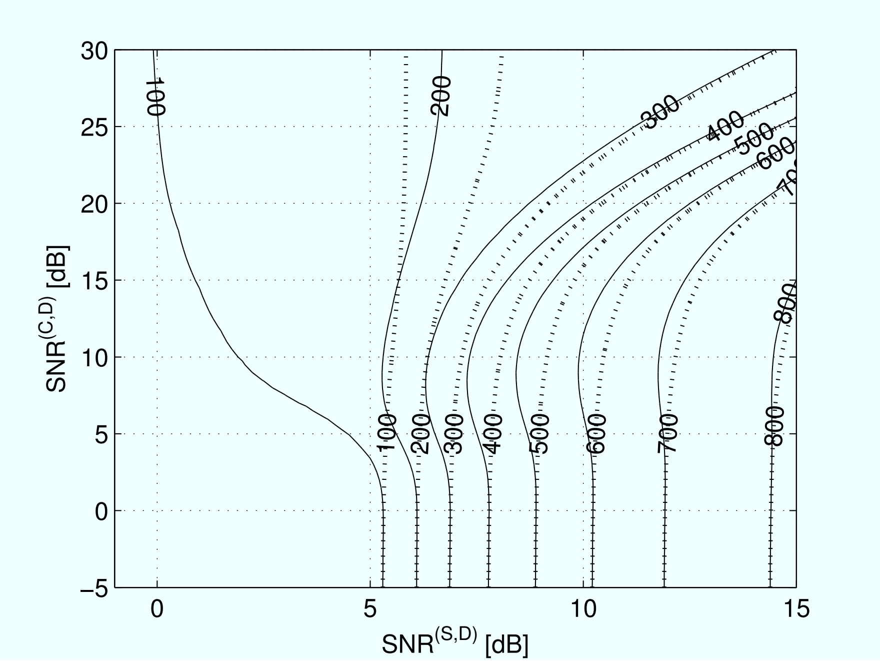


Figure 4.28. *Average throughput in kbps as a function of the average $SNR^{(S,D)}$ and the average $SNR^{(C,D)}$. $SNR^{(S,C)} = 20$ dB for LCCS (solid lines) and LCS (dashed lines).*

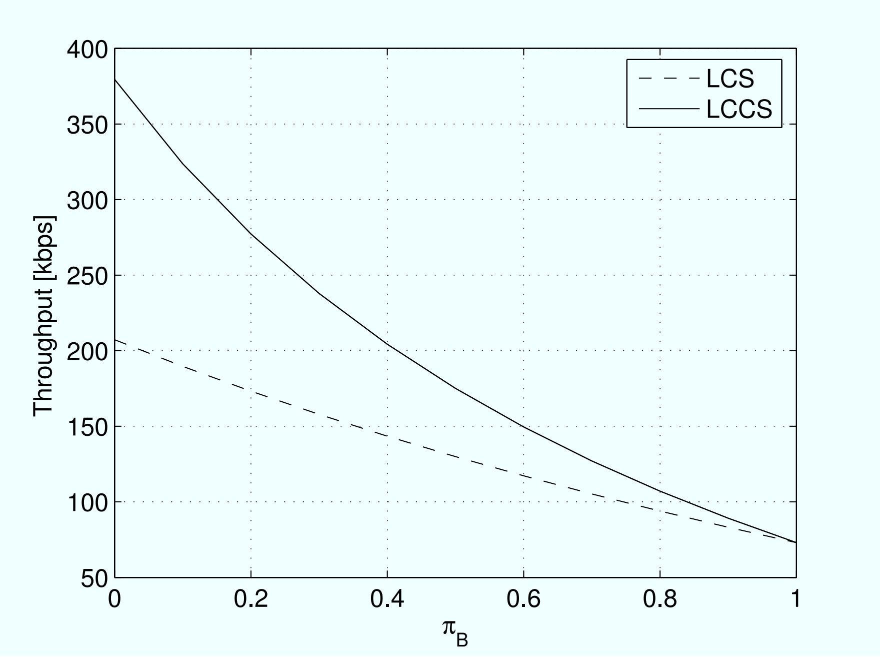


Figure 4.29. *Average throughput as a function of the busy probability π_B. $SNR^{(S,D)} = 6$ dB, $SNR^{(C,D)} = 10$ dB and $SNR^{(S,C)} = 20$ dB.*

obtain a good balance between the spectral efficiency and the interference level. Indeed, the spreading factor could be optimized in order to maximize the overall network throughput, trading off interference level and spectral efficiency. A useful quantity to be optimized could be the information efficiency [32, 33]. However, this optimization should take into account the traffic load as well as channel characteristics and would result in an adaptive technique, which is left for future investigation.

Channel variations are described with an autoregressive moving average (ARMA) model [34,35] and the complex fading coefficient for slot t is $g_{i,j}^{(k,m)}(t) = \rho g_{i,j}^{(k,m)}(t-1) + \sqrt{1-\rho^2}\xi(t)$ where ρ is the correlation coefficient and $\xi(t)$ are independent complex Gaussian variables with zero mean and unit variance. The path-loss has been modeled according to Hata, i.e. $\alpha^{(k,m)} \propto [d^{(k,m)}]^{-4}$, where $d^{(k,m)}$ is the distance between nodes k and m. We assume perfect channel estimation. Detection performance of LASTMUD at each receive node is assessed by the semi-analytical technique of [36].

In order to achieve a deeper understanding of the effects of cooperation on the network performance, we also consider two variants of LCCS:

- **Forced LCCS (FLCCS)**: an idle node C that correctly decodes an RP/GP exchange and matches SNR requirements for cooperation is forced to maintain the idle state until D transmits FP, or until C fails to decode a sub-packet.
- **Idealized LCCS (ILCCS)**: K_C cooperative nodes per communication are chosen among all the candidates, i.e., idle nodes that correctly receive the data packet. The K_C candidates with the highest SNR from the destination are chosen. Simulations are performed with $K_C = 1$. Note that a real implementation of the selection of cooperative nodes would require a high coordination overhead and may not be a viable solution from a protocol perspective. Nevertheless, ILCCS is provided as a guideline for protocol design and for comparison with the uncoordinated version.

Fig. 4.30 shows the overall network throughput, as a function of λ, where the throughput is calculated over the successfully received and acknowledged data packets. We first observe that in a fully connected IEEE 802.11 network where only one node is allowed to transmit at a given time, the maximum network throughput coincides with the maximum node throughput, which in our simulation scenario is 937.5 kbps for a node with two antennas. In our system instead, nodes may transmit simultaneously, achieving a higher network throughput for both LCS and LCCS, despite signaling and redundancy overhead. Moreover, both LCCS and ILCCS outperform LCS, meaning that the throughput gain due to cooperation is larger than the throughput loss due to deferring a node's own communications. Still, a balance between cooperation and non-cooperation is needed, since even FLCCS, that forces cooperation, has a lower throughput than LCCS and ILCCS. Lastly, LCCS and ILCCS perform similarly in this environment, although a totally uncoordinated behavior may result in too many cooperative nodes in more dense networks. This may be counteracted, e.g., with the identification of groups of cooperative nodes or with a probabilistic selection of cooperative nodes.

Fig. 4.31 shows the average probability that a destination fails to decode the data packet after M_{fec} FEC phases. All the cooperative protocols achieve a lower failure rate than LCS,

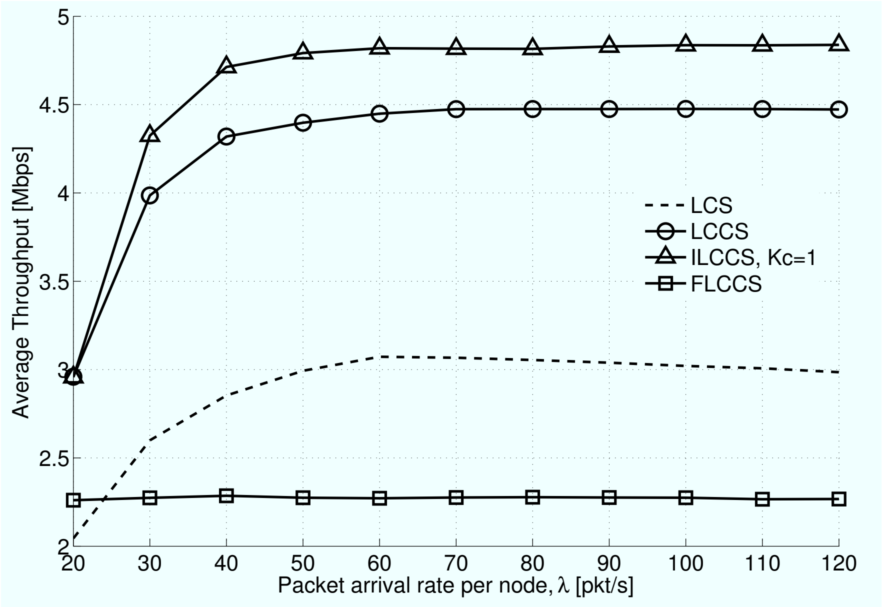


Figure 4.30. *verage network throughput as a function of the per node arrival rate λ for the simulation parameters of Table 4.4.*

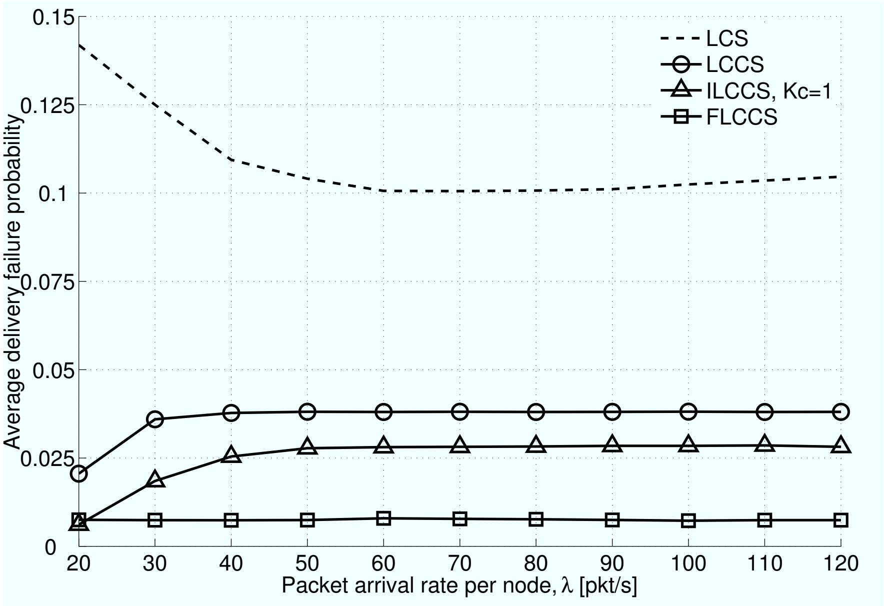


Figure 4.31. *Average transmission error rate as a function of the per node arrival rate λ for the simulation parameters of Table 4.4.*

since cooperative transmissions provide higher coding and diversity gains. On the other hand, when more nodes are cooperating, the interference level is increased and we observe that ILCCS obtains a lower failure probability than LCCS because it selects the best cooperative node among all the candidates and enhances coding and diversity gains while limiting the interference. Indeed, the limitation of the interference allows FLCCS to achieve the

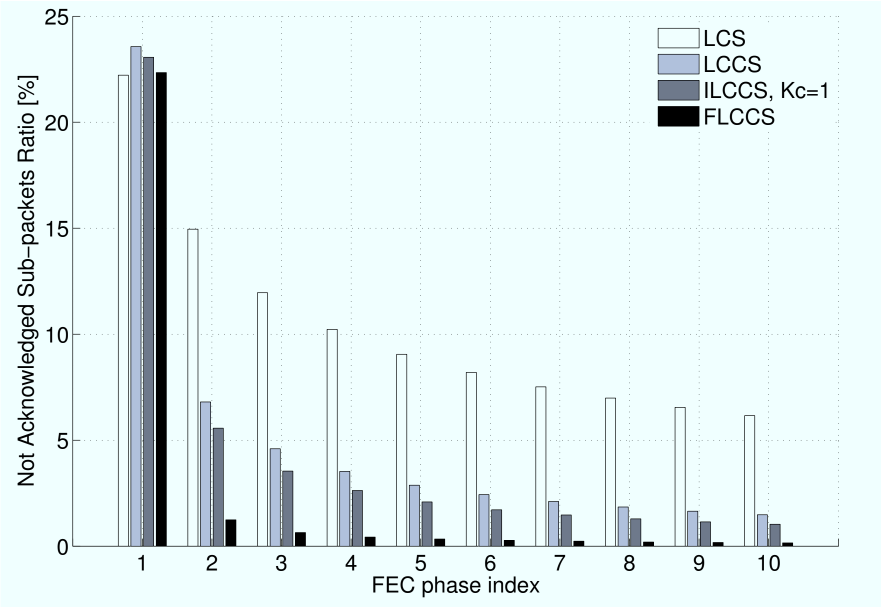


Figure 4.32. *Average not acknowledged sub-packets ratio after the ith FEC phases for the simulation parameters of Table 4.4, $\lambda = 120$ pkt/s.*

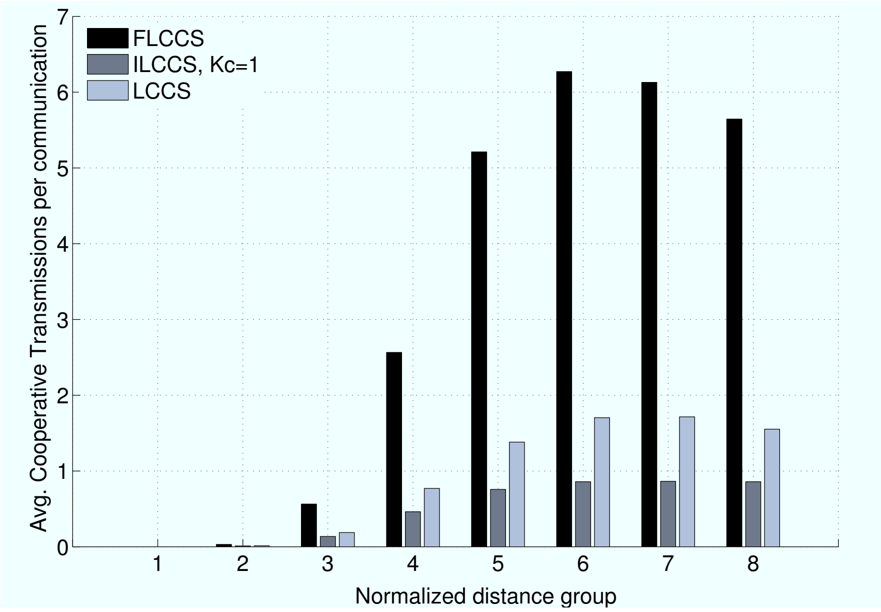


Figure 4.33. *Cooperative/non-cooperative retransmission ratio as a function of the distance group. The simulation parameters are shown in Table 4.4, $\lambda = 120$ pkt/s.*

lowest failure rate among all the presented schemes. In fact, nodes that correctly decode a RP/GP exchange are forced to keep the idle state in order to decode the ongoing data transmission, and the number of simultaneous communications is lowered, so that the receiver is not overloaded.

Simulation results, not reported due to lack of space, show that the performance of the

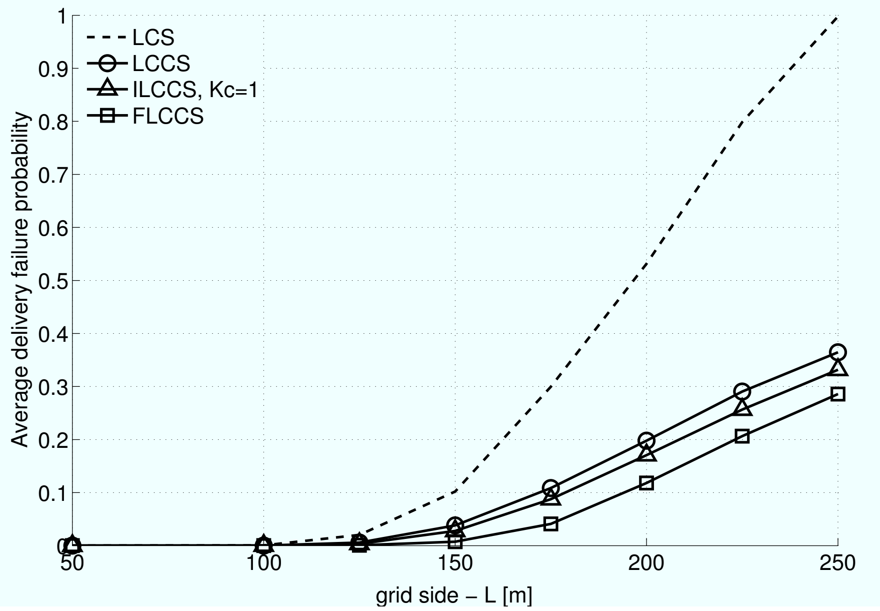


Figure 4.34. *Average delivery failure probability as a function of the grid size. The simulation parameters are shown in Table 4.4,* $\lambda = 120$ *pkt/s.*

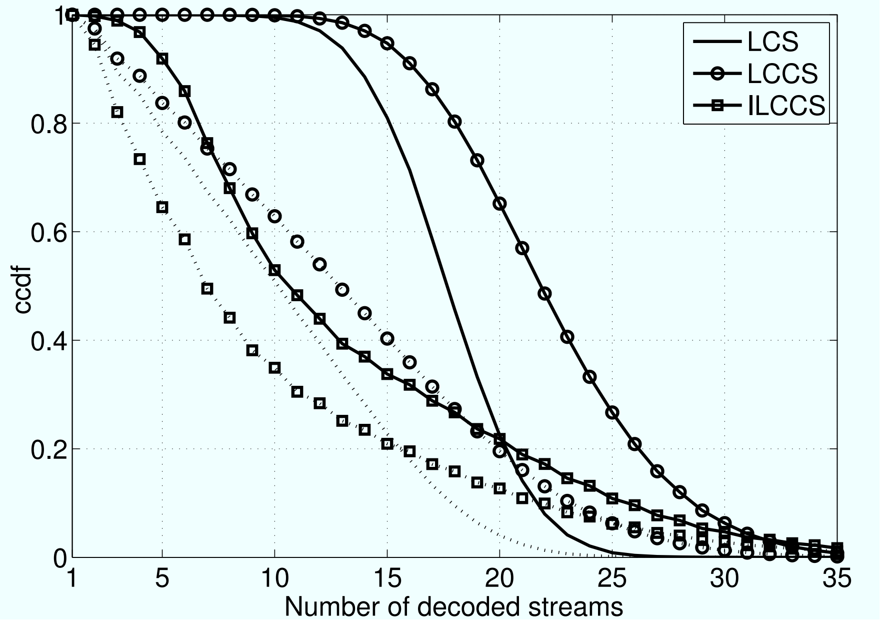


Figure 4.35. *ccdf of the number of decoded streams by a single receiver. HI system is represented with dotted lines. The simulation parameters are shown in Table 4.4,* $\lambda = 120$ *pkt/s.*

network, in terms of throughput and failure rate, is not very sensitive to the value of the backoff parameter W. This is due to the good balance provided by the chosen spreading factor N_S and to the effective HARQ scheme and backoff policy. However, a more aggressive choice of N_S may result in an increased dependence of the achieved performance on W. In order to provide a deeper insight into the evolution of the HARQ process, we report in

Fig. 4.32 the average percentage of blocks needed for correct packet decoding. LCS shows a slowly exponential decrease of the percentage of missing blocks as an effect of channel correlation, because of both fading and interference. Cooperative schemes strongly reduce this percentage at the first phase in which cooperation is active (second FEC phase), providing coding and diversity gain. For the next FEC phases, the percentage decreases slowly because of channel correlation. Indeed, we have observed that most packet deliveries are resolved within the second FEC phase when a node is cooperating.

Fig. 4.33 shows the number of cooperative nodes per communication, as a function of the link distance normalized with respect to the minimum distance ($d_{\min}$) among nodes. Links are collected in groups, where the i–th group comprises links characterized by a source–destination distance between $(i-1)\cdot d_{\min}$ and $i \cdot d_{\min}$. We observe that all cooperative protocols have a similar behavior, with increasing cooperation within groups 1 to 6, as links experience a higher decoding failure probability at the first FEC phase and more nodes match the SNR requirements. In groups 7 and 8 the cooperation level decreases as the decoding probability decreases even for cooperative candidates.

The proposed cooperative protocols increases the correct packet delivery on longer links. Fig. 4.34 plots the failure rate, as defined for Fig. 4.31, as a function of the network grid side L. LCCS, ILCCS and FLCCS soften the threshold behavior of LCS. Despite the protection offered by FEC, as the distance between the nodes increases, links quickly become less stable in LCS, while cooperative systems offer an increased reliability and sustain connectivity. Still, note that power consumption increases with the average distance between nodes, due to the higher number of FEC phases. We can conclude that cooperation is an efficient alternative to routing, as routing may lead to increased delays and further signaling overhead.

In order to evaluate the processing complexity of the considered schemes, Fig. 4.35 shows the complementary cumulative distribution functions (ccdf) of the number of signals (one per transmitting antenna) decoded by the receivers. For the sake of clarity, we do not include the plot associated with ILCCS, since we have verified that it behaves similarly to LCCS. We also plot the ccdf relative to the header inspection (HI) protocol, in which the receive node inspects the headers of the incoming slots and decodes the entire slot only when needed. Note that LCCS requires more decoding than LCS. On the other hand, ILCCS reduces the cooperation effort and has a lower number of decoded streams than LCCS.

4.4.4 Appendix 4.A: Outage probability of DF

For the computation of the outage probability of DF we define the events of outage for cooperative and non-cooperative transmission, as well as the event for cooperation of the first transmission by the cooperator as a function of the SNIR's $\gamma_{\mathtt{S,D}}$, $\gamma_{\mathtt{C,D}}$ and $\gamma_{\mathtt{S,C}}$ as

$$\mathcal{C}_{\text{DF}} = \{(a,b,c) : (1 + a/(N_0 + c))(1 + b/(N_0 + c))^N < 2^R; a,b,c \geq 0\}, \tag{4.80}$$

$$\mathcal{N}_{\text{DF}} = \{(a,c) : (1 + a/(N_0 + c))^{(N+1)} < 2^R; a,c \geq 0\}, \tag{4.81}$$

$$\mathcal{E}_{\mathrm{DF}} = \left\{(a,c) : (1 + a/(N_0 + c)) \geq 2^R; a, c \geq 0\right\}, \tag{4.82}$$

Then the corresponding probabilities of the events are given by the following integrals

$$I_{\mathcal{C},\mathrm{DF}} = \int_{\mathcal{C}_{\mathrm{DF}}} f_{P|d}(a|d_{\mathtt{S},\mathtt{D}}) f_{P|d}(b|d_{\mathtt{C},\mathtt{D}}) f_{P_{\mathrm{tot}}}(c) \mathrm{d}a\mathrm{d}b\mathrm{d}c$$

$$I_{\mathcal{E},\mathrm{DF}} = \int_{\mathcal{E}_{\mathrm{DF}}} f_{P|d}(a|d_{\mathtt{S},\mathtt{C}}) f_{P_{\mathrm{tot}}}(c) \mathrm{d}a\mathrm{d}c$$

$$I_{\mathcal{N},\mathrm{DF}} = \int_{\mathcal{N}_{\mathrm{DF}}} f_{P|d}(a|d_{\mathtt{S},\mathtt{D}}) f_{P_{\mathrm{tot}}}(c) \mathrm{d}a\mathrm{d}c\,.$$

In order to compute the integrals we derive the maximum value of b that gives outage as a function of a and c from (4.80) as

$$b_{\mathrm{DF}}(a,c) = \left[\left(\frac{2^R}{(1 + a/(N_0 + c))}\right)^{1/N} - 1\right](N_0 + c) \tag{4.83}$$

and the maximum value of a, as a function of c, that provides outage when $b = 0$ from (4.80) is

$$A(c) = \left(2^R - 1\right)(N_0 + c). \tag{4.84}$$

Then we can rewrite the probability of event $\mathcal{C}_{\mathrm{DF}}$ as

$$I_{\mathcal{C},\mathrm{DF}} = \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) \int_{a=0}^{A(c)} \left(1 - e^{-\frac{b_{\mathrm{DF}}(a,c)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}}\right) e^{-\frac{a}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \mathrm{d}a\mathrm{d}c\,. \tag{4.85}$$

We now rearrange terms in order to solve the integral over c. Let us define $x = a/(N_0 + c)$ and $a = (N_0 + c)x$ and $\mathrm{d}a = (N_0 + c)\mathrm{d}x$ we can define $b'_{\mathrm{DF}}(x)$ from (4.41) and A' from (4.42) to obtain

$$I_{\mathcal{C},\mathrm{DF}} = \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) \int_{x=0}^{A'} \left(1 - e^{-\frac{b'_{\mathrm{DF}}(x)(c+N_0)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}}\right) e^{-\frac{x(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} (N_0 + c)\mathrm{d}x\mathrm{d}c \tag{4.86}$$

$$\begin{aligned} I_{\mathcal{C},\mathrm{DF}} =& \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) e^{-\frac{x(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} (N_0 + c)\mathrm{d}x\mathrm{d}c \\ & - \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) e^{-\frac{b'_{\mathrm{DF}}(x)(c+N_0)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{x(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} (N_0 + c)\mathrm{d}x\mathrm{d}c \end{aligned} \tag{4.87}$$

Since the characteristic function of P_{tot} is

$$\psi_{P_{\mathrm{tot}}}(\omega) = \left(1 - j\frac{\sigma^2}{\mu}\omega\right)^{-\frac{n_{\mathtt{I}}\mu^2}{\sigma^2}}, \tag{4.88}$$

by the definition of the characteristic function we have

$$\int_0^{\infty} e^{\alpha c} f_{P_{\mathrm{tot}}}(c) \mathrm{d}c = \psi_{P_{\mathrm{tot}}}(-j\alpha)\,. \tag{4.89}$$

By defining the function

$$\begin{aligned}\psi'(\omega) &= \int_{c=0}^{\infty} e^{j\omega c} c f_{P_{\mathrm{tot}}}(c)\mathrm{d}c \\ &= \mu n_{\mathtt{I}} \left(1 - j\frac{\sigma^2}{\mu}\omega\right)^{-\left(\frac{n_{\mathtt{I}}\mu^2}{\sigma^2}+1\right)} .\end{aligned} \tag{4.90}$$

we obtain

$$\begin{aligned}I_{\mathcal{C},\mathrm{DF}} = & \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c)\left[1 - e^{-\frac{A'(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}}\right]\mathrm{d}c \\ & - \frac{N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(x)N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{xN_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left[j\left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x \\ & - \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(x)N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{xN_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi'\left[j\left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x\end{aligned} \tag{4.91}$$

Then we have

$$\begin{aligned}I_{\mathcal{C},\mathrm{DF}} = & 1 - e^{-\frac{A'N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left[j\frac{A'}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right] \\ & - \frac{N_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(x)N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{xN_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi_{P_{\mathrm{tot}}}\left[j\left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x \\ & - \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{x=0}^{A'} e^{-\frac{b'_{\mathrm{DF}}(x)N_0}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}} e^{-\frac{xN_0}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} \psi'\left[j\left(\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}\right)\right]\mathrm{d}x\end{aligned} \tag{4.92}$$

where

$$\frac{b'_{\mathrm{DF}}(x)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} = \frac{2^{R/N}(1+x)^{-1/N}}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} - \frac{1}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}} + \frac{x}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} . \tag{4.93}$$

Lastly, we can rewrite the outage probability as in (4.31), where

$$I_{\mathcal{E},\mathrm{DF}} = \int_{c=0}^{\infty} e^{-\frac{(2^R-1)(N_0+c)}{P_r d_{\mathtt{S},\mathtt{C}}^{-\kappa}}} f_{P_{\mathrm{tot}}}(c)\mathrm{d}c \tag{4.94}$$

$$I_{\mathcal{N},\mathrm{DF}} = 1 - \int_{c=0}^{\infty} e^{-\frac{(2^{R/(N+1)}-1)(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} f_{P_{\mathrm{tot}}}(c)\mathrm{d}c\,. \tag{4.95}$$

and from (4.89) we obtain (4.38) and (4.40).

For O-DF, the condition $\gamma_{\mathtt{S},\mathtt{D}} \leq \gamma_{\mathtt{C},\mathtt{D}}$ can be rewritten as $P(d_{\mathtt{C},\mathtt{D}}) \geq P(d_{\mathtt{S},\mathtt{D}})$ since the denominator of both SNIR's is the same. Therefore

$$I_{\mathcal{S}} = \mathrm{P}[P(d_{\mathtt{C},\mathtt{D}}) \geq P(d_{\mathtt{S},\mathtt{D}})]. \tag{4.96}$$

From (4.20) the event of cooperation can be described as

$$\mathcal{S} = \{(a,b) : b d_{\mathtt{C},\mathtt{D}}^{-\kappa} \geq a d_{\mathtt{S},\mathtt{D}}^{-\kappa}, a > 0, b > 0\} \tag{4.97}$$

and we obtain

$$I_{\mathcal{S}} = \int_{\mathcal{S}} f_{P|d}(b|d) f_{P|d}(a|d) \mathrm{d}a \mathrm{d}b \,. \tag{4.98}$$

By inserting (4.21) into (4.98) we obtain (4.44).

By defining the set

$$\mathcal{C}_{\mathrm{O-DF}} = \{(a,b,c) : (1 + a/(N_0 + c))(1 + b/(N_0 + c))^N < 2^R; a, b, c \geq 0 | b > a\} \,, \tag{4.99}$$

and defining the maximum value of a, as a function of c, that provides outage when $a = b$ as

$$A_{\mathrm{O}}(c) = \left(2^{R/(N+1)} - 1\right)(N_0 + c) \,, \tag{4.100}$$

we have

$$I_{\mathcal{C},\mathrm{O-DF}} = \frac{1}{I_{\mathcal{S}}} \frac{1}{P_r d_{\mathtt{S,D}}^{-\kappa}} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) \int_{a=0}^{A_{\mathrm{O}}(c)} \left(e^{-\frac{a}{P_r d_{\mathtt{C,D}}^{-\kappa}}} - e^{-\frac{b_{\mathrm{DF}}(a,c)}{P_r d_{\mathtt{C,D}}^{-\kappa}}} \right) e^{-\frac{a}{P_r d_{\mathtt{S,D}}^{-\kappa}}} \mathrm{d}a \mathrm{d}c \,. \tag{4.101}$$

Following similar derivations as for $I_{\mathcal{C},\mathrm{DF}}$, we obtain (4.46).

4.4.5 Appendix 4.B: Outage probability of MISO

For the computation of the outage probability of MISO, the set of triplets $(\gamma_{\mathtt{S,D}}, \gamma_{\mathtt{C,D}}, \gamma_{\mathtt{S,C}})$ for which we have outage in case of cooperation is

$$\begin{aligned} \mathcal{C}_{\mathrm{MISO}} = \{(a,b,c) : [(1 + a/(N_0 + c))(1 + b/(N_0 + c + a))^N \cdot \\ (1 + a/(N_0 + b + c))^N] < 2^R; a, b, c \geq 0\} \,, \end{aligned} \tag{4.102}$$

whose probability is given by the integral

$$I_{\mathcal{C}_{\mathrm{MISO}}} = \int_{\mathcal{C}_{\mathrm{MISO}}} f_{P|d}(a|d_{\mathtt{S,D}}) f_{P|d}(b|d_{\mathtt{C,D}}) f_{P_{\mathrm{tot}}}(c) \mathrm{d}a \mathrm{d}b \mathrm{d}c \,. \tag{4.103}$$

From the definition of the set $\mathcal{C}_{\mathrm{MISO}}$ we have the following bound on b

$$b^2 + b[(1 - \xi(a,c))(N_0 + c + a)] + [a - \xi(a,c)(N_0 + c)](N_0 + c + a) < 0 \,. \tag{4.104}$$

Let us define

$$\xi(a,c) = 2^{R/N}\left(1 + \frac{a}{N_0 + c}\right)^{-1/N} - 1 \,, \tag{4.105}$$

$$b_{\mathrm{MISO}}(a,c) = \frac{1}{2}(N_0 + c + a)\left[-(1 - \xi(a,c)) + \sqrt{(1 - \xi(a,c))^2 - \frac{4[a - \xi(a,c)(N_0 + c)]}{(N_0 + c + a)}}\right]^+ \,. \tag{4.106}$$

From the definition of $b_{\mathrm{MISO}}(a,c)$ in (4.106), we obtain

$$\begin{aligned} I_{\mathcal{B}_{\mathrm{MISO}}} = & \frac{1}{P_r d_{\mathtt{S,D}}^{-\kappa}} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) \\ & \cdot \int_{a=0}^{\infty} \left(1 - e^{-\frac{b_{\mathrm{MISO}}(a,c)}{P_r d_{\mathtt{C,D}}^{-\kappa}}}\right) e^{-\frac{a}{P_r d_{\mathtt{S,D}}^{-\kappa}}} \mathrm{d}a \mathrm{d}c \,. \end{aligned} \tag{4.107}$$

Note that for $\frac{a}{N_0+c} \to \infty$ the square root in (4.106) becomes imaginary, i.e., there is no value of b for which outage occurs. However, determining the value of a such that the argument of the square root is zero yields a transcendent equation. Still, we observe that a necessary condition for outage is

$$\xi(a,c) > \frac{a}{N_0+c} \Longrightarrow a < \left(2^{\frac{R}{N}} - 1\right)(N_0+c) = A(c)\,.$$

Hence, the inner integral of (4.107) can be computed over a limited interval and by defining $x = a/(N_0+c)$ and $\xi'(x)$ and $b'_{\mathrm{MISO}}(x)$ by (4.54) and (4.55) we have

$$I_{\mathcal{C},\mathrm{MISO}} = \frac{1}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}} \int_{c=0}^{\infty} f_{P_{\mathrm{tot}}}(c) \int_{b=0}^{A'} \left(1 - e^{-\frac{b'_{\mathrm{MISO}}(x)(c+N_0)}{P_r d_{\mathtt{C},\mathtt{D}}^{-\kappa}}}\right) e^{-\frac{x(N_0+c)}{P_r d_{\mathtt{S},\mathtt{D}}^{-\kappa}}} (N_0+c)\mathrm{d}x\mathrm{d}c. \quad (4.108)$$

Hence, following the same computations as for the DF case, we have (4.56).

References

[1] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.

[2] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[3] L. Dai and K. B. Letaief, "Cross-layer design for combining cooperative diversity with truncated ARQ in ad-hoc wireless networks," in *Proc. Globecom 2005*, St. Louis, MO, Nov. 2005.

[4] M. Haenggi, "On routing in random Rayleigh fading networks," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1533–1562, Jul. 2005.

[5] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *ACM SIGCOMM Computer Commun.*, vol. 34, no. 1, pp. 69–74, Jan. 2004.

[6] K. Navaie and H. Yanikomeroglu, "Induced cooperative multi-user diversity relaying for multi-hop cellular networks," in *Proc. Vehic. Tech. Conf. (VTC) Spring*, 2006.

[7] P. Larsson and N. Johansson, "Multiuser diversity forwarding in multihop packet radio networks," in *Proc. Wireless Commun. Networks Conf.*, 2005.

[8] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comp.*, vol. 2, pp. 337–348, Oct. 2003.

[9] R. Knopp and P. A. Humblet, "Information capacity and power control in single cell multiuser communications," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Jun. 1995, pp. 331–335.

[10] A. Gyasi-Agyei, "Multiuser diversity based opportunistic scheduling for wireless data networks," *IEEE Commun. Letters*, vol. 9, no. 7, pp. 670–672, Jul. 2005.

[11] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, pp. 1939–1948, Nov. 2003.

[12] P. Mitran, H. Ochiai, and V. Tarokh, "Space-time diversity enhancements using collaborative communications," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 2041–2057, Jun. 2005.

[13] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456–1467, Jul. 1999.

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

[15] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 6, pp. 1089–1098, Aug. 2004.

[16] X. E. Li, M. Chen, and W. Liu, “Application of STBC-encoded cooperative transmissions in wireless sensor networks,” *IEEE Signal Processing Lett.*, vol. 12, no. 2, pp. 134–137, Feb. 2005.

[17] T. Tabet, S. Dusad, and R. Knopp, “Achievable diversity-multiplexing-delay tradeoff in half-duplex ARQ relay channels,” in *Proc. Int. Symp. Information Theory (ISIT)*, Sep. 2005, pp. 1828–1832.

[18] S. Tomasin, M. Levorato, and M. Zorzi, “Analysis of outage probability for cooperative networks with HARQ,” in *Proc. Int. Symp. Information Theory (ISIT) 2007*, Nice, France, Jun. 2007.

[19] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, “An efficient protocol for realizing cooperative diversity in wireless networks,” in *Proc. IEEE ISIT 2001*, Washington, DC, June 2001, p. 294.

[20] G. E. Corazza, G. D. Maio, and F. Vatalaro, “CDMA cellular systems performance with fading, shadowing, and imperfect power control,” *IEEE Trans. Vehic. Tech.*, vol. 47, no. 2, pp. 450–459, May 1998.

[21] A. Papoulis, *The Fourier integral and its application*. New York (NY): McGraw-Hill, 1962.

[22] M. Dohler, *Virtual Antenna Arrays*. University of London: Ph.D. Thesis, 2003.

[23] B. Zhao and M. C. Valenti, “Practical relay networks: a generalization of hybrid-ARQ,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 7–18, Jan. 2005.

[24] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, “A simple cooperative diversity method based on network path selection,” *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 659–671, Mar. 2006.

[25] S. Sfar, R. D. Murch, and K. B. Letaief, “Layered space–time multiuser detection over wireless uplink systems,” *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 653–668, Jul. 2003.

[26] S. Hara and R. Prasad, “Overview of multicarrier CDMA,” *IEEE Trans. Commun.*, vol. 35, no. 12, pp. 126–133, Dec. 1997.

[27] L. Rizzo, “Effective erasure codes for reliable computer communications protocols,” *ACM Computer Commun. Review*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[28] E. Martinian and C.-E. W. Sundberg, “Burst erasure correction codes with low decoding delay,” *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2494–2502, Oct. 2004.

[29] L. Libman and A. Orda, “Optimal packet-level FEC strategies in connections with large delay-bandwidth products,” *IEEE Trans. Wireless Commun.*, 2006, accepted for publication.

[30] N. Prasad and M. Varanasi, “Outage analysis and optimization for multiaccess/V-BLAST architecture over MIMO Rayleigh fading channels,” in *Proc. 41st Annual Allerton Conf. on Commun. Control and Computing*, Monticello, IL, Oct. 2003.

[31] P. Larsson, “Large-scale cooperative relaying network with optimal coherent combining under aggregate relay power constraint,” in *Proc. FTC2003*, Beijing, China, 2003, pp. 166–170.

[32] M. W. Subbarao and B. L. Hughes, “Optimal transmission ranges and code rates for frequency-hop packet radio networks,” *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 670–678, Apr. 2000.

[33] M. Souryal, B. Vojcic, and R. Pickholtz, “Information efficiency of multihop packet radio networks with channel-adaptive routing,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 40–50, Jan. 2005.

[34] X. W. R. Chen and J. S. Liu, “Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering,” *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2079–2094, Sep. 2000.

[35] D. Qingyuan and E. Shwedyk, “Detection of bandlimited signals over frequency selective Rayleigh fading channels,” *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 941–950, Feb/Mar/Apr 1994.

[36] M. Levorato, P. Casari, S. Tomasin, and M. Zorzi, “An approximate approach for layered space-time multiuser detection performance and its application to MIMO ad hoc networks,” in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Istanbul, Turkey, Jun. 2006.

Chapter 5

Access and Power Control

Contents

5.1 Introduction

In the previous chapters, we addressed issues concerning the control and the performance of ad hoc networks with multiple simultaneous access. We based our investigation on the assumption that nodes have someway to coordinate in a distributed fashion, or face the unpredictability of the interference in the network. In this last part of the thesis, we take a different approach, mostly motivated by emerging issues in networking, but that adds an important piece to the big picture.

In the last few years there has been a renewed effort in the investigation of optimization strategies for access and transmission control in wireless networks. This is mostly due to the considerable interest attracted by dynamic spectrum allocation and the so called *cognitive radio* [1,2].

In cognitive radio, the scarce usage of the wireless spectrum is improved by granting channel access to unlicensed (secondary) users. The latter are assumed to implement smart algorithms able to adapt their operations to the environment. In particular, access by secondary users must generate minimal interference to the network formed by the licenced (primary) users. Most prior work on the design of secondary users' operations confines their transmissions to frequency/time spaces left unused by primary users, generally referred to as *white spaces* [3–6].

We believe that this approach heavily constrains the performance achieved by secondary users, whose transmissions may in some case coexist with those of the primary users. This is especially true in wireless networks where nodes may implement physical layer architectures resilient to interference, such as multiuser detection (MUD). In the literature there exist approaches for the coexistence of primary and secondary users in the same time/frequency slot, but they are mostly limited to physical layer considerations [2, 7–10] and based on interference temperature or rate distortion rationales.

In part of the thesis, we investigate in depth the optimization of the control of nodes' activity in a scenario where users are allowed to simultaneously access the channel for packet transmission. In this framework, interference mitigation, rather than collision avoidance, is clearly the correct basic working assumption.

We focus on the mutual interference among simultaneously transmitting sources, and in particular on the interaction among the stochastic processes modeling the various sources in the network. This is, in our opinion, a very important point, that opens up interesting issues and scenarios. We want to underscore its connection with cognitive networks. Previous studies on optimizing secondary users' operations assume a fixed model for primary users. Nevertheless, in real-word networks secondary activity would influence the statistics of the stochastic process of primary users.

Differently from most prior work on optimization [11–13], our framework explicitly includes packet delivery failure and retransmission-based error control into the model. In particular, we consider a wireless network with packet arrivals and buffering, where nodes employ a type I hybrid automatic repeat request (HARQ) error control scheme to improve reliability. In this scenario, not only does transmission by a source have an *instantaneous* effect, increasing the error probability of other simultaneous communications, but it also influences the future evolution of the states of other sources, biasing the statistics of the associated stochastic processes. In order to account for this interesting effect, we define a novel interference measure, which we call *process distortion*.

We address the maximization of a general network performance metric, for instance throughput, delay or energy consumption, through the optimization of the policy returning the set of transmitting users and their transmitted power given the state of the network. Differently from prior optimization frameworks for communication networks using a dynamic programming approach [14, 15], we have to solve an infinite horizon average cost per stage problem [16], as we consider the performance of the network averaged over time. If the Markov process modelling the network is homogeneous (*i.e.*, the network is static or slowly varying), it is proved in [16] that the optimal policy is a stationary policy. This poses some technical constraints on the structure of the Markov chain. In particular, to guarantee the existence of the solution of the optimization problem, the space of the considered policies must contain only unichain policies, *i.e.*, policies whose associated Markov chain has a single recurrent class [17]. We provide an insightful discussion on when this condition is guaranteed to hold and a practical example of construction for the space of the policies.

Our interference measure is based on the average performance loss that a transmission policy associated with a group of users generates to the other users. Note that the average performance depends on the steady-state distribution of the Markov chain of the network, which is in turn determined by the transition probabilities. As the interference by a group of users biases the transition probabilities of the Markov process of the others, a certain performance loss corresponds to a *distortion* generated by the activity of the former to the aggregate process.

We exemplify and illustrate this concept through a scenario with two groups of users with different priorities. We first maximize the performance of the group of users with higher priority by optimizing its policy when the low-priority group remains idle, via an unconstrained infinite-horizon dynamic programming approach. We assume that the transmission of the group with lower priority is constrained by a bound on the process distortion measured on the high-priority group, here defined as the performance loss caused to the first group with respect to the previously computed optimum. We model this problem as a constrained stochastic control problem [18] and propose an algorithm that finds a suboptimal solution.

We also provide a detailed model of a cognitive network where secondary users can superpose their transmissions to those of primary users with a bound on the maximum performance loss incurred by the latter. We apply our framework to this model and show numerical results assessing the considerable gain earned by secondary users.

The rest of the chapter is organized as follows. In Section 5.2, we describe the features of the network. Section 5.3 presents the optimization framework. In Section 5.4 we discuss the construction of the Markov model of the network and the issues arising in the various scenarios. In Section 5.5 we apply our framework to a cognitive networking scenario and show how to overcome the technical issues related to the structure of the resulting Markov chain. Section 5.6 concludes the chapter. The reference papers for this part of the work are [J10ml, C22].

5.2 System Description

We consider a single-hop wireless network with a set of N nodes $\mathcal{N}$. Nodes of the network are divided into two disjoint sets, namely the sources' set $\mathcal{S}$ and the destinations' set $\mathcal{D}$, with cardinality S and D, respectively. $\mathcal{D}$ is further divided into S sets $\mathcal{D}_i$, $i=1,\ldots,S$, where

$\mathcal{D}_i$ has cardinality D_i and contains the destinations of source i. For the sake of simplicity, we assume $\mathcal{D}_k \cap \mathcal{D}_j = \emptyset, \forall k \neq j$.

Each source i, $i = 1, \ldots, S$ has to deliver fixed-size packets to a set of destinations $\mathcal{D}_i$. We assume unicast traffic, so that a given packet of a source i is intended to a single destination in $\mathcal{D}_i$. Sources store packets in a finite first-in first-out (FIFO) buffer of size B.

We assume that time is divided into frames, indexed with positive integer numbers t, and that nodes are frame-synchronous. We consider a network where the physical layer architecture is based on MUD. Due to the inherent resilience of MUD receivers to incoming interference, unlike in traditional carrier sense based access systems we allow the simultaneous activation of more than one link in the same area of the network. Therefore, sources do not perform channel sensing before channel access.

Each frame consists of three phases, namely handshake, data and acknowledment. In the handshake phase, nodes negotiate the transmission of the data packets to be transmitted in the data phase, through short control packets. In the data phase, all the packets whose handshakes were successful are transmitted simultaneously. In the acknowledgment phase, the destinations of the packets transmitted send a short control packet reporting the correct/erroneous decoding of the packet.

We do not provide a detailed design of control packets exchange in the network, that is left for future investigation of specific implementations and distributed frameworks. However, in our framework control messages do not prevent neighbors' transmission, as in collision avoidance systems, but are intended to probe the destinations for availability and to negotiate access to the frame.

Sources employ a HARQ error control scheme to improve reliability. In particular, we consider a type I Hybrid ARQ scheme, that relies on both packet encoding and retransmission to increase delivery probability. In particular, the codeword resulting from packet encoding is transmitted to the destination. The packet is removed from the queue if a positive acknowledgment is received by the source. Otherwise, the packet remains in the queue and a further transmission is performed by the source in one of the next frames. The packet is removed from the queue after F failed transmissions. The span $[t_1, t_2]$, where t_1 and t_2 are the frames in which the packet becomes the oldest in the buffer and is removed from the queue, respectively, is referred to as the service of the packet. Due to the access policy and the retransmission scheme, the service of a single packet may last several frames. However, we will show that under reasonable assumptions the service of any packet of any source involves a finite number of frames.

An original aspect of our work is the accurate modeling of the performance of the transmitter/receiver architecture through the relation between the power of the incoming useful and interfering signals received at a given destination and the error probability of the wanted packet.

As discussed in detail in the next section, sources are divided into two groups with different priorities. In particular, transmissions by sources belonging to the lower priority group are constrained by an upper bound on the performance loss they generate to the higher priority group, which is assumed to use the policy that maximizes its own performance in the absence of interference due to low-priority users. We assume that there is no explicit interaction between the two groups in terms of network operations, *i.e.*, they do not, for instance, cooperate to forward packets. However, the two groups indirectly interact due to the interference generated by each transmission at all receivers.

5.3 Network Control and Optimization

In this section, we present and discuss the proposed control policy and optimization framework. We consider the optimization of the average performance of the network in terms of a metric that can represent, for instance, aggregate throughput, delay or energy consumption. We first consider the group of users with higher priority and optimize their performance when users belonging to the second group are assumed to be idle. We then consider the second group of users, with the goal to optimize their performance under the constraint that they generate a bounded performance loss to the first group with respect to the optimum. To this end, we set up two stochastic control problems based on dynamic programming. As we address average performance optimization, we consider infinite horizon average cost per stage minimization problems. This poses several technical challenges, discussed in Section 5.4. The optimal policy solving this class of problems is shown to be a stationary policy $\widehat{\mu}$ mapping the current state of the network to the control vector [16].

5.3.1 Control Policy and Objectives

We model the stochastic process tracking the state of the network in the various frames with a homogeneous Markov process $\mathbf{\Theta} = \{\Theta(0), \Theta(1), \dots\}$, where $\Theta(t)$ is the state of the network in frame t and takes values in the state space $\mathcal{X}$. We define the probability of a sample-path of the process $\mathbf{\Theta}$ through the probability measure $\mathcal{P}(\mathbf{\Theta}=\boldsymbol{\theta}|\Theta(0)=\theta(0))$, where $\theta(0)\in\mathcal{X}$ is the initial state and $\mathbf{\Theta}\in\mathcal{X}^{\infty}$.

State $\Theta(t)$ is the aggregate of the states of all the sources, *i.e.*, $\Theta(t)=\{\Phi_i(t), i=1,\dots,S\}$, where $\Phi_i(t)\in\mathcal{X}_i$ is the state of i in frame t and $\mathcal{X}_i$ is the space of the states of the individual source i. We denote the stochastic process describing the state of i with $\mathbf{\Phi}_i(t)$. We will discuss in Section 5.4 how to build the spaces $\mathcal{X}_i$ to track the state of the network and carry out the optimization algorithm.

The stationary control policy μ maps the state of the network θ to a control vector $\underline{u} = \{u_1, \dots, u_S\}$ in the control set $\mathcal{U}_\theta$. The control variable u_i controls the access and the transmitted power of source i. We denote with $u_i=0$ an idle node, while a control variable $u_i > 0$ means that i accesses the channel with power $p(u_i)$. The motivation behind the choice of implementing power control in the system is its importance in an interference mitigation system. The effect of power control is of particular relevance in MUD systems, where the perceived SNIR may depend not only on the sum, but also on the individual values of the interfering powers coming from the various active sources. Consider for instance a MUD receiver using a successive interference cancellation (SIC) algorithm. In this case, a powerful interfering signal can be successfully decoded and canceled from the overall received signal, improving the quality of the signals remaining to be decoded. Thus, power control provides significant degrees of freedom to the sources to control the *effective* interference in the network. We observe that a similar rationale can be extended to transmission and encoding rate, at the cost of an increased complexity, due to the necessity to track the transmission of fragments of packets.

Since $\mathbf{\Theta}$ is a homogeneous Markov process, we can characterize its evolution under policy μ through the probability measure

$$\zeta_\mu(\theta,\theta^*)=\mathcal{P}_\mu(\Theta(t)=\theta^*|\Theta(t-1)=\theta), \tag{5.1}$$

on $\mathcal{X}\times\mathcal{X}$, $\forall\theta,\theta^*\in\mathcal{X}$ and t.

We define the transition cost function $\delta:\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}^+$ mapping each transition of the network from a state to another in $\mathcal{X}$ to a positive cost. We have $\delta(\theta,\theta^*)=\omega_{\max}(\theta,\theta^*)-\omega(\theta,\theta^*)$, where $\omega_{\max}(\theta,\theta^*)$ and $\omega(\theta,\theta^*)$ are the maximum achievable and the average instantaneous performance associated with the transition from θ and θ^*, whose definition depends on the considered performance metric. We will provide an example of cost definition in Section 5.5. The average cost incurred by the network in state θ is

$$\tilde{\delta}_\mu(\theta) = \sum_{\theta^*\in\mathcal{X}} \delta(\theta,\theta^*)\zeta_\mu(\theta,\theta^*). \tag{5.2}$$

With this construction the cost is additive in time, and we can write the average cost J_μ conditioned on the policy μ as

$$J_\mu=\lim_{T\rightarrow\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathcal{P}_\mu\left(\Theta(t)=\theta\right)\tilde{\delta}_\mu(\theta)=\sum_{\theta\in\mathcal{X}}\pi_\mu(\theta)\tilde{\delta}_\mu(\theta), \tag{5.3}$$

where π_μ is the steady-state distribution conditioned on the policy μ.[1] Thus, the policy determines the performance of the network by influencing the average cost incurred and the fraction of frames spent in each state. It is easy to show that if a policy $\widehat{\mu}$ minimizes the average cost J_μ, then it maximizes the average performance Ω of the network, defined as

$$\Omega = \lim_{T\rightarrow\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathcal{P}_\mu\left(\Theta(t)=\theta\right)\tilde{\omega}_\mu(\theta), \tag{5.4}$$

where $\tilde{\omega}_\mu(\theta)=\sum_{\theta^*\in\mathcal{X}}\omega(\theta,\theta^*)\zeta_\mu(\theta,\theta^*)$. We refer to a cost minimization framework in order to be consistent with the technical literature addressing dynamic programming.

[1] A necessary condition for optimization is that the Markov chains associated with all the considered policies have a single recurrent class [17]. In this case, it is easy to show that the steady-state distribution exists.

Let us define the first and second groups of sources as $\mathcal{S}', \mathcal{S}'' \subseteq \{1, \ldots, S\}$, respectively, with $\mathcal{S}' \cap \mathcal{S}'' = \emptyset$ and $\mathcal{S}' \cup \mathcal{S}'' = \mathcal{S}$. We also define the transition cost functions δ'_μ and δ''_μ referring to the cost incurred by the first and the second group during a transition. Using these functions, we obtain the average cost of the first and second group, *i.e.*, $\tilde{\delta}'_\mu$ and $\tilde{\delta}''_\mu$, as previously shown. We also obtain the average performance J'_μ and J''_μ according to Eq. (5.3). We decompose the overall policy μ into two policies $\mu' \in \mathcal{U}' = \{u_i, i \in \mathcal{S}'\}$ and $\mu'' \in \mathcal{U}'' = \{u_i, i \in \mathcal{S}''\}$ mapping the overall space $\mathcal{X}$ to the control vectors of the first and the second group, respectively. In particular, we have $\mu(\theta) = \underline{u} = \{\underline{u}', \underline{u}''\} = \{\mu'(\theta), \mu''(\theta)\}$. Our first goal is to find the optimal policy $\widehat{\mu}'$ for the first group, defined as

$$\widehat{\mu}' = \arg \min_{\mu' \in \mathcal{U}'} J'_{\{\mu', \underline{0}''\}}, \tag{5.5}$$

where $\underline{0}''$ is a vector of $|\mathcal{S}''|$ zeros. Thus, $J'_{\{\widehat{\mu}', \underline{0}''\}}$ represents the minimum average cost achieved by sources in $\mathcal{S}'$ when sources in $\mathcal{S}''$ keep idle.

We now define the process distortion $\Delta_{\mu'}(\mu''_1, \mu''_2)$ as the difference between the average costs incurred by the first group, that adopts policy μ', when the second group uses policies μ''_1 and μ''_2.[2] Thus, we have

$$\Delta_{\mu'}(\mu''_1, \mu''_2) = J'_{\{\mu', \mu''_1\}} - J'_{\{\mu', \mu''_2\}}. \tag{5.6}$$

As we will explain in detail later when discussing the optimization of the second group of sources, this rather simple measure based on a performance loss accounts for the average instantaneous cost accumulated when the network is in a certain state, but also directly involves the statistics of the process describing the first group of sources, and in particular the average fraction of time the network spends in that state. As stated in Section 5.2, our goal is to obtain the policy maximizing the performance of sources in $\mathcal{S}''$ with a bound on the maximum performance loss incurred by sources in $\mathcal{S}'$ with respect to their optimal performance computed without interference from $\mathcal{S}''$.

To this end, we set the following constrained stochastic control problem [18]

$$\begin{aligned} \widehat{\mu}'' = \arg \min_{\mu'' \in \mathcal{U}''} & J''_{\{\widehat{\mu}', \mu''\}} \\ & \text{s.t. } \Delta_{\widehat{\mu}'}(\underline{0}'', \mu'') \leq \epsilon, \end{aligned} \tag{5.7}$$

where $\epsilon > 0$ is the maximum average performance loss that the activity of group $\mathcal{S}''$ is allowed to generate to $\mathcal{S}'$.

Note that in this case, we have $\Delta_{\widehat{\mu}'}(\underline{0}'', \mu'') \geq 0$ for any μ''. In fact, $J'_{\{\widehat{\mu}', \underline{0}''\}}$ is a global maximum, *i.e.*, the best possible situation for the high-priority users happens when low-priority users are silent.

[2] Note that the distortion $\Delta_{\mu'}(\mu''_1, \mu''_2)$ also depends on the transition matrix of the process associated to $\mathcal{S}'$ and, generally speaking, to the functions describing receivers performance and interference. We omit a complete description of the dependencies of the measure Δ in order to avoid cumbersome notation.

We want to remark that this constraint confines the interference generated by sources in $\mathcal{S}''$ within a certain region. However, differently from previous approaches addressing interference mitigation, we have that this bound does not represent an interference temperature or rate distortion approach in the traditional sense (see for instance [7,10]), but rather consists in a bound on the *interference* of transmissions of $\mathcal{S}''$ to the stochastic process describing the users in $\mathcal{S}'$. In fact, transmissions by sources of the second group *distort* the Markov process of the network, since the generated interference influences the SNIR, and, thus, the correct decoding probability perceived by the receivers of the first group. The interference modifies the transition probabilities of the overall process, and hence the steady-state probabilities $\pi_{\mu}(\theta)$ and average costs $\tilde{\delta}_{\mu}(\theta)$. This results in an increased average cost.

5.3.2 Optimization

Here we describe the optimization algorithm used to find the policies $\widehat{\mu}'$ and $\widehat{\mu}''$. As said before, we first optimize the access and transmitted power of $\mathcal{S}'$ when $\mathcal{S}''$ keeps idle. To this end, we set up an unconstrained infinite horizon stochastic control problem.

It is possible to show that if the scalar λ and the set of differential costs $\{h(\theta), \theta \in \mathcal{X}\}$ solve Bellman's equation

$$\lambda + h'(\theta) = \min_{\mu'} \left[\tilde{\delta}'_{\mu}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\mu',\underline{0}''\}}(\theta,\theta^*) h'(\theta^*) \right] \tag{5.8}$$

for all $\theta \in \mathcal{X}$, with the normalizing condition $h'(\theta_r) = 0$, where θ_r is an arbitrarily chosen state in $\mathcal{X}$, then λ is the optimal average cost per stage and the corresponding policy $\widehat{\mu}'$ is the optimal policy for the first group.

We want to briefly clarify the structure of the problem. As we are addressing an infinite horizon stochastic control problem, and we do not assume an initial and a termination state, the optimization is carried out over infinite-time realizations of the Markov process. Thus, optimization works in cycles where the process starts and returns to a reference state. This mandates that the Markov process to be considered for optimization has a particular structure, that ensures the existence of cycles of finite duration. In the previous equation, differential costs $h'(\theta)$ represent the difference between the average cost incurred by the process while returning to the reference state from θ and the average optimal cost.

It is proved in [16] that if every policy within the class of stationary policies is unichain, *i.e.*, its associated Markov chain has a single recurrent class (see [17] for a precise definition), corresponding to the requirement on the existence of cycles, then Eq. (5.8) has a solution, and the optimal policy and average per stage cost are independent of the initial state $\Theta(0)$. In this case, we have $J'_{\{\widehat{\mu}',\underline{0}''\}} = \lambda$.

To find the optimal policy $\widehat{\mu}'$ we refer to the *policy iteration* algorithm [16]. This algorithm, given a stationary policy μ', produces an improved policy by means of cost minimization. Thus, from an initial policy μ'_0 we obtain a sequence $\mu'_1, \mu'_2, \mu'_3, \ldots$ of policies. If all the generated policies are unichain the algorithm produces improved policies, *i.e.*, $J'_{\{\mu'_{k+1},\underline{0}''\}} <$

$J'_{\{\mu'_k,\underline{0}''\}}$, until it reaches in a finite number of iterations the optimal policy $\mu'_K = \widehat{\mu}'$, $K < \infty$, with $\mu'_k = \mu'_K = \widehat{\mu}'$, $\forall k \geq K$.

Given a policy μ'_k, the algorithm first performs the *policy evaluation* step, computing λ'_k and $\{h'_k(\theta)\}, \theta \in \mathcal{X}$ satisfying

$$\lambda'_k + h'_k(\theta) = \tilde{\delta}'_{\{\mu'_k,\underline{0}''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\mu'_k,\underline{0}''\}}(\theta,\theta^*) h'_k(\theta^*), \tag{5.9}$$

$\forall \theta \in \mathcal{X}$, with the normalizing condition $h'_k(\theta_r) = 0$. This system can be solved directly, or using a modified version of the value iteration method [16, Prop. 4.4.3]. Then the algorithm produces an improved policy μ'_{k+1} such that

$$\begin{aligned}\tilde{\delta}'_{\{\mu'_{k+1},\underline{0}''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\mu'_{k+1},\underline{0}''\}}(\theta,\theta^*) h'_k(\theta^*) &= \\ = \min_{\underline{u}' \in \mathcal{U}'} \left[\tilde{\delta}'_{\{\underline{u}',\underline{0}''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\underline{u}',\underline{0}''\}}(\theta,\theta^*) h'_k(\theta^*) \right]&, \end{aligned} \tag{5.10}$$

$\forall \theta \in \mathcal{X}$.

The achieved average performance can be computed as

$$\Omega' = \sum_{\theta \in \mathcal{X}} \pi_{\{\widehat{\mu}',\underline{0}''\}}(\theta) \left(\omega_{\max}(\theta) - \tilde{\delta}'_{\{\widehat{\mu}',\underline{0}''\}}(\theta) \right), \tag{5.11}$$

where $\omega'_{\max}(\theta)$ is the maximum achievable performance in state θ when considering only the first group.

We observe that, since the sources of the second group keep idle, the sources of the first group can work ignoring their presence in the network. In fact, they do not create interference to the intended receivers of the first group, and, therefore, do not influence the processes $\mathbf{\Phi}_i$, $i \in \mathcal{S}'$. Hence, we can apply the algorithm to the reduced space $\mathcal{X}' = \prod_{i \in \mathcal{S}'} \mathcal{X}_i$ with transition probabilities redefined accordingly. Thus, we find a policy $\widehat{\mu}'$ defined on the space $\mathcal{X}'$. When considering the whole network, we redefine the policy on the whole state space according to $\widehat{\mu}'(\theta',\theta'') = \widehat{\mu}'(\theta')$, for any state $\theta'' \in \mathcal{X}''$.

Once the optimal policy $\widehat{\mu}'$ is found, we have to find the policy μ'' that minimizes the average cost per stage J'' while guaranteeing a bounded performance loss to $\Delta_{\widehat{\mu}'}(\underline{0}'', \mu'')$. To this end, we define the average cost increase $\tilde{\psi}'_{\{\widehat{\mu}',\mu''\}}(\theta)$ as the difference between the average cost of the first group associated with state θ when the policy $\{\widehat{\mu}', \mu''\}$ is used and the optimal average cost per stage of the first group when the second one keeps idle, *i.e.*, $\tilde{\psi}'_{\{\widehat{\mu}',\mu''\}}(\theta) = \tilde{\delta}'_{\{\widehat{\mu}',\mu''\}}(\theta) - \tilde{\delta}'_{\{\widehat{\mu}',\underline{0}''\}}(\theta)$. Thus, $\tilde{\psi}'_{\{\widehat{\mu}',\mu''\}}$ represents the average performance loss of the high-priority group when the low-priority group transmits with policy μ''.

The constrained optimization problem defined in Eq. (5.7) can be restated as

$$\begin{aligned}&\lambda'' + h''(\theta) = \min_{\mu''} \left[\tilde{\delta}''_{\{\widehat{\mu}',\mu''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\widehat{\mu}',\mu''\}}(\theta,\theta^*) h''(\theta^*) \right] \\ &\text{s.t. } \eta'_{\{\widehat{\mu}',\mu''\}} \leq \epsilon, \end{aligned} \tag{5.12}$$

where $\eta'_{\{\widehat{\mu}',\mu''\}}$ is the solution of the system of $|\mathcal{X}|$ equations

$$\eta'_{\{\widehat{\mu}',\mu''\}} + h'(\theta) = \tilde{\psi}'_{\{\widehat{\mu}',\mu''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\widehat{\mu}',\mu''\}}(\theta,\theta^*)h'(\theta^*) \tag{5.13}$$

in $|\mathcal{X}|$ unknowns $\eta'_{\{\widehat{\mu}',\mu''\}}$, $h'^{(\theta)}$, $\theta \neq \theta_r$, due to the normalization $h'(\theta_r) = 0$.

It is possible to observe that the bias the second group activity generates to the transition probabilities has two important effects in Eq. (5.13). First, it potentially increases the average cost incurred by the first group of sources in the various states. In fact, for any reasonable physical layer architecture and environment we have $\tilde{\psi}'_{\{\widehat{\mu}',\mu''\}}(\theta) > 0$, $\forall\theta$. Interference by users with low-priority also modifies the future evolution from each state θ. This leads to increased values for the differential costs, that is, the process collects higher average costs while moving along the cycles.

We find a suboptimal solution with an approach similar to that proposed in [19]. In particular, we modify the policy iteration algorithm described before as follows. After a policy evaluation step analogous to (5.9) in order to update the differential costs, the algorithm generates the policy μ''_{k+1} from μ''_k according to

$$\begin{aligned} &\tilde{\delta}''_{\{\widehat{\mu}',\mu''_{k+1}\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\widehat{\mu}',\mu''_{k+1}\}}(\theta,\theta^*)h''_k(\theta^*) = \\ &= \min_{\underline{u}'' \in \mathcal{W}''(\theta)} \left[\tilde{\delta}''_{\{\widehat{\mu}',\underline{u}''\}}(\theta) + \sum_{\theta^* \in \mathcal{X}} \zeta_{\{\widehat{\mu}',\underline{u}''\}}(\theta,\theta^*)h''_k(\theta^*) \right] \end{aligned} \tag{5.14}$$

where $\mathcal{W}''(\theta) \subseteq \mathcal{U}''$ is the set of policies such that $\eta'_{\{\widehat{\mu}',\mu''\}}$ is less than or equal to ϵ. If all the considered policies are unichain, then the algorithm produces an improved policy at each iteration that does not violate the constraint, *i.e.*, $J''_{\{\widehat{\mu}',\mu''_{k+1}\}} \leq J''_{\{\widehat{\mu}',\mu''_k\}}$ with $\eta'_{\{\widehat{\mu}',\mu''_{k+1}\}} \leq \epsilon$. We could not formally prove the convergence of the algorithm to the optimal policy $\widehat{\mu}''$. However, the good performance obtained by the generated policies empirically shows that it converges to the optimal policy in many cases, or that the achieved local optimum point is not much worse than the optimal performance.

5.4 Discussion and Technical Issues

In the following, we discuss the construction of the Markov model of the network depending on the considered scenario and the technical issues related to the structure of the chain.

The information encoded in the state of each individual source depends on the features of the considered system and the performance metric. The minimum information required to evolve the model is the triple $\Phi_i(t) = \{b_i(t), d_i(t), f_i(t)\}$, where $b_i(t) \in \{0, 1, \ldots, B\}$, $d_i(t) \in \mathcal{D}_i$ and $f_i(t) \in \{0, \ldots, F-1\}$ are the number of packets in the buffer, the destination, and the number of times the packet currently being served by source i in frame t has been transmitted, respectively. Note that if the buffer of i is empty, then $f_i(t)$ and $d_i(t)$ are forced to zero. It can be shown that the space of the states $\mathcal{X}$, built from these individual sources' spaces if the channels are memoriless, routing is not considered and the performance metric does not require additional information, allows the construction of a probability measure such that the Markov property holds. In fact, in this case the next state of a single source i depends on the number of arrivals in its buffer, that does not require any further information. The number of transmissions f_i is needed to remove the packet from the queue when the F-th transmission is performed. The current destination determines the statistics of the channel between i and d_i. It is important to observe that even under these assumptions, the stochastic processes describing the various individual sources are not Markov, as the probability that these processes move to a given state depends on the overall state $\Theta(t)$ and the policy μ. This is due to interference, that makes the performance of each receiver a function also of the vector $\mu(\Theta(t))$.

Note that a channel with memory requires the inclusion of previous channel coefficients associated with the link between i and all the destinations in $\mathcal{D}_i$. Analogously, correlated arrivals would require to track the number of prior arrivals. Finally, the implementation of routing in the network requires to at least keep memory of the final destination of the packet to be forwarded. These extensions, although conceptually straighforward, result in an increased computational complexity.

Also the performance metric potentially influences the amount of information we need to encode in $\Phi_i(t)$ as we need transition costs to be a function only of the pair of states and the adopted policy. For instance, the construction for the individual source space shown before is sufficient to define the transition costs to optimize aggregate throughput. The average aggregate throughput associated with a transition can be written as the average number of packets successfully delivered to their intended destinations. We say average as this number is not entirely deterministic. In fact, the packet is removed from the queue at the F-th transmission even if not delivered to the intended destination. In this case, delivery failure or success failure cannot be distinguished by the pair of states associated to the transition and the policy. Thus, the average number of packets successfully delivered by the source is equal to the correct decoding probability. As an example, we will provide a detailed characterization of a performance metric, state space, transition probabilities and costs for a practical case in the next section.

The main technical challenge concerns the structure of the Markov chain modeling the network. It can be shown that, in any scenario, there exists a subset of the space of the policies containing non-unichain policies. We recall that unichain policies are associated with Markov chains with a single recurrent class, and a certain number of transient classes [17]. Let us briefly recall the definition and the properties of recurrent and transient classes. A class $\mathcal{C}$ is a subset of $\mathcal{X}$ where, for any pair of states $\theta, \theta^* \in \mathcal{C}$, there is a positive probability that the process moves from θ to θ^* and returns to θ in a finite time, *i.e.*, $\exists \tau_1, \tau_2 > 0 :$ $\mathcal{P}_\mu(\Theta(t+\tau_1) = \theta^* | \Theta(t) = \theta) > 0$ and $\mathcal{P}_\mu(\Theta(t+\tau_1+\tau_2) = \theta | \Theta(t+\tau_1) = \theta^*) > 0$. Note that in this case, also the probability that the process returns in θ from θ in a finite time is positive.

If $\sum_{\tau=1}^{+\infty} \mathcal{P}_{\mu}(\Theta(t+\tau)=\theta|\Theta(t)=\theta)$ is equal to one for any pair of states in $\mathcal{C}$, then the class is recurrent. Thus, if a class is recurrent, if $\Theta(t) \in \mathcal{C}$ in a given frame t, then $\Theta(t+\tau) \in \mathcal{C}$ for all $\tau > 0$ [17]. Moreover, it is easy to see that in this case, starting from a state of the class, the process visits every state of the class an infinite number of times in an infinite sample-path [17]. Conversely, if the infinite sum reported above is strictly less than one, then the class is said to be transient, and from any state $\theta \in \mathcal{C}$ the process will eventually leave the class and never come back with probability one.

In a finite state space, the process is eventually absorbed by one of the recurrent classes and only recurrent states contribute to the average per stage cost, as in an infinite sample-path the number of visits to transient states is finite. The computation of the average cost per stage and of the optimal policy is based on the construction of sample-paths starting and ending in a reference state [16]. Thus, the existence of these cyclic paths (or more precisely, the existence of cyclic paths with positive probability) is a necessary condition for the existence and derivation of the optimal policy. If the chain decomposes into multiple recurrent classes, Bellman's system of equations may not admit a solution.

As stated before, $\mathcal{U}$ contains policies whose associated Markov chains have more than one recurrent class. Consider, for instance, a degenerate policy where source i does not transmit when its individual state is $\{b_i, f_i=f, d_i\}$, $\forall b_i>0$, $d_i \in \mathcal{D}_i$. If i has a strictly positive arrival rate and error probabilities, then it can be easily shown that the overall process, once it visits a state $\{\Theta : \phi_i = \{b_i, f_i < f, d_i^a\}\}$, will eventually enter the set of states $\mathcal{C}^a = \{\Theta : \phi_i=\{B, f, d_i = d^a\}\}$, from which it cannot escape. Note that this is true for any destination $d^a \in \mathcal{D}_i$, and such absorbing sets for distinct destinations are disjoint, Therefore, the process decomposes into several recurrent classes [17].

This is only an extreme example of non-unichain policy contained in $\mathcal{U}$. In order to carry out the optimization, we have first to identify a unichain subset of $\mathcal{X}$ and then show that it contains the optimal policy. While the former step is straightforward in most cases, the difficulty of the latter depends on the considered metric.

If we do not consider cases where the decoding probability of packets sent by a source to a given destination is equal to one, and exclude degenerate traffic regimes (*i.e.*, for which the arrival rate is equal to zero or infinity) and channel models, then if a certain source i transmits infinite times in an infinite realization of the process, it hits infinite times each of the states in $\mathcal{X}_i$ built as shown before. Note that this means that the service of a packet has a finite duration. It is possible to prove that, with the assumption on the decoding probability and the traffic regimes reported before, if any source hits infinite times any of its states and we consider a finite state space, then we have a single recurrent class (and a certain number of transient classes). However, we have to restrict the search space to those policies that have a single recurrent class We will provide an example of reduced search space later in this Chapter.

5.5 Case Study: Cognitive Networking

In this Section, we provide an example of application of the presented optimization framework. In particular, we consider the maximization of the aggregate throughput of a wireless cognitive network. Mitola's first definition of cognitive radio [1] appears to be particularly suited for a scenario that has been attracting considerable attention in recent years, that is, the improvement of the poor spectrum usage of wireless networks by granting channel access to unlicensed (secondary) users. These users are required to be strongly adaptive, in order to be able to exploit transmission chances, in an environment where licensed (primary) users are prioritized and assumed not to be significantly damaged by secondary users' activity.

Our optimization framework and interference measure can represent a step forward in the study of this problem. In fact, although it proposes centralized control, it enables the investigation of an interesting scenario, where the rather complex interactions among users are rigorously modeled. Moreover, our distorsion bound takes into account instantaneous interference, but also the long term bias on the stochastic model of primary users due to secondary user activity. We leave the design of distributed algorithms and protocols based on our proposal for future work.

In the following, we describe the considered network and present numerical results assessing the throughput gains achieved by secondary users when removing the zero-interference constraint that corresponds to approaches where secondary users' access is limited to white spaces only.

5.5.1 Network Model

In the cognitive network scenario previously described, the first and second group of our framework correspond to primary and secondary users, respectively. We consider a single-hop network and assume stationary channels with exponential path loss, *i.e.*, the attenuation of a signal received at a distance ℓ by its source is attenuated by a factor $\ell^{-\beta}$, where β is the path loss exponent. The packet arrival process at each source i is modeled as a Poisson process of intensity ν_i[pkt/frame]. As previously shown, with this assumptions we can build a Markov model of the network by encoding in the state of each source the number of packets in the buffer, the number of retransmissions and the destination of the packet currently being served. Thus, we define $\mathcal{X}_i$ as the space of the vectors $\{b, f, d\}$, with $b \in \{0, \dots, B\}$, $d \in \mathcal{D}_i$ and $f \in \{0, \dots, F-1\}$. The overall state space of the network is $\mathcal{X} = \prod_{i=\{1,\dots,S\}} \mathcal{X}_i$.

We address the maximization of the average aggregate throughput, defined as the average sum of the rates of the successfully received packets per frame. We have

$$\omega(\theta) = \sum_{i:[\mu(\theta)]_i > 0} R\,\mathbf{1}(\Xi_\mu(\theta, d_i)), \tag{5.15}$$

where R is the transmission rate, $[\mu(\theta)]_i = u_i$, $\mathbf{1}(\cdot)$ is the indicator function and $\Xi_\mu(\theta, d_i)$ is the correct decoding event at receiver d_i when the network is in state θ and policy μ is used. The aggregate transition costs $\delta_\mu(\theta, \theta^*)$, $\theta, \theta^* \in \mathcal{X}$, can be written as $\delta_\mu(\theta, \theta^*) = \sum_{i=1,\dots,S} \gamma^i_\theta(\phi_i, \phi^*_i)$, where $\gamma^i_\theta(\phi_i, \phi^*_i)$ is the cost associated with source i's transition from state ϕ_i (with overall state θ) to state ϕ^*_i. $\gamma^i_\theta(\phi_i, \phi^*_i)$ is defined as the difference between the maximum number of packets that can be delivered in θ and the average number of packets actually delivered by source i associated to the transition to the individual state ϕ^*.

It is worth noting that transitions having positive probability when $[\mu(\theta)]_i = 0$, *i.e.*, the source is idle, have individual source cost equal to 0 if the buffer is empty, since the maximum number of packets that can be delivered is equal to zero, and to the rate R if $b_i > 0$. If $[\mu(\theta)]_i = 1$ (and thus b_i is strictly positive), we have individual costs

$$\gamma^i_\theta(\{b_i, d_i, f_i\}, \{b^*_i, d_i, f_i + 1\}) = R, \tag{5.16}$$

for all $b_i \leq b^*_i$,

$$\gamma^i_\theta(\{b_i, d_i, f_i\}, \{b^*_i, d^*_i, 0\}) = 0, \tag{5.17}$$

for all $b_i - 1 \leq b^*_i, 0 < b_i$. Thus we have the individual source cost equal to 0 if the process moves to states corresponding to an empty queue or a new service, as these states can be reached only if the packet is removed from the queue.

Let us denote by $\rho^\mu_i(\theta)$ the correct decoding probability at the intended receiver of i when the network is in state θ and policy μ is used. Note that state and policy univocally correspond to the set of transmitters in the network and their transmitted power, and therefore determine the average success probability at a certain receiver.

If $f_i = F - 1$ then

$$\gamma^i_\theta(\{b_i, d_i, F-1\}, \{b^*_i, d^*_i, 0\}) = R(1 - \rho^\mu_i(\theta)), \tag{5.18}$$

$\forall b_i - 1 \leq b^*_i$. In fact, since $f_i = F-1$, the packet is mandatorily removed from the queue during the transition. Therefore, the average cost associated with these transitions is proportional to the probability that the attempt fails. All other transitions have probability equal to zero, and then their cost does not influence the average costs.

We can decompose the transition probabilities in a way similar to the cost, writing $\zeta_\mu(\theta, \theta^*) = \prod_{i=1,\dots,S} \sigma^i_\theta(\phi, \phi^*)$, where $\sigma^i_\theta(\phi, \phi^*)$ is the probability associated with source i's transition from state ϕ_i (with overall state θ) to state ϕ^*_i. If the buffer of i is empty (and therefore its decision variable is forced to zero) then the transition probabilities are

$$\begin{aligned} \sigma^i_\theta(\{0, d_i, 0\}, \{0, d_i, 0\}) &= \alpha_i(0), \\ \sigma^i_\theta(\{0, d_i, 0\}, \{b^*_i, d^*_i, 0\}) &= \alpha_i(b^*_i)/|\mathcal{D}_i|, \ 0 < b^*_i \end{aligned} \tag{5.19}$$

where $\alpha_i(m)$ is the probability that m packets arrive in the buffer of i during a frame and $1/|\mathcal{D}_i|$ is the probability that d^*_i is selected as destination of the next packet to be served. The other individual transition probabilities are reported in Eqs. (5.20), (5.21), (5.22).

$$[\mu(\theta)]_i = 0 \begin{cases} \sigma^i_\theta(\{b_i, d_i, f_i\}, \{b^*_i, d^*_i, f^*_i\}) & = \alpha_i(b^*_i - b_i) \\ \sigma^i_\theta(\{b_i, d_i, f_i\}, \{B, d_i, f_i\}) & = \sum_{m=B-b_i}^{+\infty} \alpha_i(m) \end{cases} \tag{5.20}$$

$$[\mu(\theta)]_i = 1,\ f_i < F-1 \begin{cases} \sigma^i_\theta(\{b_i, d_i, f_i\}, \{b^*_i, d_i, f_i+1\}) & = (1-\rho^\mu_i(\theta))\alpha_i(b^*_i - b_i) \\ \sigma^i_\theta(\{b_i, d_i, f_i\}, \{B, d_i, f_i+1\}) & = (1-\rho^\mu_i(\theta)) \sum_{m=B-b_i}^{+\infty} \alpha_i(m) \\ \sigma^i_\theta(\{b_i, d_i, f_i\}, \{b^*_i, d^*_i, 0\}) & = \frac{\rho^\mu_i(\theta)}{|\mathcal{D}_i|}\alpha_i(b^*_i - b_i + 1) \\ \sigma^i_\theta(\{b_i, d_i, f_i\}, \{B, d^*_i, 0\}) & = \frac{\rho^\mu_i(\theta)}{|\mathcal{D}_i|} \sum_{m=B+1-b_i}^{+\infty} \alpha_i(m) \end{cases} \tag{5.21}$$

$$[\mu(\theta)]_i = 1,\ f_i = F-1 \begin{cases} \sigma^i_\theta(\{b_i, d_i, F-1\}, \{b^*_i, d^*_i, 0\}) & = \frac{1}{|\mathcal{D}_i|}\alpha(b^*_i - b_i + 1) \\ \sigma^i_\theta(\{b_i, d_i, F-1\}, \{B, d^*_i, 0\}) & = \frac{1}{|\mathcal{D}_i|} \sum_{m=B-b_i+1}^{+\infty} \alpha_i(m) \end{cases} \tag{5.22}$$

where $b^*_i < B$ in in Eq. (5.20) and $b_i \leq b^*_i < B$ in the first and third case of Eqs. (5.21) and the second case of Eq. (5.22).

If the node is serving a packet and it does not transmit, *i.e.*, $[\mu(\theta)]_i = 0$ and $b_i > 0$, then the transition probabilities are as in Eq. (5.20), with $b^*_i \geq b_i$. In fact, the number of transmissions f_i is not increased and the packet remains in the buffer with probability one. Therefore, the process can only move to states with a higher or equal number of packets in the buffer.

If the buffer is not empty and the node chooses to transmit, *i.e.*, $[\mu(\theta)]_i = 1$ and $b_i > 0$, and the current attempt is not the last allowed attempt, *i.e.*, $f_i < F - 1$, then we have the transition probabilities shown in Eq. (5.21). Thus, if the transmission fails, f_i is increased by one and the number of packets in the buffer is updated with new arrivals. If the packet is correctly received, it is removed from the buffer. In this case, depending on the number of new arrivals and of packets in the buffer before transmission, the state of source i can either move to the empty buffer state or to a new service. If the node is going to perform the last attempt of a service we have transition probabilities as defined in Eq. (5.22).

In fact, in this case, the packet is removed from the buffer as the maximum number of transmissions is reached. All other individual source transitions have probability equal to zero.

5.5.2 Network Optimization

As stated before, we have to reduce the space of the policies considered during the optimization algorithm to a set of unichain policies, that we call $\mathcal{U}_{\text{unichain}}$, that contains the optimal policy.

Let us consider first the optimization of primary users' policy.[3] We introduce the definition of a persistent policy.

Definition 1. *We say that the policy μ' is persistent if the following holds:*

$$\sum_{i \in \mathcal{S}'_{\text{tx}}(\theta, \mu')} [\mu'(\theta)]_i > 0 \textit{ if } \mathcal{S}'_{\text{tx}}(\theta, \mu') \neq \emptyset, \tag{5.23}$$

[3] In this part we consider only the set of sources $\mathcal{S}'$ and the associated processes, transitions and costs.

where $\mathcal{S}'_{\text{tx}}(\theta,\mu')$ is the set of primary sources transmitting when the network is in state θ and policy μ' is used.

Thus, a policy is persistent if at least one source with a packet in the buffer transmits. We denote with $\mathcal{U}_p$ the set of the persistent policies.

Proposition 2. *In the considered case, if μ' is a persistent policy, then μ' is unichain.*[4]

We will prove this proposition in Sec. 5.8.

Persistent policies also ensure that every individual source process $\mathbf{\Phi}_i$ hits an infinite number of times each state of $\mathcal{X}_i$. In general, the optimization of the throughput can lead to the suppression of a set of poorly performing links. In this case, a portion of the network incurs deadlock. The aforementioned property ensures that this does not happen for any policy associated with a separately irreducible process. In fact, it means that each source sends an infinite number of packets to each of its destinations over an infinite sample path and that the throughput of each link is strictly positive. Let us define the individual source i occupation time for a set of individual source states $C_i \subset \mathcal{X}_i$ as $O(C_i) = \sum_{t=0}^{\infty} \mathbf{1}\{\Theta_t = \theta : \phi_i \in C_i\}$, with $\theta = \{\phi_j, j \in \mathcal{S}'\}$. Thus, $O(C_i)$ is the number of times the single-source process $\mathbf{\Phi}_i$ hits the set of single-source set of states C_i.

Definition 3. *We say that the process $\mathbf{\Theta}$ is* separately irreducible *if $\mathcal{P}(O(\phi_i) < \infty) = 0$, $\forall \phi_i \in \mathcal{X}_i$, $\forall i = \mathcal{S}'$.*

Individual source processes $\mathbf{\Phi}_i$ are not Markov, since their transition probabilities do not depend only on the state of $\mathbf{\Phi}_i$ in the current frame. Thus, we can not properly define a Markovian transition kernel and use standard definitions and properties related to Markov processes. However, if $\mathbf{\Theta}$ is separately irreducible, then each single-source state $\phi_i \in \mathcal{X}_i$ *communicates* with every other single-source state in $\mathcal{X}_i$. By communicate we mean that there exists a sample path of the overall process $\mathbf{\Theta}$ with positive probability leading from a single-source state ϕ_i to another ϕ_i^* and vice versa. Note that this does not mean that $\mathbf{\Theta}$ hits any combination of single-process states infinite times. We clarify this point with an example. Consider a network with $B = 1$, $F = 3$ and two sources, namely i_1 and i_2, each with a single destination. Consider the single-source states $\phi_i^k = \{1, 1, k\}$, $i = 1, 2$, $k = 0, 1, 2$.[5] Define a policy μ' such that $\mu'(\{\phi_1^0, \phi_2^0\}) = (1, 0)$, *i.e.*, source 1 transmits, $\mu'(\{\phi_1^1, \phi_2^0\}) = (1, 1)$ and $\mu'(\{\phi_1^2, \phi_2^0\}) = (1, 1)$. It is easy to see that with this policy the state $\{\phi_1^1, \phi_2^0\}$ can not be reached by $\mathbf{\Theta}$, but that the single-source processes can hit any of the aforementioned states. In fact, from $\{\phi_1^0, \phi_2^0\}$ in case of failure the process moves to $\{\phi_1^1, \phi_2^0\}$ and then to $\{\phi_1^2, \phi_2^1\}$ (while in case of success the processes can move to empty buffer states or return in single-source states with zero transmissions). It is easy to see that state $\{\phi_1^1, \phi_2^0\}$ can not be reached even considering any combination of packet arrivals and queue length. Separate irreducibility is

[4] We remind that we are not considering secondary sources and their state.

[5] We recall that a single-source state is $\phi_i = \{b, d, f\}$, where b is the number of packets in the buffer, d is the destination and f is the number of transmissions.

a weaker connectivity index than irreducibility, that prescribes that the overall process hits every state $\theta \in \mathcal{X}$ an infinite number of times from any initial state.

The following proposition ensures that a persistent policy is associated with a separately irreducible process.

Proposition 4. *If μ' is a persistent policy, then $\mathbf{\Theta}$ is separately irreducible. (Proof in App. 5.9).*

However, before restricting the unconstrained search algorithm of the primary network to the persistent set $\mathcal{U}'_p$ we have to show that the optimal policy lies in the persistent set. This is guaranteed by the following proposition.

Proposition 5. *If the optimal policy $\widehat{\mu}'$ exists then it lies in the persistent set,* i.e., $\overline{\mu}' \in \mathcal{U}'_p$. *(Proof in App. 5.10).*

Thus, we can set our restricted policy space as $\mathcal{U}'_{\text{unichain}} = \mathcal{U}'_p$.

Moreover, the following holds:

Proposition 6. *If $\mu'_k \in \mathcal{U}'_p$ then $\mu'_{k+1} \in \mathcal{U}'_p$. (Proof in App. 5.11).*

Thus, if we initialized the unconstrained policy iteration algorithm with a policy μ^0 belonging to the persistent set, the algorithm produces a sequence of persistent policies. Therefore, all considered policies are unichain and the algorithm terminates finitely with the optimal policy (see Section 5.3). We can, for instance, initialize the algorithm with a greedy policy μ'_g defined on the immediate cost, that is $\mu'_g(\theta) : \min_{\mu' \in \mathcal{U}'(\theta)} \tilde{\delta}_{\mu'}(\theta)$, where nodes with empty buffer are forced to be silent. Note that μ'_g is a persistent policy. In fact, if $\mathcal{S}'_{\text{tx}}(\theta, \mu') \neq \emptyset$, any policy such that $\sum_{j \in \mathcal{S}'} [\mu'_g]_j > 0$ achieves an average cost $\tilde{\delta}_{\mu'_g}(theta)$ lower than the average cost $\tilde{\delta}_{\underline{0}'}(\theta)$ achieved by a policy where all sources are idle. Thus, the greedy policy provides transmission by at least one source in any state where there is at least one source with a non empty buffer, as required by persistent policy definition.

Let consider now the optimization of secondary users. We have to ensure that the overall process $\mathbf{\Theta}$ has a single recurrent class when performing the constrained optimization algorithm. In particular, we need that all the considered policies $\{\mu', \mu''\}$ are unichain. We observe that the structure of marginal process describing the primary network is not influenced by secondary access through the following proposition.

Proposition 7. *The classification of a state of $\theta' \in \prod_{i \in \mathcal{S}'} \mathcal{X}_i$ is the same in $\mathbf{\Theta}'$, defined as the process counting for only the primary sources, under policy $\widehat{\mu}'$ and in $\mathbf{\Theta}$ under policy $\{\widehat{\mu}', \mu''\}$.*

With this proposition, we meant that θ', *i.e.*, the aggregate of the states of all primary sources, has the same recurrence/transience characterization when considering the process describing primary sources neglecting secondary ones or the process describing primary and secondary sources, where primary sources use the same policy.

We omit the proof of this proposition. However, it is easy to show that, since the policy used by primary users is fixed, the state and the policy of the secondary network influence only the value of positive transition probabilities of $\mathbf{\Theta}'$, while the set of transitions with probability equal to zero remains the same. In this sense, the structure of the primary process is independent with respect to secondary users operations. Moreover, Proposion 7 implies that if empty-buffer states of the primary network are recurrent in the reduced primary process, they are recurrent also when we consider secondary transmissions. Therefore, we have a positive fraction of time in which primary sources are silent and secondary sources can transmit without any loss of performance of the primary network. Thus, as observed before, even with $\epsilon = 0$ the secondary network can achieve a positive throughput.

As for the constrained optimization, we define a class of policies similar to the persistent class of the unconstrained algorithm:

Definition 8. *we say that μ'' is a secondary persistent policy under the primary policy μ', and we write $\mu'' \in \mathcal{U}''_{\text{unichain}}(\mu_p)$, if the following holds for all $\theta \in \mathcal{X}$*

$$\sum_{i \in \mathcal{S}''_{\text{tx}}(\theta,\mu'')} [\mu''(\theta)]_{n_s} > 0 \text{ } \textit{if} \, \mathcal{S}''_{\text{tx}}(\theta, \mu'') \neq \emptyset, \tag{5.24}$$

$\forall \theta \in \mathcal{X}$ where $\mathcal{S}''_{\text{tx}}(\theta, \mu'')$ is the subset of secondary sources with at least one packet in their buffer associated with state θ.

As intuition suggests, if μ'' is secondary persistent under a persistent policy μ', then the chain has a single recurrent class. In particular, we get a globally persistent policy $\{\mu', \mu''\}$. In fact, while proving previous propositions, we showed that states in which all primary sources have an empty buffer are recurrent. Thus, states in which primary sources are silent occur with a finite time interval. We can, therefore sample the overall process Θ only in these states and show properties similar to that shown for primary sources alone.

Note that, in this case, $\boldsymbol{\Theta}$ is also separately irreducible. Analogously to the unconstrained case, it can be shown that, if the current policy is secondary persistent, the constrained policy iteration algorithm provides an improved secondary persistent policy at the next step. In order to o ensure the existence of the solution of the equations of the algorithm, we can initialize it with a greedy secondary policy, by which secondary users transmit only during idle frames of primary users. We observe that this initial policy suits the standard case in which secondary users access the channel only in white spaces left by primary users. Unfortunately, the algorithm is not guaranteed to terminate with the globally optimal secondary policy.

5.5.3 Numerical Results

We consider a single-hop network where primary and secondary sources are uniformly distributed in a circle of radius r. We assume that the destinations of the primary sources are uniformly placed in the same area, whereas, according to the common assumption that secondary users' links are short distance links, we place the secondary destinations within a circle of radius r_s centered on each secondary source. We present results where destinations use a conventional matched filter direct sequence code division multiple access DS-CDMA receiver and a successive interference cancellation (SIC) DS-CDMA receiver with spreading factor M. Packets are encoded with a rate $1/2$ convolutional code with polynomials 133_8 171_8, and then transmitted with BPSK modulation. In the MF DS-CDMA receiver case the post-processing SNIR perceived by receiver d_i given the set of the active sources, can be computed as:

$$\Gamma = \frac{p([\mu(\theta)]_i)\ell_{(i,d_i)}^{-\beta}}{(N_0 + \sum_{j:[\mu(\theta)]_j=1, j\neq i} p([\mu(\theta)]_j)\ell_{(j,d_i)}^{-\beta}/M)}, \tag{5.25}$$

where N_0 is the noise power and $\ell_{(j,d_i)}$ is the distance between source j and d_i. The SIC receiver iteratively decodes and cancels the incoming signals in decreasing received power order. We model the residual cancellation errors, the interference coming from the still-to-be-decoded signals and the effective noise power as in [20] and refer the interested reader to the same study for the approximated expression of the function mapping the useful and interfering received power of incoming signals to the perceived SNIR for the SIC DS-CDMA receiver.

The performance of the convolutional decoder with BPSK modulation is obtained through the Union Bound. In particular, the success probability conditioned on the perceived SNIR is $[1 - \sum_{w=w_{\text{free}}} A_w Q(\sqrt{w\Gamma})]^L$, where L is the packet length, A_w is the number of codewords of weight w, w_{free} is the free distance of the code and Q is the complementary Gaussian distribution function.

By the previous relations, one can easily define the function $\rho_i^{\mu}(\theta)$ which we used when listing transition probabilities and costs.

The power transmitted by an active source i is equal to $0.25u_i/3$[W], with $u_i = \{1, 2, 3\}$ ($u_i = 0$ corresponds to source idleness). The path loss exponent is $\beta = 3$. The circle ranges are (unless a different value is specified) $r = 120$[m] and $r_s = 40$[m]. The performance results presented in the following are obtained averaging the achieved throughput over 200 random topologies. We assume the transmission rate to be fixed and equal to 1[bit/s/Hz] for all the sources.

Fig. 5.1 shows the average per stage throughput achieved by the optimal policy in a scenario with 3 primary users as a function of the radius r for $F = 1$ and $F = 2$. In this plot, we assume that the set of secondary users is empty, and we set $B = 1$ and arrival rate equal to 1[packet/slot] per source. We also plot the throughput achieved by a scheme where only a single source per frame is allowed to access the channel, namely single access (SA) scheme.[6] It it possible to observe that both the receiver architectures achieve a throughput greater than that of the SA scheme. As the range decreases, *i.e.*, the area in which the nodes are placed becomes smaller, the throughput increases, as destinations become closer to the sources. However, as r becomes smaller, also the interfering sources get closer to other sources' intended receivers. In this setting, the superior resilience to interference of the SIC DS-CDMA receiver grants improved performance with respect to the MF DS-CDMA

[6] In this case the MF DS-CDMA and the SIC DS-CDMA are equivalent.

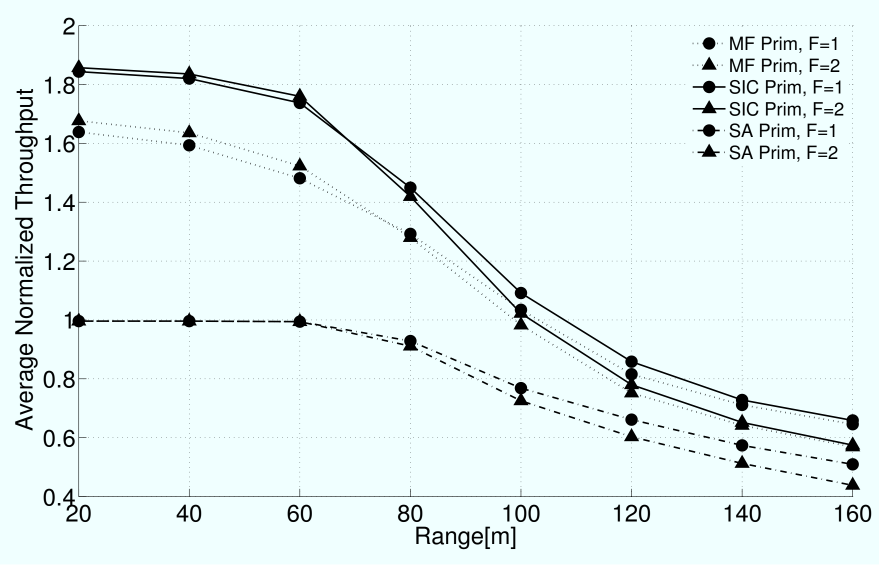


Figure 5.1. *Average per stage throughput as a function of the range of the area for a topology with three sources, each with* 2 *destinations.*

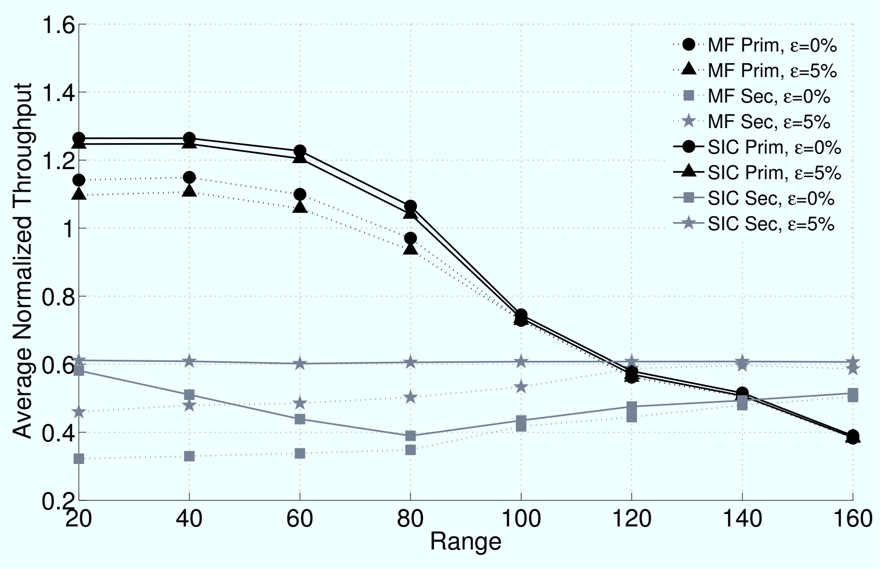


Figure 5.2. *Average per stage throughput as a function of the range of the area for a topology with two primary sources and one secondary source, each with* 2 *destinations.*

case. Note that the case with maximum number of transmissions equal to 2, *i.e.*, $F = 2$, achieves better performance than that with $F = 1$. In fact, in retransmission-based error control a source experiencing bad channel conditions towards a destination must exhaust all its available transmission attempts before it can try a different destination, and this results in an increased fraction of slots in which transmission failure occurs.

Fig. 5.2 depicts the average throughput per stage as a function of r for a scenario with two primary and one secondary source. We plot the achieved throughput for a maximum

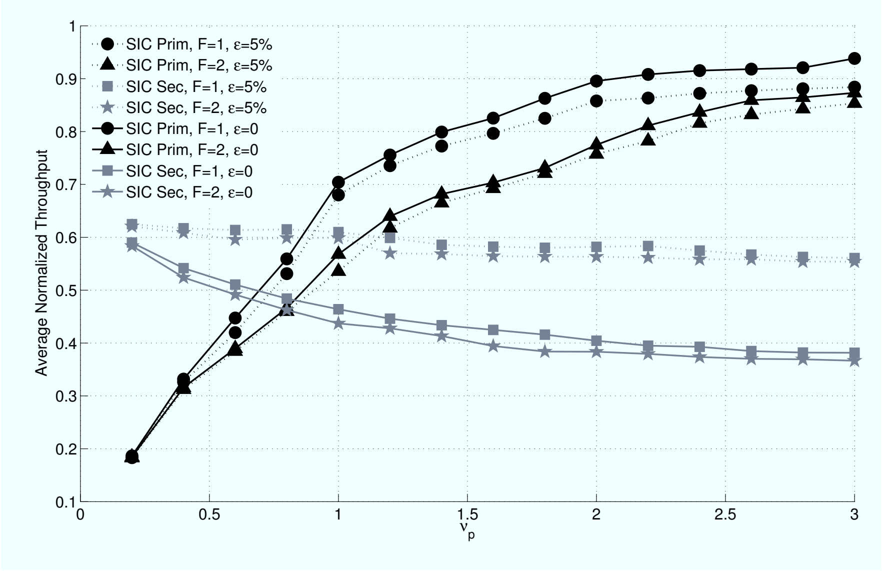


Figure 5.3. *Average per stage throughput as a function of the arrival rate of primary users for a topology with two primary sources and one secondary source, each with* 2 *destinations.*

allowed performance loss equal to 0 and 5% of the optimal throughput. As in the previous case, the throughput of the primary sources increases as the range decreases and the SIC DS-CDMA receiver shows a greater resilience to interference with respect to MF DS-CDMA for small values of r. It can be observed that even a small maximum performance loss, equal to 5% of the optimal throughput, enables a considerable throughput gain for the secondary user. It is also interesting to see that the throughput achieved by the SIC DS-CDMA receiver for $\epsilon = 0$ initially decreases and then increases as the network becomes larger. This is due to interference cancellation, that allows the effective cancellation of strong interferers, whereas, when the power of the interfering sources is comparable to that of the intended signal, the decoding of the latter suffers residual cancellation errors. Note that the throughput of the secondary user, when the MF DS-CDMA receiver is used, is less than the throughput achieved with the SIC DS-CDMA receiver, as the effective interference generated by the transmission of the secondary user is higher.

Fig. 5.3 shows the average throughput per stage as a function of the arrival rate of the primary users. As expected, as the traffic of the primary users is increased, the throughput of the secondary user decreases. Nevertheless, the secondary user can effectively superpose its transmissions to that of primary users even when the latter occupy a significant fraction of the frames. Note that a small performance loss of primary users allows a considerable improvement of the throughput achieved by the secondary user.

5.6 Conclusions

In this paper, we addressed the optimization of the control for a wireless network where users simultaneously access the channel. We focused in particular on the interaction among the stochastic processes describing the various sources of the network. We defined a novel interference measure called process distortion. We presented an optimization framework for a scenario where users are divided into two groups with different priorities and transmission by users with the lower priority is constrained by a bound on the distortion generated to the stochastic process modeling the higher priority group. As an example, we applied our framework to a cognitive network with primary and secondary users and presented numerical results for this scenario. Although our proposal is based on centralized control, we believe that it represents an interesting framework, that opens up to distributed protocols.

5.7 Appendix 2.A: Useful definitions, properties and observations

Here we recall some definitions and properties of stationary Markov processes $\mathbf{\Theta}$ with a countable state space and probability measures defined as in Section 5.3. We follow, for most of the section, the notation and definitions of [17]. We consider states $\theta, \theta^1, \theta^2, \theta^3 \in \mathcal{X}$ and the set of states $\mathcal{A}, \mathcal{A}_1 \subseteq \mathcal{X}$. For the sake of simpler notation, in the following we drop the indication of the policy in the notation.

Definition 9. *For any set $C \subseteq \mathcal{X}$), the variables*

$$\begin{aligned} O(\mathcal{A}) &:= \sum_{t=1}^{\infty} \mathbf{1}\{\Theta_t \in \mathcal{A}\}, && (5.26)\\ H_{\mathcal{A}}(\mathcal{A}_2) &:= \min\{t \geq 0 : \Theta_t \in \mathcal{A}_1 | \Theta_0 \in \mathcal{A}\}, && (5.27) \end{aligned}$$

are the occupation time of C and the first hitting times on $\mathcal{A}_1$ from $\mathcal{A}$, respectively [17].

Definition 10. *We define the kernel U and the hitting probability as [17]*

$$\begin{aligned} U(\theta, \mathcal{A}) &:= \sum_{z=1}^{\infty} \zeta_{\mu}^{z}(\theta, \mathcal{A}) = E_{\theta}[O(\mathcal{A})], && (5.28)\\ L(\theta, \mathcal{A}) &:= \mathcal{P}(H_{\theta}(\mathcal{A}) < \infty). && (5.29) \end{aligned}$$

We say that A *leads to* the set $\mathcal{A}_1$, which we write $\mathcal{A} \rightarrow \mathcal{A}_1$ if $L(\mathcal{A}, \mathcal{A}_1) > 0$, and that $\mathcal{A}$ and $\mathcal{A}_1$ *communicate*, written $\mathcal{A} \leftrightarrow \mathcal{A}_1$, if $L(\theta, \mathcal{A}_1) > 0$, $\forall \theta \in \mathcal{A}$, and $L(\theta^1, \mathcal{A}) > 0$, $\forall \theta^1 \in \mathcal{A}_1$. By convention, a state leads to itself, *i.e.*, $\theta \rightarrow \theta$. Note that $L_{\mu}(\theta, A) > 0$ is true if and only if there exists a path with positive probability that leads from θ to A, *i.e.*, $\exists z : \zeta^{z}(\theta, \mathcal{A}) > 0$. If $\mathcal{A}$ does not lead to $\mathcal{A}_1$ we write $\mathcal{A} \not\rightarrow \mathcal{A}_1$.

Property 11. *The relation $\leftrightarrow$ is an equivalence relation, and so the equivalence classes $\mathcal{C}(\theta) = \{\theta^1 : \theta \leftrightarrow \theta^1\}$ cover $\mathcal{X}$ [17].*

Definition 12. *If $\mathcal{C}(\theta) = \mathcal{X}$ for some θ then we say that $\mathcal{X}$ is irreducible [17].*

Definition 13. *We say $\mathcal{C}(\theta)$ is absorbing if $\zeta_{\mu}(\theta^1, \mathcal{C}(\theta)) = 1$ for all $\theta^1 \in \mathcal{C}(\theta)$.*

Note that if $C = \mathcal{C}(\theta)$ is absorbing, then there is an irreducible Markov chain $\mathbf{\Theta}_C$ whose state space is restricted to C.

Definition 14. *The state θ is called transient if $E_\theta[O(\theta)] = U(\theta,\theta) < \infty$ and recurrent if $E_\theta[O(\theta)] = U(\theta,\theta) = \infty$ [17].*

If a class is transient, the process visits its states a finite number of times before leaving the class and does not return. Thus, if $\mathcal{C}(\theta)$ is transient and $\Theta_{t_0} \in \mathcal{C}(\theta)$, there exists an index $t_1 \geq t_0$ such that $\Theta_{t_1} \in \mathcal{C}(\theta)$ and $\Theta_{t_2} \notin \mathcal{C}(\theta)$, $\forall t_2 > t_1$. Note that if θ is transient, then $\mathcal{C}(\theta)$ is transient (*i.e.*, all states of $\mathcal{C}(\theta)$ are transient). If θ is recurrent, then $\mathcal{C}(\theta)$ is recurrent.

Property 15. *For any $\theta \in \mathcal{X}$, $U(\theta,\theta) = \infty$ if and only if $L(\theta,\theta) = 1$ [17].*

Property 16. *If $\mathcal{X}$ is finite, we have at least one recurrent class.*

Proof. Let consider the set of classes $C_k \subseteq \mathcal{X}$, $k = 0,\ldots,I-1$, with $\bigcup \mathcal{C}_k = \mathcal{X}$ and $\mathcal{C}_k \cap \mathcal{C}_q = \emptyset$, $\forall k,q = 0,\ldots,I-1$, $k \neq q$. Note that $I \leq |\mathcal{X}| < \infty$. Assume that all classes are transient, and let us focus on a state $\theta_0 \in \mathcal{C}_{k_0}$ and set it as the initial state, *i.e.*, $\Theta_0 = \theta_0$. If a state is transient, then the whole class is transient and the process will visit the states of $\mathcal{C}_{k_0}$ a finite number of times. Thus, $\exists t_0 < \infty : \Theta_{t_0} \in \mathcal{C}_{k_0}, \Theta_t \notin \mathcal{C}_{k_0} \forall t > t_0$. Assume $\Phi_{t_0+1} \in \mathcal{C}_{k_1}$, then there exists an index $t_1 < \infty$ such that $\theta_t \notin \mathcal{C}_{k_0} \cup C_{k_1}$, $\forall t > t_1$. In this way, we create a sequence of classes $\{k_0,\ldots,k_{I-1}\}$, with $k_i \neq k_j$ and $\{k_u, u = 0,\ldots,I-1\} = \{0,\ldots,I-1\}$ with an associated index $t_{I-2}+1$ where the process enters class $\mathcal{C}_{k_{I-1}}$ and can not return to any state of $\bigcup_{k_u, u=0,\ldots,I-2} \mathcal{C}_{k_u}$. Thus, the process is trapped into $\mathcal{C}_{k_{I-1}}$ and will visit an infinite number of times its states, that contradicts the hypothesis that all classes are transient. □

Property 17. *If $\exists \theta^1 \in \mathcal{C}(\theta) : \theta^1 \rightarrow \mathcal{A}$ then $\mathcal{C}(\theta) \rightarrow \mathcal{A}$.*

Proof. Since $\theta^1 \rightarrow \mathcal{A}$, then $L_\mu(\theta,\mathcal{A}) > 0$. By definition of class, we have $L(\theta^1,\theta) > 0$, $\forall \theta^1 \in \mathcal{C}(\theta)$. Therefore, $L(\theta^1,\mathcal{A}) > 0$, $\forall \theta^1 \in \mathcal{C}(\theta)$. □

Property 18. *If $\mathcal{C}(\theta)$ and $\mathcal{C}(\theta^1)$ are two recurrent classes then $\mathcal{C}(\theta) \not\rightarrow \mathcal{C}(\theta^1)$ and $\mathcal{C}(\theta^1) \not\rightarrow \mathcal{C}(\theta)$, or $\mathcal{C}(\theta) = \mathcal{C}(\theta^1)$.*

Proof. Assume that both $\mathcal{C}(\theta)$ and $\mathcal{C}(\theta^1)$ are recurrent and thus $U_\mu(\theta^2,\theta^2) = \infty$, $\forall \theta^2 \in \mathcal{C}(\theta)$ and $U(\theta^3,\theta^3) = \infty$, $\forall \theta^3 \in \mathcal{C}(\theta^1)$. Assume that $\exists \theta^2 \in \mathcal{C}(\theta) : \theta^2 \rightarrow \mathcal{C}(\theta^1)$. If $\Theta_0 \in \mathcal{C}(\theta^1)$, then $\exists t : \Theta_t \in \mathcal{C}(\theta)$. If $\mathcal{C}(\theta^1) \not\rightarrow \mathcal{C}(\theta)$, then $\nexists t_1 > t : \theta_{t_1} \in \mathcal{C}(\theta)$, that contradicts the hypothesis that $\mathcal{C}(\theta)$ is recurrent ($L(\mathcal{C}(\theta),\mathcal{C}(\theta)) = 1$, that is equivalent to say that there exists a path leading from $\mathcal{C}(\theta)$ to $\mathcal{C}(\theta)$). If $\mathcal{C}(\theta^1) \rightarrow \mathcal{C}(\theta)$, then $\mathcal{C}(\theta) \leftrightarrow \mathcal{C}(\theta^1)$ and $\mathcal{C}(\theta)$ and $\mathcal{C}(\theta^1)$ are the same class. □

From the previous property, if a class is recurrent then it is absorbent, while classes that lead to recurrent classes are transient.

Definition 19. *We say that a policy μ is unichain if the associated chain has only one recurrent class, [16].*

5.8 Appendix 2.B: Proof of Proposition 2

To prove that each persistent policy is unichain, we first show that from any $\theta \in \mathcal{X}$ it is possible to reach every state in which all sources have an empty buffer. Let as consider a reference state $\theta^r = \{\{0, d_i^r, 0\},\ i \in \mathcal{S}\}$ and assume that the process $\mathbf{\Theta}$ is in a general initial state $\Theta_0 = \theta = \{\{b_i, d_i, f_i\},\ i \in \mathcal{S}\}$. Since $\alpha(0) > 0, \forall i \in \mathcal{S}$, we can move the process on a positive probability path with no arrivals at any source. Since the policy is persistent, if the set of sources with non empty buffer $\mathcal{S}_{\mathrm{ne}}(\Theta_t, \mu)$ is non empty,[7] then the set of transmitting sources $\mathcal{S}_{\mathrm{tx}}(\Theta_t, \mu) \subseteq \mathcal{S}_{\mathrm{ne}}(\Theta_t, \mu)$ is non empty. If we consider non degenerate error probability, i.e., $0 < \rho_i^{\mu}(\Theta)$ all transmitted packets in a frame are successfully delivered to their intended destination with positive probability. Thus, we can remove in each frame t from the buffers $|\mathcal{S}_{\mathrm{tx}}(\Theta_t, \mu)|$ packets. Each source selects with positive probability the same destination d_i with positive probability. Therefore, in $t_1 \leq\ |\mathcal{S}_{\mathrm{tx}}(\Theta_t, \mu)|B \leq |\mathcal{S}|B$ steps the process moves with positive probability to state $\theta^1 = \{\{0, d_i, 0\}, i \in \mathcal{S}\}$, *i.e.*, where all sources have empty buffer and with last served destination equal to d_i.

Since $\alpha(1) > 0$ and the destination of the next packet is randomly chosen, from θ^1 the process moves with positive probability to a state $\theta^2 = \{\{1, d_i^r, 0\}, i \in \mathcal{S}\}$ in one step. With a path similar to that from θ to θ^1 the process can then move from θ^2 to θ^r in at most $t_2 = |\mathcal{S}|$ steps. Thus we have $L(\theta, \theta^r) > \zeta_{\mu}^{t_1}(\theta, \theta^1)\zeta_{\mu}^{t_2+1}(\theta^1, \theta^r) > 0$ and $\theta \rightarrow \theta^*$, $\forall \theta \in \mathcal{X}$. θ is a general state of $\mathcal{X}$ and θ^r is a state of the empty buffer set $\mathcal{E} = \prod_{i \in \mathcal{S}} \mathcal{E}_i$, $\forall d_i \in \mathcal{D}_i$, where $\mathcal{E}_i = \{\phi_i : b_i = 0\}$. Hence, any state of $\mathcal{X}$ leads to any empty buffer state in $\mathcal{E}$ and $L(\theta, \mathcal{E}) > 0$ and $\theta \rightarrow \mathcal{E}$, for any $\theta \in \mathcal{X}$. Note that, since $\mathcal{E} \subseteq \mathcal{X}$, we have that $L(\mathcal{E}, \mathcal{E}) > 0$ and therefore $\mathcal{E}$ is a subset of a recurrent class.

Due to Property 18, we have that two non identical recurrent classes do not admit a path with positive probability leading from one class to the other and vice versa and therefore we have less than two recurrent classes in $\mathcal{X}$. Moreover, since $\mathcal{X}$ is finite, Property 16 ensures that there exists at least one recurrent class. The number of recurrent classes in the chain is therefore exactly 1.

It can not be guaranteed that the process is able to reach any service state from a state of $\mathcal{E}$. We can decompose $\mathcal{X}$ into two disjoint sets covering $\mathcal{X}$: the set $\mathcal{A}_{\mathrm{reach}} \subseteq \mathcal{X}$ whose elements are the states reachable by $\mathcal{E}$, *i.e.*, $\mathcal{A}_{\mathrm{reach}} = \{\theta : L(\mathcal{E}, \theta) > 0\}$, and the set $\mathcal{A}_{\mathrm{nreach}}$ whose elements are the states that can not be reached by E, *i.e.*, $A_{\mathrm{nreach}} = \{\theta : L(E, \theta) = 0\}$. It is easy to show that $\mathcal{A}_{\mathrm{reach}} \supset \mathcal{E}$ is the recurrent class and A_{nreach} is covered by a certain number of transient classes. Moreover, in $\mathcal{A}_{\mathrm{reach}}$ there is a positive number of states in which the buffer of any subset of sources is non-empty, and in which the persistent policy provides the transmission of at least one source.

[7] Note that here we are not focusing on a group in particular, and thus we drop the superscripts.

5.9 Appendix 2.C: Proof of Proposition 4

The overall process $\mathbf{\Theta}$ is separately irreducible if each individual source process hits infinite times every individual source state $\phi_i \in \mathcal{X}_i$ or, equivalently, $\mathbf{\Theta}$ hits a state Θ_t with i-th component ϕ_i infinite times for any ϕ_i and from any initial state Θ_0. We showed in the previous proof that $\mathcal{E}$ is a recurrent set in $\mathbf{\Theta}$ and that $L(\Theta, \mathcal{E}) > 0, \forall \theta \in \mathcal{X}$. The i–th component of both $\mathcal{E}$ and θ can be any individual source state in $\mathcal{E}_i$ and $\mathcal{X}_i$, respectively. Therefore, there exists an individual source process path with positive probability leading from a individual source state $\phi_i \in \mathcal{X}_i$ to any state ${\phi_i}^* \in \mathcal{E}_i$. We have now to show that from any state $\Theta_0 = \Theta^* \in \mathcal{E}$ the process can move to a state $\Theta_t = \phi^{**} \in \mathcal{X}$ with any possible individual source state $\phi_j^{**} = \{b_j^{**}, d_j^{**}, f_j^{**}\}$ for a fixed j. If this holds, there exists a sample-path with non-zero probability connecting two overall states having the same j–th component.[8]. Therefore ϕ_j is hit infinite times from any overall state in $\mathcal{X}$, since any state leads to $\mathcal{E}$. We exploit also in this case the possibility of zero arrivals with positive probability. Then, with positive probability we have zero arrivals in the buffers of sources $i \in \mathcal{S} \setminus j$. With positive probability we can also have $u_i = b_j^{**}$ arrivals in the buffer of the source j and the source selects d_j^{**} as destination. Thus $\Theta_t = \{\Phi_i = {\phi_i}^*, \forall i \in \mathcal{S} \setminus j, \Phi_j = \{b_j^{**}, d_j^{**}, 0\}\}$. Since the policy is persistent, and j is the only source with a packet in the buffer, then $[\mu(\Theta_t)]_j = 1$. Note that there is a positive probability that transmission by node j fails. Thus, the process can move with positive probability to Θ_t through a path with no more arrivals and f_j^{**} failed transmissions.

5.10 Appendix 2.D: Proof of Proposition 5

By this proposition, the optimal policy $\widehat{\mu}$ lies in the persistent policies set $\mathcal{U}_p'$. Let us focus on the primary group of users. In order to prove this, we show that for any non-persistent policy $\mu_{np} \in \mathcal{U}' \setminus \mathcal{U}_p'$ there exists a policy $\mu_p \in \mathcal{U}_p'$ such that $J^{\mu_{np}'}(\Theta_0) > J^{\mu_p'}(\Theta_0)$, for any Θ_0. Since μ_{np}' is non-persistent, there exists a set of states C^* where i has a non-empty buffer and $\mu_{np}'[\theta] = 0^{S'}$. We prove that for a given arrival sample-path[9] the total cost incurred by the process when using a non-persistent policy is strictly greater than the cost incurred by the process when using a particular persistent policy. The same inequality holds for the average cost per stage. In fact, the average cost per stage is equal to the average over all the possible arrivals sample-path of the total cost over a time range growing to infinity divided by the time range. For the sake of simplicity, we will provide an in depth discussion for the case in which there is a single source and a sketch of the proof for the multiple sources case.

Thus, as a first step, let us consider a network with a single source i, where $\mathbf{\Theta} = \mathbf{\Phi}_i$. Let us assume there exists a state $\theta^r = \{b^r, d^r, f^r\}$, with $b^* > 0$, where θ^* is a non-absorbing state. We construct the persistent policy μ_p' as $\mu_p'(\theta) = \mu_{np}'(\theta)$, $\theta \neq \theta^r$, and $\mu_p'(\theta^r) = 1$. We choose θ^r as initial state, *i.e.*, $\Theta_0 = \phi^r$ and we compare the sample-paths of the processes

[8] We can exchange θ and θ^{**}

[9] *i.e.*, the sample-path of the process whose state in a frame is associated with the number of packet arrivals in the frame.

associated with the two policies. In particular, we consider sample-paths with arrivals-path equal to $\chi_t = \{m_0, m_1, m_2, \dots, m_t\}$, where m_k is the number of packets arrived in the buffer of i in frame k. Note that an arrival does not depend on the node's decisions or state. In the following, we refer to the process associated to μ'_{np} and μ'_p as $\mathbf{\Theta}_{\mu'_{np}}$ and $\mathbf{\Theta}_{\mu'_p}$, respectively. We define the functions $T_{k_1}^{k_2}(\chi_t) = \min\{k_1 < k < k_2 : \sum_{z=k_1-1}^{k-1} \mathbf{1}(\mathbf{m_z} > \mathbf{0}) > \mathbf{0}\}$[10] and $Z_{k_1}^{k_2}(\chi_t) = \sum_{z=k_1}^{k_2-1} m^z$. Thus, $T(\chi_t)$ and $Z_k(\chi_t)$ track the time until a frame occurs with a positive number of arrivals and the total number of arrivals until frame k_2 from frame k_1, respectively. We now define as $\mathcal{W}_{\chi_t}(0, k, b)$ the average cost incurred by the process in the sample-path from frame 0 to k, and b is the initial number of packets in the buffer. We show in the following that $\mathcal{W}_{\chi_t}^{\mu'_{np}}(0, k, b^*) > \mathcal{W}_{\chi_t}^{\mu'_p}(0, k, b^*)$, $\forall k$, where the superscript denotes the policy used. Let us define the probabilities

$$V_d(\ell) = \begin{cases} (1-\rho_d)^{\ell-1}\mathcal{G}_d & \text{if } 1 < \ell < F \\ (1-\rho_d)^{\ell-1} & \text{if } \ell = F, \end{cases} \tag{5.30}$$

where ρ_d is the correct decoding probability of a packet sent by i to destination d, $d \in \mathcal{D}_i$. Thus, if $\ell < F$, $V_d(\ell)$ is the probability that a packet is delivered to d in ℓ transmissions, while, if $\ell = F$, it represents the probability that the source performs F transmissions. When a packet is served, the cost to exit the service in ℓ transmissions is

$$Y_d(\ell) = \begin{cases} (\ell-1)R & \text{if } 0 < \ell < F \\ \ell R - \rho_d & \text{if } \ell = F, \end{cases} \tag{5.31}$$

We compute the total cost through a recursive expression. Consider first policy μ' and $f^* = 0$. $W_{\chi_t}^{\mu'_p}(0, k, b^r)$ can be written as in Eq. (5.32), where the remaining costs $W_{\chi_t}^{\mu'_p}(k_1, k_2, b^r, d^r)$ admit the recursive definition of Eq. (5.33) and $\lceil x \rceil = \min\{B, x\}$. The function $Q_{k_1}^{k_2}(b, \chi_t) = \lceil \lceil b + Z_{k_1}^{k_2-1}(\chi_t) \rceil + m_{k_2-1} - 1 \rceil$ denotes the number of packets in the buffer after the end of a service, where we take into account for the packet removal.

$$\begin{aligned} W_{\chi_t}^{\mu'_p}(0, k, b^r) &= V_{d^*}(F)\left[Y_{d^*}(F) + \prod_{d^*} \frac{1}{|\mathcal{D}_i|} W_{\chi_t}^{\mu'_{np}}(F+1, k, Q_0^F(b^*, \chi_t), d^*)\right] \\ &+ \sum_{\ell=1}^{F-1} V_{d^*}(\ell)\left[Y_{d^*} + \prod_{d^*} \frac{1}{|\mathcal{D}_i|} W_{\chi_t}^{\mu'_{np}}(\ell+1, k, Q_0^\ell(b^r, \chi_t), d^*)\right] \end{aligned} \tag{5.32}$$

$$W_{\chi_t}^{\mu'_p}(k_1, k_2, b, d) = \begin{cases} 0 & \text{if } b = 0, T_{k_1}^{k_2}(\chi_t) = 0 \\ 0 \cdot T_{k_1}^{k_2}(\chi_t) + W_{\chi_t}^{\mu'_{np}}(T_{k_1}^{k_2}(\chi_t)+1, k_2, \lceil b + Z_0^{T_{k_1}^{k_2}(\chi_t)}(\chi_t) \rceil, d) & \text{if } b = 0, T_{k_1}^{k_2}(\chi_t) > 0 \\ V_{d^*}(F)\left[Y_{d^*}(F) + \prod_{d^*} \frac{1}{|\mathcal{D}_i|} W_{\chi_t}^{\mu'_{np}}(F + k_1 + 1, k_2, Q_{k_1}^{k_1+F}(b, \chi_t), d^*)\right] + & \\ + \sum_{\ell=1}^{F-1} V_{d^*}(\ell)\left[Y_{d^*} + \prod_{d^*} \frac{1}{|\mathcal{D}_i|} W_{\chi_t}^{\mu'_{np}}(\ell + k_1 + 1, k_2, Q_{k_1}^{k_1+\ell}(b, \chi_t), d^*)\right] & \text{if } b > 0, k_2 - k_1 \geq F - 1 \\ \sum_{\ell=1}^{k_2-k_1} V_{d^*}(\ell)\left[Y_{d^*} + \prod_{d^*} \frac{1}{|\mathcal{D}_i|} W_{\chi_t}^{\mu'_{np}}(\ell + k_1 + 1, k_2, Q_{k_1}^{k_1+\ell}(b, \chi_t), d^*)\right] & \text{if } b > 0, k_2 - k_1 < F - 1 \end{cases} \tag{5.33}$$

$$W_{\chi_t}^{\mu'_{np}}(0, k, b^r) = \begin{cases} R \cdot k & \text{if } T_0^k(\chi_t) = 0 \\ RT_0^k(\chi_t) + W_{\chi_t}^{\mu_{np}}(T_{k_1}^{k_2}(\chi_t)+1, k_2, \lceil b^r + Z_0^\ell(\chi_t) \rceil, d^r) & \text{otherwise.} \end{cases} \tag{5.34}$$

[10] We set $T_{k_1}^{k_2}(\chi_t) = 0$ if there are no arrivals in $[k_1, k_2]$.

$$\begin{cases} RT_{k_1}^{k_2}(\chi_t)+C_{d^*}(F)\left[Y_{d^*}(F)+\prod_{d^*}\frac{1}{|\mathcal{D}_i|}W_{\chi_t}^{\mu_p}(T_{k_1}^{k_2}(\chi_t)+F+k_1+1,k_2,Q_{k_1}^{k_1+F}(b^r,\chi_t),d^*)\right]+ & \\ +\sum_{\ell=1}^{F-1}V_{d^*}(\ell)\left[Y_{d^*}+\prod_{d^*}\frac{1}{|\mathcal{D}_i|}W_{\chi_t}^{\mu_p}(T_{k_1}^{k_2}(\chi_t)+\ell+k_1+1,k_2,Q_{k_1}^{k_1+\ell}(b^r,\chi_t)-1\rceil,d^*)\right] & \text{if } k_2-k_1\geq F-1 \\ RT_{k_1}^{k_2}(\chi_t)+\sum_{\ell=1}^{k_2-k_1}C_{d^*}(\ell)\left[Y_{d^*}+\prod_{d^*}\frac{1}{|\mathcal{D}_i|}W_{\chi_t}^{\mu_p}(T_{k_1}^{k_2}(\chi_t)+\ell+k_1+1,k_2,Q_{k_1}^{k_1+\ell}(b^r,\chi_t),d^*)\right] & \text{if } k_2-k_1<F-1 \end{cases} \tag{5.35}$$

Thus, from the initial state[11] the source performs a number of transmissions distributed according to C_{d^r}, incurring a cost defined by Y_{d^r}. Then, the remaining cost is computed through the recursive function $W_{\chi_t}^{\mu_p'}(k_1,k_2,b,d)$, representing the average cost the source incurs in $[k_1,k_2]$ with b packets in the buffer and with next destination equal to d. If the buffer is empty in k_1, the process spends the period until the next arrival (determined by $T_{k_1}^{k_2}(\chi_t)$) in one of the empty-buffer states with cost equal to zero. Then, a new service starts, the function is invoked with $b>0$ and the source performs a number of transmissions distributed according to C_d with cost defined by Y_d. If $f>0$, the definition must be slightly modified to take into account the reduced number of transmissions of the initial service. If policy μ is used, the cost is as reported in Eq. (5.34). In this case, until new packets arrive in the buffer[12], the process is trapped in θ and accumulates a cost equal to R in each frame. The recursive cost formulation is similar to Eq. (5.33), but if $d=d^r$ and $b=b^r$ we have to take into account frames in which the process is again trapped into θ^*, and we use Eq. (5.35).

We observe that C_d and Y_d are equal for both the policies and the sequence of the destination of the packets being served is the same. Therefore, the cost incurred during the frames in which the source transmits is the same. Moreover, the average cost incurred during transmitting frames is lower than R. In fact, a service is a sequence of frames in which the process accumulates a cost R and a final frame with cost equal to 0 or $(1-\rho_d)$. The difference between the two policies lies in the number of served packets and on the cost incurred during idle frames. It is clear from Eq. (5.34) that μ_{np}' incurs an initial delay (that occurs every time the process hits θ^*) in which it accumulate the maximum possible cost R in each frame. Policy μ_p' starts immediately the sequence of transmissions. From the equations governing the cost in the two policies it is possible to derive the following observations: in a given frame k

- the number of packets in the buffer of $\mathbf{\Theta}_{\mu_{np}'}$ is higher than or equal to the number of packets in the buffer of $\mathbf{\Theta}_{\mu_p'}$;
- the number of packets served by policy μ_{np}' is lower than or equal to the number of packets served by μ_p'.

From the former observation we can infer that, if in a frame k the process associated with policy μ_{np}' is in an empty buffer state (with cost zero), also the process associated with μ_p' is in an empty buffer state. Thus, the number of frames in which $\mathbf{\Theta}_{\mu_{np}'}$ incurs a cost equal to zero is bounded by the number of analogous frames of $\mathbf{\Theta}_{\mu_p'}$. The latter observation results in the conclusion that $\mathbf{\Theta}_{\mu_p'}$ spends a greater amount of time in frames with a cost lower than R with respect to $\mathbf{\Theta}_{\mu_{np}'}$. Therefore, since in remaining frames not associated with an

[11] Where the source is serving a packet intended to d^r

[12] We observe that, if we consider a single source, θ is absorbing if $b=B$. Thus, $b<B$ and, since ϕ is the only one where $b>0$ and the source does not transmit, the process can move from θ with positive probability.

empty buffer or packet transmission $\boldsymbol{\Theta}_{\mu'_{np}}$ accumulates a cost equal to R, we conclude that $W^{\mu'_{np}}_{\chi_t}(0,k,b^r) > W^{\mu'_p}_{\chi_t}(0,k,b^r)$.

Let us clarify this point through an example. Consider the service of the initial packet. Let say policy μ'_p terminates the service in l frames, $1 \le l \le F$, with$Q^l_0(b^r,\chi_t)$ packets in the buffer. Policy μ' terminates the service in $T^k_0(\chi_t)+l$ frames with $Q_0^{T^k_0(\chi_t)+l}(b^r,\chi_t) \ge Q^l_0(b^r,\chi_t)$ packets in the buffer, where the inequality holds since $T^k_0(\chi_t) > 0$ and $m_k \ge 0$. In the remaining $T^k_0(\chi_t)$ frames, $\boldsymbol{\Theta}_{\mu'_p}$ can either stay in an empty buffer state or serve another packet depending on the number of packets in the buffer. Thus, $\boldsymbol{\Theta}_{\mu'_p}$ spends this time incurring an average total cost that is in any case lower than $T^k_0(\chi_t)R$, that is the cost incurred by $\boldsymbol{\Theta}_{\mu'_{np}}$ during the frames in which it is in ϕ^*. After frame $T^k_0(\chi_t)+l$, $\boldsymbol{\Theta}_{\mu'_{np}}$ can start the service $\boldsymbol{\Theta}_{\mu'_p}$ started in frame $l+1$, or stay idle if the buffer is empty (and in this case also $\boldsymbol{\Theta}_{\mu'_p}$ has an empty buffer, since it removed at least the same number of packets of $\boldsymbol{\Theta}_{\mu'_{np}}$). $\boldsymbol{\Theta}_{\mu'_p}$ can start the service of new packets or keep idle if the buffer is empty. Thus, $\boldsymbol{\Theta}_{\mu'_p}$ has an advantage of some frames with respect to $\boldsymbol{\Theta}_{\mu'_{np}}$ in terms of number of transmissions, and therefore it removes a higher number of packets from the buffer.

Finally, since $W^{\mu'_{np}}_{\chi_t}(0,k,b^r) > W^{\mu'_p}_{\chi_t}(0,k,b^r)$, $\forall k$, we have

$$J_{\mu'_{np}}(\theta^r)\mathrm{E}_{\chi_k}\left[\lim_{k\to+\infty}\frac{1}{k}W^{\mu'}_{\chi_k}(0,k,b^r)\right] > \mathrm{E}_{\chi_k}\left[\lim_{k\to+\infty}\frac{1}{k}W^{\mu'_p}_{\chi_k}(0,k,b^r)\right] = J_{\mu'_p}(\theta^r), \tag{5.36}$$

since χ_k is independent of the policy. Moreover $J_{\mu'_{np}}(\theta) > J_{\mu'_p}(\theta)$, $\forall\theta \in \mathbb{X}$. In fact, it is easy to show that if θ^r is the only state with $b>0$ such that $\mu'(\theta)=0$ then $\theta \to \theta^r$, $\forall\theta \ne \theta^r$. Since the two policies are equal in all states $\theta \ne \theta^r$, the average cost and number of frames to reach θ^r is equal and the inequality still holds. It is also easy to show that the inequality holds if there are multiple non-absorbing states θ with a non empty buffer and $\mu'(\theta)=0$.[13] In fact, the computation of the cost $W^{\mu'_{np}}_{\chi_t}$ is similar, but we have that the associated process incurs a delay analogous to that of θ^r in each of those states.

To conclude the discussion of the single source case, we have to prove that the average cost of the non-persistent policy is greater than that incurred by its associated persistent version when there exists absorbing states in C. In this case, it is sufficient to observe that if θ^r is an absorbing state, then all states in which the source transmits are transient. In fact, states in $\mathcal{X}\setminus C$ can reach C[14] with positive probability and therefore the process is eventually trapped in an absorbing state of C in a finite time. Hence, the average cost per stage incurred by $\boldsymbol{\Theta}_{\mu'_{np}}$ is equal to R. μ'_p is persistent, and thus there exists a non-empty set of recurrent states in which the source transmits. Hence, the average cost per stage incurred by $\boldsymbol{\Theta}_{\mu'_p}$ is strictly lower than R.

If we have more than one source, the proposition can be proved in a similar way. For clarity we do not include this case. We observe some differences between the single and multisource cases. A non-persistent policy admits states θ^r with $\mathcal{S}'_{\mathrm{tx}}(\theta^r,\mu'_{np}) \ne \emptyset$ and $\mathcal{S}'_{\mathrm{ne}}(\theta^r,\mu'_{np}) =$

[13] And in this case we have to modify the policy in all those states.

[14] It can be proved through a rationale similar to that presented in the previous proofs.

$\emptyset$. Analogous to the previously discussed case, the process accumulates a cost equal to $\mathcal{S}'_{tx}(\theta,\mu)R$ until the arrival of new packets which move the process to other states. If we change the non persistent into a persistent policy, setting the decision variable of one of the sources with a non-empty buffer to one, we can, as in the previous case, *accelerate* the service of the packets. The transmitting source delivers its packet with a positive probability. Thus, the process incurs a lower immediate average cost and, moreover, the decoding probability of other nodes is increased by the potential idleness of this source in the next frames. In this case, in order to prove the proposition, we have to construct a more involved persistent policy, remapping the policy in order to preserve the sequence of the served packets. However, the inequalities on the number of served packets and the buffer level also hold in this case. Moreover, a similar rationale can be applied to secondary users.

5.11 Appendix 2.E: Proof of Proposition 6

By this proposition, if the policy at the k–th iteration of policy iteration algorithm μ'_k is persistent, and thus unichain, the policy produced by the algorithm is still persistent. We recall that the algorithm at a given iteration visits each state θ of the space and selects the value of the decision vector (the policy in θ) associated with θ as in Equ. (5.10). Thus, in each state, θ selects the value of the decision vector $\underline{u}'$ minimizing

$$\tilde{\delta}'_{\underline{u}'}(\theta)+\sum_{\theta^*\in\mathcal{X}}\zeta_{\underline{u}'}(\theta,\theta^*)h'_k(\theta^*), \tag{5.37}$$

where the $h'_k(\theta)$ are the so called *differential costs* associated with μ'_k. Since μ'_k is unichain, we can select a recurrent state θ^r[15] of the associated chain and define

$$h'_k(\theta^r)=0;\;\; h'_k(\theta^s)=\mathcal{W}^{\mu'_k}(\theta^s,\phi^r)-\mathcal{C}^*H_{\theta^s}(\theta^r),\; \hat{\theta}\neq\theta^r, \tag{5.38}$$

where $\mathcal{W}^{\mu'_k}(\theta^s,\theta^r)$ is the average total cost incurred by the process under policy μ'_k in a sample-path starting in θ^s and finishing in θ_r and $H_{\theta^s}(\theta^r)$ is the average length in frames of the sample-path. Thus, the differential costs represent the difference between the average total cost to return in θ^s from θ^r and the average cost incurred if the cost in each frame is equal to λ^k. Note that, since θ^r is recurrent and the costs are bounded, $H_{\theta^s}(\theta^r)$ and $\mathcal{W}^{\mu'_k}(\theta^s,\theta^r)$ are finite. With this definition, it is clear the algorithm selects, in each state, the control vector minimizing the residual average cost to return in θ^r from the current state. If in each state $\theta\notin\mathcal{E}$, the algorithm selects a control vector $\mu'_{k+1}(\theta)=\underline{u}\neq 0^{|\mathcal{S}'|}$ then μ'_{k+1} is a persistent policy. Note that μ'_k is persistent, and thus the differential costs $h'_k(\theta)$ are associated with a persistent policy. With a system of recursive equations analogous to that presented in App. 5.10, it is possible to show that the algorithm does not select the control vector $\underline{u}=0^{|\mathcal{S}'|}$ in any $\theta\notin\mathcal{E}$. However, similarly to the discussion in App. 5.10, a policy forcing all sources to idleness essentially adds a transition incurring the maximum immediate cost and connects to states whose paths to return to the reference states can not overcome this gap.

[15] Here we assume $\theta^r\notin E$, *i.e.*, at least one source has a non-empty buffer.

References

[1] J. Mitola, "Cognitive radio: an integrated agent architecture for software-defined radio," Doctor of Technology, Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.

[2] S. Haykin, "Cognitive radio: brain–empowered wireless communications," *IEEE J. Select. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[3] Q. Zhao and L. Tong and A. Swami and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a POMPD framework," *IEEE J. Select. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.

[4] O. Simeone and Y. Bar-Ness and U. Spagnolini, "Stable throughput of cognitive radios with and without relaying capability," *IEEE Trans. Wireless Commun.*, vol. 55, no. 12, pp. 2351–2360, Dec. 2007.

[5] S. Geirhofer and L. Tong and B. M. Sadler, "Dynamic Spectrum access in the time domain: modeling and exploiting white space," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 66–87, May 2007.

[6] R. Urgaonkar and M. J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," in *in Proc. of the 27th IEEE Conference on Computer Communications (IEEE INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 1301–1309.

[7] W. Zhang and U. Mitra, "A spectrum-shaping perspective on cognitive radio: uncoded primary transmission case," in *IEEE ISIT*, Toronto, Ontario, Canada, Jul. 2008.

[8] L. Cao and H. Zheng, "Distributed rule–regulated spectrum sharing," *IEEE J. Select. Areas Commun.*, vol. 26, no. 1, pp. 130–145, Jan. 2008.

[9] H. Su and X. Zhang, "Cross–layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks," *IEEE J. Select. Areas Commun.*, vol. 26, no. 1, pp. 118–129, Jan. 2008.

[10] Y. Xing and C. N. Mathur and M. A. Haleem and R. Chandramouli and K. P. Subbalakshmi, "Dynamic spectrum access with QoS and interference temperature constraints," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 423–433, Apr. 2007.

[11] L. Tassiulas, "Adaptive back-pressure congestion control based on local information," *IEEE Trans. Automat. Contr.*, vol. 40, no. 2, pp. 236–250, Feb. 1995.

[12] W. Chen and M. J. Neely and U. Mitra, "Energy–efficient transmissions with individual packet delay constraints," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2090–2109, May 2008.

[13] K. Sundaresan and R. Sivakumar and M. A. 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] P. Charporkar and S. Sarkar, "Minimizing delay in loss-tolerant MAC layer multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4701–4713, Oct. 2006.

[15] A. Stamoulis and N. D. Sidiropoulus and G. B. Giannakis, "Time-varying fair queueing scheduling for multicode CDMA based on dynamic programming," *IEEE Trans. Wireless Commun.*, vol. 3, no. 2, pp. 512–523, Mar. 2004.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2001, vol. 2.

[17] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. London: Springer-Verlag, 1993.

[18] E. Altman, "Constrained Markov Decision Processes," in *Chapman & Hall CRC*, Boca Raton, FL, USA, 1998.

[19] H. S. Chang, "A policy improvement method in constrained stochastic dynamic programming," *IEEE Trans. Automat. Contr.*, vol. 51, no. 9, pp. 1523–1526, Sep. 2006.

[20] K. Lai and J. J.Shynk, "Analysis of the Linear SIC for DS/CDMA Signals with random Spreading," *IEEE Trans. Signal Processing*, vol. 52, no. 12, pp. 3417–3428, Dec. 2004.

Chapter 6

Recursive Analysis of Ad Hoc Networks with Packet Queueing, Channel Contention and Hybrid ARQ

6.1 Introduction

The analytical modeling of wireless networks generally requires high computational complexity due to the need to account for the overall network status. In particular, when considering packet arrivals, queueing, channel access and error control, the number of states may quickly become unmanageable even for just a few nodes. Another approach is to assume network saturation [1,2], which however does not always provide accurate results.

In this chapter we propose a recursive analytical model that makes it possible to accurately evaluate fundamental network performance metrics, such as throughput, channel occupancy, number of packets in the queue and service time, with low computational complexity even when considering many nodes and long queues. Our model, through semi-Markov and renewal theory [3], recursively computes a compact representation of the channel status. We analyze a wireless network where nodes contend for the channel to deliver their packets to the intended destination. We consider two medium access control protocols, namely collision access (CA) and perfect random access (PRA). In CA, if multiple nodes access the channel in the same slot a collision occurs, no packets are received, and every node involved enters backoff. In PRA, we make the idealized assumption that at any given time there is only one transmitter, randomly chosen among the set of contending nodes.

Errors are recovered via a Hybrid Automatic Retransmission reQuest (HARQ) mechanism, whereas failed retransmissions and unsuccessful channel access attempts are handled by a random backoff mechanism. We demonstrate the accuracy of the proposed analysis, by comparing the performance predicted by our model to that obtained through extensive simulations, showing that the proposed analytical technique is a powerful and effective tool for the study of complex networks.

This work focuses on collision avoidance-based ad hoc networks. However, we are currently addressing simultaneous access networks and we will include the extension of the following analytical model to multiple simultaneous access networks. The work presented in this chapter represents, in our opinion, an important improvement of this kind of analytical tools, and will enable the investigation of fundamental issues.

The reference paper for this work is [C15ml] (see Appendix B).

The rest of this chapter is organized as follows. In Section 6.2, we describe the model of the system under investigation. Sections 6.3 and 6.4 present the analytical of the network. In Section 6.5, we show results assessing the accuracy of our tool.

6.2 System Model

We consider a network of N nodes. Each node has a First-In First-Out (FIFO) finite queue, where at most N_Q packets of fixed size can be stored. Packet arrivals in the queue of each node are modeled as a Poisson process of intensity λ [pkt/s] per node. Packets generated by a node with full queue are discarded. We assume that time is slotted and that nodes are slot-synchronous.

Before transmission, packets are encoded with an error correction code, and the obtained codeword is split into frames that fit the size of a slot. Nodes have a maximum number of T independent attempts to successfully deliver the packet to their intended destination. After T attempts the packet is removed from the queue even though it has not been successfully delivered. At each attempt, the source transmits at most K frames, each followed by a feedback packet from the destination, that reports the outcome of packet decoding. Each attempt is stopped either after a positive acknowledgment (ACK) or after K failed frame transmissions. We assume that after the t-th failed transmission attempt, a node has to wait for a backoff period w_t uniformly distributed in the backoff window W_t, with $E[w_t] = \bar{W}_t = W_t/2$, before being allowed to contend for channel access again.

We assume that successive frames in the same attempt provide incremental redundancy and are decoded together. If correct decoding is possible, the packet is correctly received and the node becomes idle or tries again to access the channel according to the state of its queue. If decoding fails, a further attempt is scheduled after a random backoff. Subsequent attempts are independent, i.e., the receiver keeps no memory about past attempts. Let ρ_j be the probability that the receiver correctly decodes the packet upon reception of the j–th frame in a given attempt. Note that ρ_j does not depend on the particular source-destination pair, since we assume that all links have the same channel statistics, nor on the attempt index, since the receiver discards previously received frames at the end of each attempt and starts again from scratch. Note that this model also includes the transmission of a set of h frames with a single feedback from the destination (pure FEC) by simply setting $\rho_j = 0$ for $j = 1, \ldots, h-1$ and ρ_h equal to the overall success probability.

As previously stated, we consider PRA and CA protocols. In both cases, we assume that

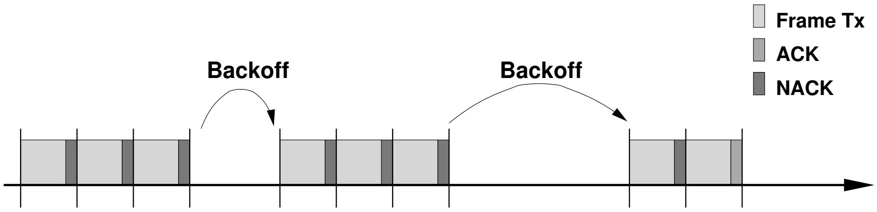


Figure 6.1. *Example of packet service.*

after a node has gained access to the channel, it has priority over all other nodes during all frame transmissions in that transmission attempt.

6.3 Analysis of the PRA protocol

In this section we derive the average channel occupancy, throughput, service time and number of packets in the queue for PRA. The analysis is based on semi-Markov theory and works in a recursive fashion. Note that the system under investigation could be modeled with a conventional semi-Markov process, but the number of states of the embedded chain required to fully describe the system is proportional to $N \times N_Q \times N_s$ and may become very large even for relatively small values of the number of nodes N, queue length N_Q and service states N_s. Our recursive model, instead, allows for an accurate estimation of the metrics listed above with affordable complexity even for high values of the system parameters.

The proposed analytical model works recursively accumulating the channel status in a simple Markov chain $\mathcal{C}$, referred to as channel chain in the following. Chain $\mathcal{C}$ is used to both compute the performance metrics and characterize the behavior of the whole network. At each step the iterative algorithm works as follows

- taking as input a channel chain $\mathcal{C}$, a chain $\mathcal{S}$ describing the service of a single packet of a *reference node*[1] is derived;
- from the service chain $\mathcal{S}$, the probabilities characterizing the behavior of a single node are computed;
- a new $\mathcal{C}$ is computed.

In the following we denote the average transition probability from state ω_1 to ω_2 of a given chain $\mathcal{X}$ as $p_{\mathcal{X}}(\omega_1, \omega_2)$. Furthermore, $d_{\mathcal{X}}(\omega_1, \omega_2)$ and $o_{\mathcal{X}}(\omega_1, \omega_2)$ denote the average number of slots associated with this transition (delay) and the number of slots in which a source transmits a frame, respectively. The steady–state probability that chain $\mathcal{X}$ is in a state ω is $\pi_{\mathcal{X}}(\omega)$.

For PRA, the channel chain $\mathcal{C}$ has $K+1$ states. State 0 represents the idle channel state (i.e., none of the nodes considered for the derivation of $\mathcal{C}$ is accessing the channel), while

[1] With packet service we refer to the whole packet delivery process, from the first attempt to successful decoding or discarding.

state $i \in \{1, \ldots, K\}$ is associated with the transmission of the i–th frame of an attempt. Each transition has duration equal to one slot.

6.3.1 Service Chain

In this section we construct the semi-Markov *service chain* $\mathcal{S}$, which describes the evolution of a packet service for the reference node, given a channel chain $\mathcal{C}$. The chain has an initial state, *s*, and an absorbing state *f*, that corresponds to either successful transmission of a packet or its dropping due to too many failed attempts. The other states are associated with the various access attempts and backoff intervals. In particular, state a_i^t corresponds to the event that the reference user wants to access the channel in attempt t and finds the channel chain in state i, whereas state b_i^t corresponds to the event that the user enters the backoff state following a transmission or access failure when in state a_i^t.

Starting from state *s*, the chain moves to state a_i^1, associated with the first access attempt, with transition probability $p_{\mathcal{S}}(s, a_i^1) = \pi_{\mathcal{C}}(i)$. This is because the source randomly samples the underlying channel process described by chain $\mathcal{C}$. In addition, we have $o_{\mathcal{S}}(s, a_i^1) = d_{\mathcal{S}}(s, a_i^1) = 0$.

The average number of slots associated with a successful and a failed attempt are $\tau_s = \sum_{k=1}^{K} k\rho_k$ and $\tau_f = K$, respectively. The average success probability of an attempt is $\rho = \sum_{k=1}^{K} \rho_k$. When the channel chain is in the idle state, the channel is assigned to the current node with probability one (no other nodes contend), and the service process moves to the associated backoff state only in case of failure. In summary,

$$\begin{aligned} p_{\mathcal{S}}(a_0^t, f) &= \rho \\ p_{\mathcal{S}}(a_0^t, b_0^t) &= 1 - \rho \\ d_{\mathcal{S}}(a_0^t, f) &= \tau_s \\ d_{\mathcal{S}}(a_0^t, b_0^t) &= K \end{aligned} \tag{6.1}$$

If the channel chain is in state 1, the reference node has to contend for the channel with other nodes in the system. We denote with ψ the probability that the reference node gains access to the channel, i.e., it wins the contention. The process moves to the final state if the node wins the contention and the attempt is successful, with probability $\psi\rho$. On the other hand, the process moves to a backoff state if either the channel is not assigned to the node (w.p. $1 - \psi$) or if the node is assigned the channel but fails to deliver the packet (w.p. $\psi(1 - \rho)$), i.e.,

$$\begin{aligned} p_{\mathcal{S}}(a_1^t, f) &= \psi\rho \\ p_{\mathcal{S}}(a_1^t, b_1^t) &= 1 - \psi + (1 - \rho)\psi = 1 - \psi\rho \\ d_{\mathcal{S}}(a_1^t, f) &= \tau_s \\ d_{\mathcal{S}}(a_1^t, b_1^t) &= (1 - \rho)\psi K/(1 - \psi\rho) \end{aligned} \tag{6.2}$$

For these transitions we have $o_{\mathcal{S}}(\omega, \omega') = d_{\mathcal{S}}(\omega, \omega')$.

When the channel chain is in states associated with the transmission of a frame with index greater than one, there is already a user who won the contention and has access priority over all others, so that the current node cannot gain access and the service process moves to the associated backoff state with probability one, i.e.

$$\begin{aligned} p_{\mathcal{S}}(a_i^t, b_i^t) &= 1, \text{ for } i=2,\dots,K \\ d_{\mathcal{S}}(a_i^t, b_i^t) &= 0, \text{ for } i=2,\dots,K \end{aligned} \tag{6.3}$$

The probability that chain $\mathcal{C}$ moves from state i to state j during the t–th backoff period is computed as

$$\phi^t_{\{i,j\}} = \sum_{w_t=1}^{W_t} \frac{1}{W_t} p_{\mathcal{C}}^{(w_t+1)}(i,j)\,, \tag{6.4}$$

where $p_{\mathcal{C}}^{(w_t+1)}(i,j)$ is the transition probability of chain $\mathcal{C}$ from state i to state j in w_t+1 slots and $\frac{1}{W_t}$ is the probability of having a backoff interval with duration $w_t \in [1, W_t]$ after the t–th transmission attempt. From the backoff states b_i^t, with $t<T$, the transition probabilities are

$$p_{\mathcal{S}}(b_i^t, a_j^{t+1}) = \begin{cases} \phi^t_{\{i,j\}} & \text{if } i \neq 1 \\ \frac{(1-\rho)\psi\phi^t_{\{0,j\}}+(1-\psi)\phi^t_{\{1,j\}}}{(1-\rho)\psi+(1-\psi)} & \text{if } i = 1, \end{cases} \tag{6.5}$$

while for $t=T$ we have $p_{\mathcal{S}}(b_i^t,f)=1$. Thus, when channel chain is in state 1, if the node is assigned the channel (probability ψ) and it fails to deliver the packet (probability $1-\rho$), the channel chain restarts from the idle state, as other nodes keep idle during the transmission of the reference node. Note that in the transition from state b_1^t to state a_j^{t+1} the channel chain is assumed to be in idle state 0 if the reference node wins the contention to access the channel. For all these transitions we have $d_{\mathcal{S}}(b_i^t, a_j^{t+1}) = \bar{W}_t$ and $o_{\mathcal{S}}(b_i^t, a_j^{t+1}) = 0$.

Finally, from state *f* we have: $p_{\mathcal{S}}(f,f)=1$ and $o_{\mathcal{S}}(f,f)=d_{\mathcal{S}}(f,f)=0$.

We now derive the average service time Δ, i.e., the average number of slots it takes the service process to reach state *f* from state *s*. Due to the structure of the service chain, we can reduce the complexity of the computation of Δ. We map states $\omega \in \mathcal{S}$ to $0,1,\dots,|\mathcal{S}|-1$with the following rule

$$m_\omega = \begin{cases} 0 & \text{if } \omega = s \\ 2(K+1)(t-1)+i & \text{if } \omega = a_i^t \\ 2(K+1)(t-1)+K+i+1 & \text{if } \omega = b_i^t \\ |\mathcal{S}|-1 & \text{if } \omega = f. \end{cases}$$

where $i=0,\dots,K$ and $t=1,\dots,T$. We define matrices $\boldsymbol{P}$, $\boldsymbol{D}$, $\boldsymbol{O}$, whose elements are the probability, average delay and number of transmissions of chain $\mathcal{C}$ with states mapped by m_ω. We define the vector $\boldsymbol{T}$, whose m_ω–th element $[\boldsymbol{T}]_{m_\omega}$ is the average number of slots it takes the service process to reach the absorbing state starting from state ω.

We compute the elements of the vector iteratively, starting from the last state and moving step-by-step to the initial state *s*. In particular, we denote with $\boldsymbol{T}(h)$ the vector at the h–th iteration, where $\boldsymbol{T}(0)$ is a vector with all elements set to zero. Then we move backward to the previous state of the map and we update the element $[\boldsymbol{T}(1)]_{|\mathcal{S}|-1}$ weighing the number of slot associated to transitions from this state with the transition probabilities, i.e.,

$$[\boldsymbol{T}(1)]_{|\mathcal{S}|-1} = (\boldsymbol{T}(0) + \boldsymbol{D}(1,:)) \times \boldsymbol{P}(1,:)^T,$$

where we denote as $\boldsymbol{X}(i,:)$ the i–th row of matrix $\boldsymbol{X}$ and T is the transpose operator. In general, at the h–th iteration, $h = 1, \ldots, \mid \mathcal{S} \mid$ we have

$$\begin{aligned} [\boldsymbol{T}(h)]_{|\mathcal{S}|-h} &= (\boldsymbol{T}(h-1) + \boldsymbol{D}(h,:)) \times \boldsymbol{P}(h,:)^T \\ [\boldsymbol{T}(h)]_j &= [\boldsymbol{T}(h-1)]_j \text{ if } j > \mid \mathcal{S} \mid -h \end{aligned}$$

The average service time is computed as $\Delta = [\boldsymbol{T}(\mid \mathcal{S} \mid)]_0$. In fact, the time before absorption is evaluated from the last reachable state, moving toward the starting state s, and at each iteration the delay of transitions from a given state i is summed to the delay associated with the states it can reach, that are in the set of the states with index $j > i$ and weighed with the transition probability. The average number of slots Γ in which the reference node transmits during the service time can be evaluated analogously, by just replacing $\boldsymbol{D}(h,:)$ with $\boldsymbol{O}(h,:)$ in the previous equations.

We now compute the probability α_t that given that a node transmits or contends for the channel, it is in the t–th attempt. To do so, we consider a slightly modified version of the service chain, called $\mathcal{S}'$, in which we change the transition probabilities from state f in order to create a loop, i.e., $p_{\mathcal{S}'}(f,s) = 1$, $o_{\mathcal{S}'}(f,s) = d_{\mathcal{S}'}(f,s) = 0$. Then, α_t is obtained from the steady–state distribution of the modified chain, $\pi_{\mathcal{S}'}$, by summing the probabilities of all the transmission states associated to the t–th attempt, normalized by the probability of being in a transmission state, i.e.,

$$\alpha_t = \frac{\sum_i \pi_{\mathcal{S}'}(a_i^t)}{\sum_i \sum_z \pi_{\mathcal{S}'}(a_i^z)}, \tag{6.6}$$

Similarly, the probability that given that the node enters a backoff period, this is associated with the t–th attempt is

$$\beta_t = \frac{\sum_i \pi_{\mathcal{S}'}(b_i^t)}{\sum_i \sum_z \pi_{\mathcal{S}'}(b_i^z)}. \tag{6.7}$$

6.3.2 Idle Node Contention Probability

In this section we discuss how α_t, β_t and Δ are used to derive the probability ν that a non transmitting node will contend for the channel in the next slot. With the aim of describing the queue status of the node, we construct the queue chain $\mathcal{Q}$. This chain has states $0, \ldots, N_Q$. Let $r(k)$ be the probability that k packets arrive during time Δ, i.e., $r(k) = (\lambda\Delta)^k e^{-\lambda\Delta}/k!$[2]. Moving from state $q > 0$, a packet is removed from the queue, and thus the

[2] As an approximation, we assume that the duration of each service time is deterministic and equal to the average duration Δ

transition probabilities of $\mathcal{Q}$ are

$$p_{\mathcal{Q}}(i,j) = \begin{cases} r(j-i+1), & j=i-1,\dots,N_Q-1 \\ \sum_{h=N_Q}^{+\infty} r(h-i+1), & j=N_Q \end{cases}$$

for $i=1,\dots,N_Q$, whereas

$$p_{\mathcal{Q}}(0,j) = \begin{cases} r(j), & j=0,\dots,N_Q-1 \\ \sum_{h=N_Q}^{+\infty} r(h), & j=N_Q \end{cases}$$

We denote with $\pi_{\mathcal{Q}}$ the steady state distribution of chain $\mathcal{Q}$. Note that at the end of each packet service a packet is always removed from the queue because it is either delivered or discarded. We also define a reduced chain $\mathcal{Q}'$, with states $\{0,1\}$, where 0 corresponds to the empty queue and 1 to the non-empty queue node status[3].

The transition probabilities are

$$\begin{aligned} p_{\mathcal{Q}'}(0,1) &= \sum_{h=1}^{N_Q} p_{\mathcal{Q}}(0,h) \\ p_{\mathcal{Q}'}(1,0) &= \sum_{h=1}^{N_Q} \pi_{\mathcal{Q}}(h) p_{\mathcal{Q}}(h,0) / \sum_{u=1}^{N_Q} \pi_{\mathcal{Q}}(u) \end{aligned} \tag{6.8}$$

A node can be idle in a slot due to either empty queue or backoff. Thus, we compute the probability that a node with empty queue will contend for the channel in the next slot and the analogous probability for a backoff slot, referred to as ϵ and σ, respectively. ϵ can be simply evaluated as the probability that an idle node with an empty queue has a packet to serve in the next slot, i.e., $\epsilon = p_{\mathcal{Q}'}(0,1)$. The evaluation of σ is more involved. We use the previously computed probabilities, β_t, that given that a node is in backoff it is in the backoff period associated with the t–th attempt, to derive the probability β_t^* that given that we select a backoff slot this is in the backoff period associated with the t–th attempt. We obtain

$$\beta_t^* = \beta_t \bar{W}_t \Bigg/ \left(\sum_{i=1}^{T} \beta_i \bar{W}_i \right) \tag{6.9}$$

This distribution is slightly different from β_t, since we must account for the increased probability of selecting a longer backoff period when a slot is picked at random. Then, we compute the probability that given that the randomly selected slot belongs to the backoff of the t–th attempt it is the last slot, and thus it may be available for channel contention, averaging this probability over the set of the suitable lengths of the period. We denote this probability with ℓ_t and we get

$$\ell_t = \frac{\frac{1}{W_t}\sum_{w_t=1}^{W_t} w_t \frac{1}{w_t}}{\frac{1}{W_t}\sum_{w_t=1}^{W_t} w_t} = \frac{W_t}{\frac{W_t(W_t+1)}{2}} = \frac{2}{W_t+1}. \tag{6.10}$$

[3] Chain $\mathcal{Q}'$ does not conserve the Markov property in general, so that considering this reduced chain introduces another approximation.

Note that also in this case, the probability of being in a longer backoff period is increased due to the random selection. In particular, at the numerator, w_t is the duration of the backoff and $1/w_t$ is the probability of picking the last slot of a backoff interval of w_t slots, while $1/W_t$ is the probability for the backoff period to be of w_t slots, due to uniform distribution assumption.

To obtain σ the probabilities ℓ_t must be weighed with the distribution β_t^*. While in backoff intervals associated with attempts $1 \leq t < T$ after the last slot of the backoff the node always contends for the channel, as a retransmission of the packet is scheduled, if the process is in the backoff interval associated with the last allowed attempt, then it contends for the channel only if the node's queue is not empty[4]. In particular we obtain

$$\sigma = \sum_{i=1}^{T-1} \beta_t^* \ell_t + \beta_T^* \ell_T p_{Q'}(1,1). \tag{6.11}$$

Finally, we obtain ν weighing σ and ϵ with the probability that a randomly selected idle slot is caused by empty queue or backoff, respectively. These probabilities are obtained from the reduced queue chain steady–state probabilities and the average fraction of service slots in which the node transmits. In particular ν is as follows

$$\nu = x_e \epsilon + x_s \sigma, \tag{6.12}$$

where weights x_e and x_s account for the probability of being in an idle slot during an empty queue and service period, respectively. More specifically, we get

$$x_s = \frac{\pi_{Q'}(1)(1-\Gamma)}{\pi_{Q'}(1)(1-\Gamma) + \pi_{Q'}(0)}, \tag{6.13}$$

and $x_e = 1 - x_s$.

6.3.3 Recursive Algorithm

The previously presented analysis assumes that the assignment probability ψ and the channel chain $\mathcal{C}$ are known. In this section we provide a technique to recursively compute the parameters for an arbitrary number of nodes N and a given value of the packet arrival intensity λ. In the following we denote as $\mathcal{C}_n$ the channel chain modeling the behavior of n nodes.

The channel chain $\mathcal{C}_n$, given the previously derived probabilities and quantities, has tran-

[4] This effect is slightly approximated in the following equation, since we do not take into account that $p_{Q'}(1,1)$ includes the service end due to success at the previous attempts. This choice is motivated by the wish to consider the packet service as a whole.

sition probabilities

$$\begin{aligned} p_{\mathcal{C}_n}(0,1) &= u_n, \\ p_{\mathcal{C}_n}(0,0) &= 1-u_n \\ p_{\mathcal{C}_n}(i,i+1) &= 1-\rho_i, \\ p_{\mathcal{C}_n}(i,1) &= \rho_i(p_{\mathcal{Q}'}(1,1)+p_{\mathcal{Q}'}(1,0)u_{n-1}), \\ p_{\mathcal{C}_n}(i,0) &= 1-p_{\mathcal{C}_n}(i,0)-p_{\mathcal{C}_n}(i,i+1), \\ p_{\mathcal{C}_n}(K,1) &= \rho_K p_{\mathcal{Q}'}(1,1)+(1-\rho_K p_{\mathcal{Q}'}(1,1))u_{n-1} \\ p_{\mathcal{C}_n}(K,0) &= 1-p_{\mathcal{C}_n}(K,1), \end{aligned} \tag{6.14}$$

with $i=1,\ldots,K-1$ and $u_n=1-(1-\nu)^n$. Thus, if the channel is in state 0, it moves to 1 if at least one node contends for the channel, otherwise it stays in the idle state. From states associated to the i–th frame transmission with $1\le i<K$, the process moves to the next frame transmission state if a failure occurs. If the current transmission achieves a success and at least one of the n nodes contend for the channel, then the process moves to the state associated with the transmission of the first frame. This is due to the capability of the PRA protocol to assign the channel to one user even in the case where multiple nodes contend for the channel. On the contrary, if a success is achieved and all nodes stay in the idle state, then the process moves to state 0. The last allowed frame transmission state is slightly different, since the current transmission is dismissed regardless of whether or not a success occurs. However, the outcome of the current transmission affects the number of the nodes that are not going to be idle with probability one in the next slot. In fact, if a failure occurs, the currently transmitting node enters backoff for at least one slot in all cases, while if it achieves a success it can immediately contend for the channel if it has a packet in its queue.

The probability that the channel is assigned to the reference node, given that the channel chain $\mathcal{C}_n$ is in state 1, is obtained by computing the probability that a given number of users contend for the channel when the channel chain is in state 1. Let us first define the distribution of the number of users contending for the channel, when m is the maximum number of users available for contention (i.e., not in backoff), as

$$\gamma_m(z,p)=\begin{cases}\binom{m}{m-z}p^z(1-p)^{m-z} & \text{if } z\le m\\ 0 & \text{otherwise.}\end{cases} \tag{6.15}$$

The channel evolution is not Markov in general. In fact, the current channel status is not sufficient to derive transition probabilities. Using the channel chain $\mathcal{C}$ we are approximating the behavior of the system. However, to construct the next chain, through reference node characterization and fundamental probabilities derivation, we can look at the previous channel chain short term evolution preserving the simplicity of the equations. For instance, to derive the probability that a node gains access to the channel when the channel chain is in status 1, we can average over the state of the channel chain in the previous slot, in order to increase the accuracy of the number of contending nodes. In fact, the number of nodes available for contention depends on the chain state from which the process enters state 1. The probability that the process enters state 1 from state i, denoted as ξ_i, is

$$\xi_i = \frac{\pi_{\mathcal{C}_n}(i) p_{\mathcal{C}_n}(i,1)}{\sum_{j=0}^{K} \pi_{\mathcal{C}_n}(j) p_{\mathcal{C}_n}(j,1)} \tag{6.16}$$

The assignment probability for the reference node, when the total number of nodes is n, is then

$$\psi = \sum_{j=0}^{K} \xi_j \psi_j, \tag{6.17}$$

where ψ_j is the assignment probability given that the process was in state j in the previous slot. For the idle state 0 we have

$$\psi_0 = \frac{\sum_{z=1}^{n-1} \frac{1}{z+1} \gamma_{n-1}(z,\nu)}{\sum_{z=1}^{n-1} \gamma_{n-1}(z,\nu)}. \tag{6.18}$$

In fact, given that the reference node and z further nodes are contending for the channel, the probability that a given node is assigned the channel is $1/(z+1)$. Note that, since the channel chain is in state 1, at least one node contends for the channel. The assignment probabilities associated with the other channel chain states are more involved. We distinguish between the last and the other frame transmissions. In fact, while in the case the process moves from state i to state 1, with $1 \leq i < K$, the end of service is necessarily due to successful packet decoding, if the process was in state K, the currently transmitting node is forced to end the attempt. This is to be taken into account to correctly compute the probability of having z nodes contending for the channel, due to the backoff status forced by failure that decreases by one the number of nodes available for contention. The expressions of ψ_K and ψ_i, with $1 \leq i < K$ are reported in (6.19) and (6.20), respectively.

$$\psi_K = \frac{\sum_{z=1}^{N-2} \frac{1}{z+1} [p_{Q'(1,1)}\ \rho_K\ \gamma_{N-1}(z,\nu) + (1 - p_{Q'(1,1)}\ \rho_K)\ \gamma_{N-2}(z,\nu)] + \frac{1}{N}\ \gamma_{N-1}(N-1,\nu)\ p_{Q'(1,1)} \rho_K}{\sum_{z=1}^{N-2} [p_{Q'(1,1)}\ \rho_K\ \gamma_{N-1}(z,\nu) + (1 - p_{Q'(1,1)}\ \rho_K)\ \gamma_{N-2}(z,\nu)] + p_{Q'(1,1)}\ \rho_K\ \gamma_{N-1}(N-1,\nu)} \tag{6.19}$$

$$\psi_i = \frac{\sum_{z=1}^{N-2} \frac{1}{z+1} [p_{Q'(1,1)} \gamma_{N-1}(z,\nu) + p_{Q'(1,0)}\ \gamma_{N-2}(z,\nu)] + \frac{1}{N}\ p_{Q'(1,1)}\ \gamma_{N-1}(N-1,\nu)}{\sum_{z=1}^{N-2} [p_{Q'(1,1)}\ \gamma_{N-1}(z,\nu) + p_{Q'(1,0)}\ \gamma_{N-2}(z,\nu)] + p_{Q'(1,1)}\ \gamma_{N-1}(N-1,\nu)}, 0 < i < K \tag{6.20}$$

In the case of (6.19), if a success is achieved and the node has a packet in the queue, all sources are available for contention, while otherwise the previously transmitting node must be assumed in backoff. In the case of (6.20) the transition from state i occurs only if the previous transmission achieves a success. Thus, the previously transmitting node does not contend for the channel only if it has an empty queue. In both cases the probability has to be normalized to take into account that at least one source of the channel chain contends for the channel.

Our first experimental results were based on an algorithm that iteratively updates the chain descriptions while keeping fixed the number of nodes. In particular, it starts with an initial channel chain based on single-node behavior, from which the N-node channel chain is derived. At each iteration the channel chain is updated as shown before, but the number of nodes remains unchanged. This algorithm was found to exhibit an undesirable oscillating behavior. This is especially true under heavy traffic conditions: in this case, the channel chain $\mathcal{C}_N$ at the first iteration is likely to show a high channel occupancy. Thus, the single node behavior computed with this chain may provide a high backoff probability, which corresponds to a non-aggressive node behavior. Thismakes the next aggregate channel busy probability likely to be low, which in turn results in the single source behavior evaluated by the algorithm at the next iteration to be very aggressive again, and so on.

To achieve a faster convergence, we proposed another algorithm to smooth the difference between channel chains computed in successive iterations. As briefly summarized before, each step of the recursive algorithm evaluating the channel chain $\mathcal{C}_n$ takes as input the single node channel chain $\mathcal{C}$ and works as follows. It first evaluates the chain $\mathcal{C}_{n-1}$, as in (6.14), and the assignment probability ψ. These are used to derive a new single node chain, that serves as input for the next step and for the computation of chain $\mathcal{C}_n$. Experimental results again showed that iterating on the number of nodes in the network, while improving with respect to the previous case, still led to an unsatisfactory convergence behavior for the algorithm. As a result, we further proposed to increase step by step also the per node arrival intensity λ. Thus, we start with a single source and a fraction of the per node arrival intensity λ, and we increase step-by-step their values. This algorithm was observed to exhibit much better behavior, although we have not studied in detail its convergence properties. Results comparing our analytical approach with extensive simulation of the system show that our proposal achieves a very good degree of accuracy.

6.3.4 Performance metrics

The goal of the proposed iterative analysis is to derive an accurate channel chain $\mathcal{C}$. From $\mathcal{C}$ we can compute the average fraction of time the channel is in the transmission state as $\Omega = \sum_{h=1}^{K} \pi_{\mathcal{C}}(h)$. This allows us to derive the overall network throughput as $\Theta = \Omega\rho/(\rho\tau_s + (1-\rho)\tau_f)$. From the steady–state probabilities of chain $\mathcal{Q}$, the average number of packets in the node queue can also be derived as $\Phi = \sum_{q=0}^{Q} q\pi_{\mathcal{Q}}(q)$. The service time Δ can be taken from the last iteration of the algorithm.

6.4 Analysis of the CA protocol

6.4.1 CA Protocol

In this section we discuss how the previously described analysis is to be modified for the CA protocol. The algorithm is analogous to PRA as far as the computation of the contention probability and the recursive procedure are concerned. The main differences lie in the channel and service chains. In particular, to characterize the channel, we add a further state to the channel chain $\mathcal{C}$ to distinguish between a successful channel access and a collision. Thus, we define states 1a and 1b, associated with a slot where a single node and multiple nodes try to access the channel, respectively. $\mathcal{C}$ has $K{+}2$ states, where the further states $i = 2, \ldots, K$ are associated with the i–th frame transmission of an ongoing attempt. Note that while state 1a corresponds to the start of an attempt, state 1b always results in a collision and all nodes that contended for the channel enter backoff.

For CA we do not need to evaluate the assignment probability, as multiple nodes contending for the channel always result in a collision, but the computation of the channel chain is much more involved than for PRA. In particular, we need to evaluate how many nodes are contending for the channel when the channel process is in state 1b, in order to be able to evaluate the transition probabilities from this state.

6.4.2 Service Chain

The derivation of the statistics for CA is analogous to that of PRA. Thus, in the following we define only the probabilities and the metrics associated with the service chain transitions, while referring to Section 6.3.1 for the characterization of the single-node behavior.

The service chain $\mathcal{S}$ has $2T \times (K+1) + 2$ states, including the initial state s and the absorbing state *f*. We use the same notation of Section 6.3.1. From the initial state the transition probabilities are $p_{\mathcal{S}}(s, a^1_\omega) = \pi_{\mathcal{C}}(\omega)$, with $\omega \in \mathcal{C}$. For these transitions we have $d_{\mathcal{S}}(s, a^1_\omega) = 0$ and $o_{\mathcal{S}}(s, a^1_\omega)$.

From states a^t_ω, with $1 \leq t \leq T$ and $\omega \in \mathcal{C}$ the transition probabilities and delays are

$$\begin{array}{rcl} p_{\mathcal{S}}(a^t_0, f) & = & \rho \\ p_{\mathcal{S}}(a^t_0, b^t_0) & = & 1-\rho \\ p_{\mathcal{S}}(a^t_\omega, b^t_\omega) & = & 1, \text{ for } \omega = \text{1a}, \text{1b}, 2, \ldots, K \\ d_{\mathcal{S}}(a^t_0, f) & = & \tau_s \\ d_{\mathcal{S}}(a^t_0, b^t_0) & = & K \\ d_{\mathcal{S}}(a^t_\omega, b^t_\omega) & = & 1, \text{ for } \omega = \text{1a}, \text{1b}, 2, \ldots, K \end{array} \tag{6.21}$$

Note that in this case state 1a of the channel chain also causes the reference node to enter backoff, as we have two nodes simultaneously contending for the channel. The number of frame transmissions associated with these transitions is $o_{\mathcal{S}}(\omega_1, \omega_2) = d_{\mathcal{S}}(\omega_1, \omega_2)$.

From the backoff states b^t_ω we simply have

$$\begin{array}{rcl} p_{\mathcal{S}}(b^t_{\omega_1}, a^{t+1}_{\omega_2}) & = & \phi^t_{\{\omega_1,\omega_2\}}, \text{ for } \omega_1 \neq \text{1a} \\ p_{\mathcal{S}}(b^t_{\text{1a}}, a^{t+1}_{\omega_2}) & = & \phi^t_{\{\text{1b},\omega_2\}}, \end{array} \tag{6.22}$$

as even though the channel chain is in state 1a, which allows to continue the attempt, the contention of the reference node causes a collision, that corresponds to state 1b. Moreover, the average delay associated with these transitions is always equal to the average backoff duration, i.e., $d_{\mathcal{S}}(b^t_{\omega_1}, a^{t+1}_{\omega_2}) = \bar{W}_t$, while the average number of frame transmissions is $o_{\mathcal{S}}(b^t_{\omega_1}, a^{t+1}_{\omega_2}) = 0$.

As stated before, all the probabilities concerning the single node behavior can be derived as in PRA, according to the increased number of states in the CA model and the corresponding need to adapt the mapping rule.

6.4.3 Channel Chain

The transition probabilities of the channel chain are much more involved than in PRA. More specifically, state 1b, that is the collision state, requires the evaluation of the number of nodes that caused the collision.

We have the following transition probabilities from the idle state for a chain associated with n nodes

$$\begin{aligned} p_{\mathcal{C}_n}(0,0) &= \gamma_n(0,\nu) \\ p_{\mathcal{C}_n}(0,1\text{a}) &= \gamma_n(1,\nu) \\ p_{\mathcal{C}_n}(0,1\text{b}) &= 1-\gamma_n(1,\nu)-\gamma_n(0,\nu), \end{aligned} \tag{6.23}$$

and thus the process stays in the idle state if none of the n nodes contend for the channel, while it moves to 1a if only one node contends and to 1b otherwise.

From states $\omega = 1\text{a}, 2, \ldots, K$ we get the transition probabilities

$$\begin{aligned} p_{\mathcal{C}_n}(\omega,0) &= \rho_\omega p_{\mathcal{Q}'}(1,0)\gamma_{n-1}(0,\nu), \\ p_{\mathcal{C}_n}(\omega,\omega+1) &= 1-\rho_\omega, \\ p_{\mathcal{C}_n}(\omega,1\text{a}) &= \rho_\omega[p_{\mathcal{Q}'}(1,0)\gamma_{n-1}(1,\nu)+p_{\mathcal{Q}'}(1,1)\gamma_{n-1}(0,\nu)], \\ p_{\mathcal{C}_n}(\omega,1\text{b}) &= 1-p_{\mathcal{C}_n}(\omega,1\text{a})-p_{\mathcal{C}_n}(\omega,\omega+1)-p_{\mathcal{C}_n}(\omega,0), \qquad \text{for } \omega = 1\text{a}, 2, \ldots, K-1 \\ p_{\mathcal{C}_n}(K,0) &= [\rho_K p_{\mathcal{Q}'}(1,0)+(1-\rho_K)]\gamma_{n-1}(0,\nu) \\ p_{\mathcal{C}_n}(K,1\text{a}) &= \rho_K[p_{\mathcal{Q}'}(1,0)\gamma_{n-1}(1,\nu)+p_{\mathcal{Q}'}(1,1)\gamma_{n-1}(0,\nu)]+(1-\rho_K)\gamma_{n-1}(1,\nu) \\ p_{\mathcal{C}_n}(K,1\text{a}) &= 1-p_{\mathcal{C}_n}(K,0)-p_{\mathcal{C}_n}(K,1\text{a}), \end{aligned} \tag{6.24}$$

where with a slight abuse of notation $\omega + 1$ is equal to 2 if $\omega = 1\text{a}$. For these states, the transition probabilities are similar to those described for the PRA case, splitting the transition to state 1 into two transitions to states 1a and 1b according to the number of nodes contending for the channel.

As mentioned before, the computation of the transition probabilities from state 1b is much more complicated, even though it is quite similar to the derivation of the assignment probability of the PRA case. Moreover, also in this case we average on the channel chain status at the previous slot, in order to increase the accuracy of the transition probabilities computation, without resorting to complicated models that take into account the whole channel history. In the following, with a slight abuse of notation, we refer to conditional probabilities given the state in the slot before the current one, meaning that we average with respect to the past channel evolution[5].

[5] We are effectively computing the one-step transition probabilities of the (non-Markov) process by explicitly

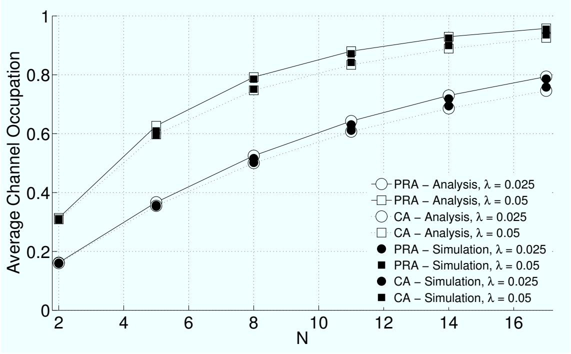


Figure 6.2. *Average channel occupancy as a function of the node number N for various values of the per node arrival rate λ and the access protocol.*

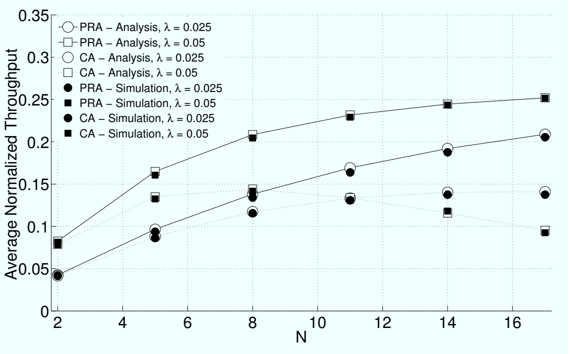


Figure 6.3. *Average throughput as a function of the node number N for various values of the per node arrival rate λ and the access protocol.*

We have first to compute the conditional transition probability $p_{\mathcal{C}_n}(1\text{b}, \omega_2 \mid \omega_0)$, i.e., the probability that the process moves from state 1b to state ω_2 given that it was in state ω_0 before entering state 1b[6].

Suppose the process moves from the idle state to state 1b, then there are $2 \le m \le n$ nodes contending for the channel. These m nodes enter backoff, and thus can not contend for the channel in the next transition. For $\omega_0 = 0$ we have

$$\begin{aligned}
p_{\mathcal{C}_n}(1\text{b}, 0 \mid 0) &= (\sum_{u=2}^{n} \gamma_n(u,\nu)\gamma_{n-u}(0,\nu))/(\sum_{u=2}^{n} \gamma_n(u,\nu)) \\
p_{\mathcal{C}_n}(1\text{b}, 1\text{a} \mid 0) &= (\sum_{u=2}^{n} \gamma_n(u,\nu)\gamma_{n-u}(1,\nu))/(\sum_{u=2}^{n} \gamma_n(u,\nu)) \\
p_{\mathcal{C}_n}(1\text{b}, 1\text{b} \mid 0) &= (\sum_{u=2}^{n} \gamma_n(u,\nu)(1-\gamma_{n-u}(1,\nu) - \gamma_{n-u}(0,\nu)))/(\sum_{u=2}^{n} \gamma_n(u,\nu)),
\end{aligned} \tag{6.26}$$

These probabilities can be easily understood keeping in mind that only the $n-m$ nodes that do not contend for the channel in the current slot are able to contend for the channel in the next slot. We also have $p_{\mathcal{C}_n}(1\text{b}, \omega_2 \,|\, 0) = p_{\mathcal{C}_n}(1\text{b}, \omega_2 \,|\, 1\text{b})$, since nodes that cause collision enter the same idle status of nodes during idle slots.

The transition probabilities with $\omega_0 = 1\text{a}, 2, \ldots, K-1$ are

$$\begin{aligned}
p_{\mathcal{C}_n}(1\text{b}, 0 \mid \omega_0) &= \frac{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))\gamma_{n-u}(0,\nu)+p_Q(1,1)\gamma_{n-1}(n-1,\nu)}{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))+p_Q(1,1)\gamma_{n-1}(n-1,\nu)} \\
p_{\mathcal{C}_n}(1\text{b}, 1\text{a} \mid \omega_0) &= \frac{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))\gamma_{n-u}(1,\nu)}{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))} \\
p_{\mathcal{C}_n}(1\text{b}, 1\text{b} \mid \omega_0) &= \frac{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))(1-\gamma_{n-u}(1,\nu)-\gamma_{n-u}(0,\nu))}{\sum_{u=2}^{n-1}(p_Q(1,0)\gamma_{n-1}(u,\nu)+p_Q(1,1)\gamma_{n-1}(u-1,\nu))}.
\end{aligned} \tag{6.27}$$

In these cases the contention probability of the node that was transmitting until the slot preceding state 1b must also be considered. Note that only the queue status of the nodes

accounting for one more step in the past, and then take these as the transition probabilities of an approximate Markov representation.

[6] Thus subscript denotes slot index, where slot 1 corresponds to the current slot with $\omega_1 = 1\text{b}$

is considered, since if the process moves from one of these states to state 1b a success is achieved with probability one.

If ω_0 is equal to K, we have

$$\begin{aligned}
p_{\mathcal{C}_n}(1\text{b},0\mid K) &= \frac{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))\gamma_{n-u}(0,\nu)+p_Q(1,1)\gamma_{n-1}(n-1,\nu)}{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))+p_Q(1,1)\gamma_{n-1}(n-1,\nu)}\\
p_{\mathcal{C}_n}(1\text{b},1\text{a}\mid K) &= \frac{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))\gamma_{n-u}(1,\nu))}{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))}\\
p_{\mathcal{C}_n}(1\text{b},1\text{b}\mid K) &= \frac{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))(1-\gamma_{n-u}(1,\nu)-\gamma_{n-u}(0,\nu))}{\sum_{u=2}^{n-1}(\rho_K p_Q(1,1)\gamma_{n-1}(u-1,\nu)+(1-\rho_K p_Q(1,1))\gamma_{n-1}(u,\nu))}. \qquad (6.28)
\end{aligned}$$

We have to take into account that the node that was transmitting in the previous slot is allowed to contend for the channel only if it achieves a success and has packets in its queue. The transition probabilities $p_{\mathcal{C}_n}(1\text{b},\omega_2)$ are obtained weighing the conditional probabilities that the process entered state 1b from state ω_0, i.e.,

$$p_{\mathcal{C}_n}(1\text{b},\omega_2) = \frac{\sum_{\omega_0}\pi_{\bar{\mathcal{C}}_n}(\omega_0)p_{\bar{\mathcal{C}}_n}(\omega_0,1\text{b})p_{\mathcal{C}_n}(1\text{b},\omega_2\mid\omega_0)}{\sum_{\omega_0}\pi_{\bar{\mathcal{C}}_n}(\omega_0)p_{\bar{\mathcal{C}}_n}(\omega_0,1\text{b})} \qquad (6.29)$$

where $\bar{\mathcal{C}}_n$ denotes the previously computed channel chain.

As a final remark, we observe that in CA the throughput is obtained by summing the steady–state probabilities of states $1\text{a},2,\ldots,K$, while channel occupancy is obtained by summing also $\pi_{\mathcal{C}_N}(1\text{b})$.

6.5 Results

In this section we show some example results obtained through our recursive analytical model, comparing them with simulations.

In this specific example, we set the parameters as follows: backoff window $W_t = 2^t\ B$, maximum queue length $N_Q = 10$, maximum number of retransmissions $K = 3$, $T = 3$ and $\rho_1 = 0.1$, $\rho_2 = 0.4$. We plot the results for various values of the number of nodes and packet arrival rate.

Figures 6.2, 6.3 show the average channel occupancy and the throughput as a function of the number of nodes N for two values of the per node arrival intensity λ and the two considered access schemes. It can be observed that the results obtained through our analysis show a good match with those obtained through simulation. The PRA protocol has increasing channel occupancy proportional to the throughput, while CA incurs frequent collision events when the channel is highly utilized, even though this effect may vary with the backoff policy.

Figures 6.4 and 6.5 show the average channel occupancy and the throughput as a function of the backoff parameter B for CA and various numbers of nodes with $\lambda = 0.06$[pkt/s] per node. Note that as the backoff window parameter B is increased, the channel occupancy generally decreases. Moreover, as expected, the larger the number of nodes, i.e., the overall traffic generated in the network, the bigger the fraction of time in which at least one node

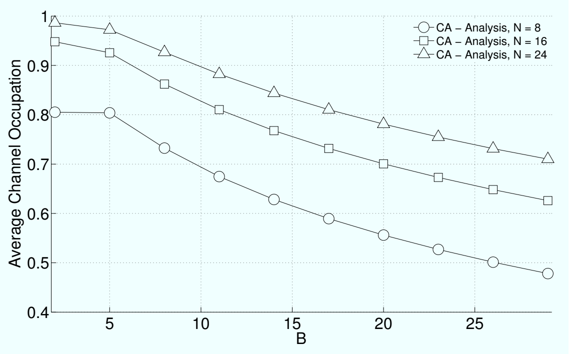


Figure 6.4. *Average Channel occupancy as a function of the backoff parameter B for various N.*

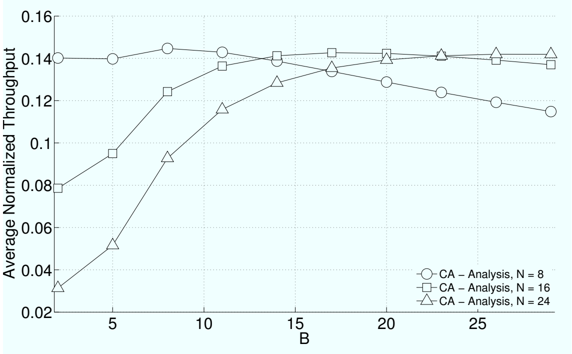


Figure 6.5. *Average throughput as a function of the backoff parameter B for various N .*

transmits. The achieved throughput results from the tradeoff between the lower collision probability, and thus the frequency with which nodes enter backoff, the backoff duration and the probability that an attempt is made. Thus, when the number of nodes is low the throughput is a decreasing function of B, since a long backoff is useless due to the low probability that a node contends for the channel. Conversely, if the number of nodes is large enough, decreasing the number of nodes contending for the channel by increasing B yields better performance, as the attempts have a higher probability of taking place with respect to small B, where collisions occur in almost every slot.

References

[1] G. Bianchi, "IEEE 802.11 - Saturation Throughput Analysis," *IEEE Communications Letters*, vol. 2, no. 12, pp. 318–320, Dec. 1998.

[2] B. J. Kwack, N. O. Song, and L. E. Miller, "Performance Analysis of Exponential Backoff," *IEEE/ACM Trans. on Networking*, vol. 13, no. 2, pp. 343–355, Apr. 2005.

[3] R. A. Howard, *Dynamic Probabilistic Systems*. New York: Wiley, 1971.

Chapter 7

Conclusions

In this thesis, we discussed several issues concerning the deployment of ad hoc networks with multiple simultaneous access. The design of the protocols for this kind of networks is clearly a difficult and complex task. We approached the problem from a wide perspective, investigating various solutions and issues.

In particular, we proposed two approaches.

We first designed a system where nodes coordinate their communications in order to let receivers to estimate the interference that will affect data reception. This enables an interference-aware resources allocation driven by receivers. In this setting, interference is controlled by a distributed access control protocol.

The second approach provides that nodes can access the channel anytime in order to disclose the fundamental mechanisms of interference in the network. We highlighted the role of error and rate control mechanisms and their complex interaction with interference and performance. We also investigated in depth techniques improving the efficiency of communications, and thus to decrease the interference load, such as cooperation and adaptive hybrid ARQ.

We also derived the optimal access and power control policy through dynamic programming. The Markov model of the network evidences how interference ties together the stochastic processes modeling individual nodes. We underscored this aspect by defining a novel interference measure based on the distortion interfering transmissions causes to individual sources' processes.

From the discussions and the results presented in this thesis it is clear that we cannot use traditional design rationales for this scenario. Both the proposed approaches grant considerable performance improvement with respect to standard systems. The best choice depends on the capabilities of the physical layer and the specific application. However, a deep understanding of interference characterization and of the mechanisms to face channel unpredictability are the base for a correct design.

Appendix A

Other Research Activities

In the following we briefly summarize some work done in these years that does not directly address the main issue of the thesis. It focuses on Markov modeling of error control schemes.

The importance of error control in multiple simultaneous access ad hoc networks has been widely underscored along the chapters presented in the main body of the thesis. Besides the, in our opinion, relevant technical contribution, we believe that the work briefly discussed in this chapter provides significant tools for an in depth understanding of fundamental issues and tradeoffs in error control.

This work has been done in collaboration with Leonardo Badia.

In [J3ml, C08ml] (see Appendix B) we present an analytical model for the study of Hybrid ARQ techniques on Discrete Time Markov Channels by means of an appropriate Markov chain, which tracks the transmission outcome and can be used to evaluate several performance metrics, including throughput, loss probability, number of retransmissions, and delay. The analysis is carried out with the assumptions that the information frame is encoded by the source with a linear block code and hard decoding is used at the receiver side. We provide numerical evaluations for the performance of a truncated Type II Hybrid ARQ technique based on Reed Solomon erasure codes.

In [J5ml, C13ml] (see Appendix B) we present a methodology to obtain a channel description tailored on performance evaluation for Incremental Redundancy Hybrid Automatic Repeat reQuest schemes. Such techniques counteract channel errors by using data coding and transmitting parts of the codeword over different channel realizations. We focus on coding performance models where the error probability is asymptotically zero if the channel parameters of these realizations fall within a given region. To map this region in a compact but still precise manner, we adopt a finite-state channel model. This approach is quite common in the literature; however, differently from existing work, we propose a novel method to derive efficient channel partitioning rules, i.e., a code-matched quantization of the channel state. Such a representation enables the use of accurate Markov models to study the system performance. Compared to existing channel representation methods, our proposed technique leads to a more accurate evaluation of higher layer statistics while at the same time keeping the computational complexity low.

We are further extending this work applying these models to relevant issues. In [C20ml] (see Appendix B) we consider a scenario where packets transmitted over the channel have different reliability and QoS constraints, and are, therefore, differently protected by the error control protocol (for instance, only packets belonging to a given class are worth of retransmission, while other can be protected only by means of packet encoding). An important application of this extension is video streaming.

We are also investigating the importance of feedback quantization in interference constrained networks. The reference paper is [C19ml] (see Appendix B).

Other recent work listed in Appendix B falls out of the scope of this discussion, as it addresses different novel issues such as error control for underwater and multimedia communications and cooperation in cellular networks.

Appendix B

Complete List of Contributions

B.1 Journal publications (accepted for publication or published)

[J1ml] 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

[J2ml] M. Levorato, S. Tomasin and M. Zorzi, "Cooperative Spatial Multiplexing for Ad Hoc Networks with Hybrid ARQ: Design and Performance Analysis," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1545–1555, Sept. 2008

[J3ml] L. Badia, M. Levorato and M. Zorzi, "Markov Analysis of Selective Repeat type II Hybrid ARQ Using Block Codes," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1434–1441, Sept. 2008

[J4ml] P. Casari, M. Levorato and M. Zorzi, "MAC/PHY Cross-Layer Design of MIMO Ad Hoc Network with Layered Multiuser Detection," *IEEE Trans. Commun.*, vol. 7, no. 11, part 2, pp. 4596–4607, Sept. 2008

[J5ml] L. Badia, M. Levorato and M. Zorzi, "A Channel Representation Method for the Study of Hybrid Retransmission-Based Error Control," accepted for publication, *IEEE Trans. Commun.*

[J6ml] S. Tomasin, M. Levorato and M. Zorzi, "Steady State Analysis of Coded Cooperative Networks with HARQ Protocols," accepted for publication, *IEEE Trans. Commun.*

[J7ml] M. Levorato and M. Zorzi, "On the Performance of Ad Hoc Networks with Multiuser Detection, Rate Control and Hybrid ARQ," accepted for publication, *IEEE Trans. Wireless Commun.*

B.2 Journal publications (Under Revision)

[J8ml] M. Levorato, Federico Librino and M. Zorzi, "Distributed Cooperative Routing and Hybrid ARQ in MIMO-BLAST Ad Hoc Networks," under revision, *IEEE Trans. Commun.*

[J9ml] L. Badia, M. Levorato, F. Librino and M. Zorzi, "Cooperation Techniques for Wireless Systems from a Network-Wide Perspective," under revision, *IEEE Wireless Comm. Mag.*

[J10ml] M. Levorato, Urbashi Mitra and M. Zorzi, "On Optimal Cognitive Access with Packet Buffering and ARQ," *in preparation*

B.3 Publications in international conferences

[C01ml] M. Rossi, P. Casari, M. Levorato and M. Zorzi, "Multicast Streaming over 3G Cellular Networks through Multi-Channel Transmissions: Proposals and Performance Evaluation," *IEEE WCNC, New Orleans, LA, Mar. 13-17, 2005*

[C02ml] P. Casari, M. Levorato and M. Zorzi, "On the Implications of Layered Space-Time Multiuser Detection on the Design of MAC Protocols for Ad Hoc Networks," *IEEE PIMRC, Berlin, Germany, Sept. 11-14, 2005*

[C03ml] P. Casari, M. Levorato and M. Zorzi, "Some Issues Concerning MAC Design in Ad Hoc Networks with MIMO Communications," *WPMC, Aalborg, Denmark, Sep. 17-22, 2005*

[C04ml] P. Casari, M. Levorato, S. Tomasin and M. Zorzi, "Analysis of Spatial Multiplexing for Cross-Layer Design of MIMO Ad Hoc Networks," *IEEE VTCspring, Melbourne, Australia, May 7-10, 2006*

[C05ml] M. Levorato, P. Casari, S. Tomasin and M. Zorzi, "An Approximate Approach for Layered Space-Time Multiuser Detection Performance and its Application to MIMO Ad Hoc Networks," *IEEE ICC, Instanbul, Turkey, Jun. 11-15, 2006*

[C06ml] M. Levorato, S. Tomasin and M. Zorzi, "Analysis of Cooperative Spatial Multiplexing for Ad Hoc Networks with Adaptive Hybrid ARQ," *IEEE VTCfall, Montreal, Canada, Sept. 26-28, 2006*

[C07ml] M. Levorato, P. Casari and M. Zorzi, "On the Performance of Access Strategies for MIMO Ad Hoc Networks," *IEEE Globecom, San Francisco, USA, 27 Nov.-1 Dec., 2006*

[C08ml] L. Badia, M. Levorato and M. Zorzi, "Analytical Investigation with Markov Models of Selective Repeat Truncated Type II Hybrid ARQ," *IEEE Globecom, San Francisco, USA, Nov. 27-Dec. 1, 2006*

[C09ml] P. Casari, M. Levorato and M. Zorzi, "DSMA: an Access Method for MIMO Ad Hoc Networks Based on Distributed Scheduling," *ACM IWCMC, Vancouver, Canada, July 3-6, 2006*

[C10ml] M. Levorato, S. Tomasin and M. Zorzi, "Strategies and Tradeoffs for Coded Cooperation in Wireless Networks," *International Workshop on Wireless Networks: Communications, Cooperation and Competition (WNC3), part of WiOpt, Limassol, Cyprus, April 16-20, 2007*

[C11ml] M. Levorato, S. Tomasin and M. Zorzi, "Coded Cooperation for Ad Hoc Networks with Spatial Multiplexing," *IEEE ICC, Glasgow, Scotland, June 24-28, 2007*

[C12ml] S. Tomasin, M. Levorato and M. Zorzi, "Analysis of Outage Probability for Cooperative Networks with HARQ," *IEEE ISIT, Nice, France, June 24-29, 2007*

[C13ml] L. Badia, M. Levorato and M. Zorzi, "An Improved Channel Quantization Method for Performance Evaluation of Incremental Redundancy HARQ Based on Reliable Channel Region," *Allerton Conference, Monticello, IL, USA, Sept. 26-28, 2007*

[C14ml] F. Librino, M. Levorato and M. Zorzi, "Distributed Cooperative Routing and Hybrid ARQ in MIMO-BLAST Ad Hoc Networks," *IEEE Globecom, Washington, DC, USA, Nov. 26-30, 2007*

[C15ml] M. Levorato, S. Tomasin and M. Zorzi, "Recursive Analysis of Ad Hoc Networks with Packet Queueing, Channel Contention and Hybrid ARQ," *ITA, San Diego, CA, USA, Jan. 27-Feb. 1, 2008*

[C16ml] M. Levorato and M. Zorzi, "Performance Analysis of Type II Hybrid ARQ with Low-Density Parity-Check Codes," *ISCCSP, St. Julians, Malta, Mar. 12-14, 2008*

[C17ml] M. Levorato and M. Zorzi, "On Error Control Schemes for Ad Hoc Networks with Multiuser Detection and Rate Control," *CISS, Princeton, NJ, USA, Mar. 19-21, 2008*

[C18ml] P. Casari, M. Levorato and M. Zorzi, "On the Design of Routing Protocols in MIMO Ad Hoc Networks under Uniform and Correlated Traffic," *IWCMC, Crete Island, Greece, Aug. 6-8, 2008*

[C19ml] M. Levorato, L. Badia and M. Zorzi, "Efficient Quantization for Feedback Controlled Networks with Type II Hybrid ARQ," *IEEE RAWNET, Berlin, Germany, Mar. 31st, 2008*

[C20ml] L. Badia, M. Levorato and M. Zorzi, "Analysis of Selective Retransmission Techniques for Differentially Encoded Data," *IEEE International Conference on Communications (IEEE ICC), Dresda, Germany, June 14-18 2009*

[C21ml] F. Librino, M. Levorato and M. Zorzi, "Performance Analysis and Resource Allocation in CDMA Cellular Networks with Relay Stations," *Workshop on Cooperative Mobile Networks (COCONET2) part of IEEE ICC 2009, Dresda, Germany, June 16-17 2009*

[C22ml] M. Levorato, U. Mitra and M. Zorzi, "On Optimal Control of Wireless Networks with Multiuser Detection, Hybrid ARQ and Distortion Constraints," *the 28th Conference on Computer Communications (IEEE INFOCOM), Rio de Janeiro, Brazil, April 19-25, 2009*

[C23ml] L. Badia, P. Casari, M. Levorato and M. Zorzi, "Analysis of an Automatic Repeat Request Scheme Addressing Long Delay Channels," *IEEE International Workshop on Underwater Networks (WUnderNet), Bradford, UK, May 26-29, 2009.*