We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Cooperative ARQ: A Medium Access Control (MAC) Layer Perspective

Jesús Alonso-Zárate*, Elli Kartsakli**, Luis Alonso**
and Christos Verikoukis*
*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
**Universitat Politècnica de Catalunya (UPC-EPSC)*

1. Introduction

Cooperative Automatic Retransmission reQuest (C-ARQ) schemes have become a very active research topic over the last years. C-ARQ schemes constitute a practical way of executing cooperation in wireless networks with already existing equipment. C-ARQ schemes exploit feedback from the receiver, i.e. cooperation is only executed when needed, and thus are sometimes referred to as cooperation on-demand cooperative schemes.

In short, the idea of C-ARQ is to exploit the fact that, due to the broadcast nature of the wireless channel, any transmission can be received by any of the stations in the transmission range of the transmitter. What has been traditionally considered as interference, is exploited in C-ARQ schemes to attain spatial diversity. Upon a transmission error, a retransmission can be requested from any (or some) of the stations which overheard the original transmission, which can act as spontaneous helpers (or relays). The result is that the destination of a packet can receive different copies of the same information arriving via statistically independent transmission paths, i.e., space diversity.

C-ARQ schemes have been already studied in the literature from a theoretical point of view and there is no doubt that, under some conditions, they can dramatically boost the performance of wireless communications compared to traditional ARQ, where retransmissions are performed only from the source. However, involving a number of users in a communication link requires coordination. To this end, efficient Medium Access Control (MAC) protocols are necessary to get the maximum efficiency of the communications. In this chapter we emphasize the important role of the MAC layer in this context of C-ARQ.

Along the chapter, we first review in Section 2 the motivation and operation of C-ARQ schemes into detail. We go through the parameters that affect the performance of these schemes and we point out the role of the MAC layer. Taking into account the specific requirements of the MAC layer in this kind of schemes, we present in Section 3 a novel high-performance MAC protocol specifically tailored for this purpose. Computer-based simulations are presented to evaluate the performance of the protocol. Finally, Section 4 concludes the chapter.

2. Cooperative ARQ (C-ARQ)

2.1 Background and Motivation

Traditionally, ARQ schemes have been used in communication networks to guarantee the reliable delivery of data packets. Upon the reception of a packet with errors, retransmissions are requested from the source (and along the same channel) until either the packet can be properly decoded or it is discarded for the benefit of the backlogged data.

Several variations of ARQ schemes have been proposed in the past to improve the performance of communications. These schemes perform well in wired networks where there is no correlation between consecutive packet error probabilities, i.e., packet errors are random and sparse. However, their performance in wireless networks is compromised by phenomena such as the shadowing and fading of the radio channel. In wireless channels, packet errors might come into bursts, and thus if a packet is received with errors, the immediate retransmissions will be also received with errors with high probability if they are performed through the same channel (Zorzi et al., 1997).

C-ARQ schemes constitute a practical solution to combat this fading nature of the wireless channel. Their operation is described in the following section.

2.2 Description of C-ARQ

Consider a wireless network formed by an arbitrary number of stations equipped with half-duplex radio frequency transceivers. In order to be able to execute a C-ARQ scheme, all the stations must listen to (overhear) every ongoing transmission in order to be able to cooperate if required. In addition, they should keep a copy of any received data packet (regardless of its destination address) until it is acknowledged (positively or negatively) by the destination. This packet is discarded whenever the destination successfully decodes the original packet.

It is assumed that, although both error detection and Forward Error Correction (FEC) bits are attached to all the transmitted data packets, errors can still occur due to the severe wireless channel impairments. Whenever a destination receives a data packet with unrecoverable errors, it broadcasts a retransmission request in the form of a control packet. This packet is referred to as the Call for Cooperation (CFC) packet. A *cooperation phase* is then initiated.

A subset of the stations which overheard both the original transmission from the source and the CFC from the destination, become *active relays* or *helpers*. As it will be further discussed later, some relay selection criteria can be attached to the CFC in order to activate the most appropriate subset of stations to act as helpers. Orthogonally in time (TDMA), frequency (FDMA or OFDMA), or code (CDMA), these active relays attempt to retransmit a copy of the original packet to assist in the failed transmission. For the sake of clarity in the explanation and without loss of generality, the data packets retransmitted by the relays will be referred to as *cooperative packets*.

Eventually, the destination might either receive a correct copy of the original packet from a relay or may be able to properly combine the different retransmissions from the relays to successfully decode the original packet. Otherwise, if the destination is not able to recover the data packet after some predefined time (cooperation time-out), it discards it. In any of the two cases, the cooperation phase is finished.

Although slight different variations to this general operation can be found in the literature, most of the proposed C-ARQ schemes follow this description. It is worth mentioning that the CFC has sometimes received the name of Negative ACK (NACK) in the literature (Dianati et al. 2006). However, this name falls short in describing the real function of the CFC. Besides informing the Negative ACK, it also calls for cooperation and, indeed, it could attach some relay selection criteria, among other control information required for the execution of a cooperative technique.

An example of operation of a C-ARQ mechanism is illustrated in Fig. 1. Therein, the communication between a source and a destination stations is assisted by an arbitrary number (*N*) of relays. In this particular example, the relays retransmit data orthogonally in time until the destination station can send the ACK.

The performance of a C-ARQ scheme might be mainly influenced by the following *four* parameters:

1) *The relay selection criteria*; as it could be expected, the number of potential helpers and the "quality" of those helpers will have a direct impact on the efficiency of the C-ARQ scheme. For this reason, there are several works focused on the design of efficient techniques to select either the best or a subset of the best potential helpers to act as relays (Gómez et al., 2007; Biswas & Morris, 2005).

2) *The PHY forwarding technique executed by the relays* (Nosratinia et al., 2004):

a. *Amplify and forward techniques*, when the relays transmit an amplified version of the original received signal, without demodulating or decoding it.

b. *Compress and forward techniques*, when the relays transmit a compressed version of the original transmitted signal, without decoding it.

c. *Decode and forward techniques*, when the relays transmit recoded copies of the original message. Note that using decode and forward, the recoding process can be done on the basis of repeating the original codification, recoding the original data (or only a relevant part of it), or using more sophisticated Space-Time Codes (STC) (Fitzek & Katz, 2006).

3) *The number of required retransmissions* necessary to decode a packet which can mainly depend on:
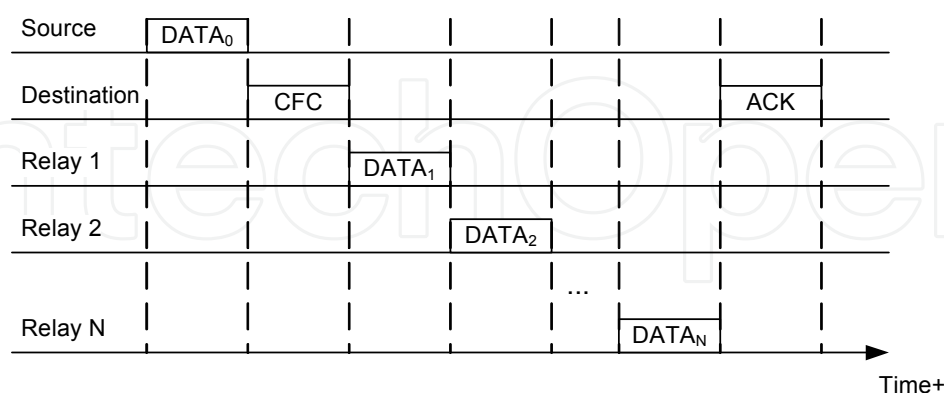

Fig. 1. C-ARQ Scheme with Time-Orthogonal Relays

a. The channel conditions between the source and the destination, the source and the relays, and the relays and the destination (Gómez & Pérez-Neira, 2006; Pfletschinger & Navarro, 2008).

b. The transmission scheme, which includes the forwarding technique executed by the relays and the combination technique executed by the destination station to combine the different retransmissions received from independent paths. The approach of combining different erroneous copies of a same packet to decode the original packet has been tackled in the past (Charaborty et al., 2005; Morillo-Pozo & García-Vidal, 2007).

4) *The MAC protocol* which is necessary to tackle with the contention among the relays. Just as an example, the ideal scheduling among the relays represented in Fig. 1 is impossible to attain in fully distributed networks without a central coordinator. Therefore, the set of active relays should contend for the channel in order to retransmit the packets. Efficient MAC protocols are necessary to execute a C-ARQ scheme in order to exploit the benefits of cooperation in wireless networks.

2.3 Motivation and Contributions of the Chapter

C-ARQ schemes have been so far analyzed from a fundamental point of view and mainly with emphasis on the PHY layer (Dianati et al. 2006; Zimmermann et al., 2004; Zimmermann et al., 2005; Gupta et al., 2004; Cerruti et al., 2008; Morillo-Pozo et al., 2005). These previous works put in evidence that C-ARQ schemes can yield an improvement in performance, lower energy consumption, and interference, as well as an extended coverage area by allowing communication at low Signal to Noise Ratios (SNRs). However, all of these contributions assume simplified topologies (with one or very few relays) and perfect scheduling among the relays at the MAC level. This scheduling might be difficult to attain in the fully decentralized scenario represented by the *cloud* of relays without infrastructure. Therefore, both the design of efficient MAC protocols and the evaluation of the actual performance of C-ARQ techniques considering the MAC overhead are mandatory if C-ARQ schemes are to find real application. Indeed, *this* is the main motivation for this chapter.

The focus in this chapter is on *time-orthogonal C-ARQ schemes*, which might be the easiest approach to implement with already existing off-the-shelf equipment. By slightly modifying the wireless controller (or driver), existing wireless cards could implement a   C-ARQ scheme. The emphasis is on the design and analysis of a novel MAC protocol to deal with the unique characteristics of the contention process that takes place among the active relays within a cooperation phase. Note that in the considered C-ARQ schemes, upon the initialization of the cooperation phase, the network has the three following *unique* characteristics:

1) The spontaneous "sub-network" formed by the active relays is ad hoc and thus there is no infrastructure responsible for managing the access to the channel.

2) This sub-network formed by the active relays surrounding the node calling for cooperation is *suddenly* (sharply) set into saturation conditions whenever the cooperation phase is initiated. Upon the transmission of a CFC packet, all the active relays have a data packet ready to transmit in order to assist the failed transmission. Therefore, heavy contention takes place in a previously idle network.

3) Opposite to general communications systems, now fairness is not a major issue to achieve. Indeed, the main goal is to attempt to assist the failed transmission as fast and reliable as possible, minimizing the use of the radio resources.

These three characteristics determine the way MAC protocols should be designed within the context of C-ARQ schemes in wireless networks. Considering the aforementioned

characteristics, we present in this chapter the design and performance evaluation of a novel high-performance MAC protocol for C-ARQ schemes, named DQCOOP.

It is worth mentioning at this point that, in the literature, there exists a family of *cooperative MAC* protocols which have not been designed for the execution of C-ARQ schemes in wireless networks, but they are aimed at solving other kind of interesting cooperative issues. For completeness, they are overviewed in the following section.

2.4 Related Work: Cooperative MAC Protocols

Some MAC protocols for cooperative communications have been proposed in the literature. Most of them have been designed to achieve a throughput enhancement, but actually none of them takes into account all of the unique characteristics of the on-demand C-ARQ schemes. It has to be mentioned that all these MAC protocols have been designed more as routing protocols with a cross-layer design that takes into account the transmission rates to decide the shortest route to a destination than as MAC protocols themselves. In what follows, a summary of the most relevant contributions is summarized.

In (Liu et al. 2007) two versions of the CoopMAC protocol are designed in the context of 802.11b WLANs in order to solve the performance anomaly induced by the multi-rate capability of the Distributed Coordination Function (DCF) of the standard (IEEE 802.11 Standard, 2007). Users with low transmission rate occupy the channel for long periods of time, reducing the overall throughput of the system and reducing the throughput seen by stations with higher transmission rates. The main idea of CoopMAC protocols is that stations transmit first to intermediate stations at a higher rate and then those intermediate stations transmit to the access point, reducing the total transmission delay. In the first version of CoopMAC, referred to as CoopMAC I, any station keeps updated a table with those stations that could potentially help in a transmission. Before transmitting any packet, a station calculates the shorter transmission path, either using direct communication with the intended destination or through any of the potential helpers with an entry in the table. In the case of using a relay, a previous handshake is done between the source station and the selected relay in order to ensure the validity of the route. The main drawback of CoopMAC I is that it requires the addition of three new fields in the Request To Send (RTS) frame and the addition of a new control frame named Helper ready To Send (HTS). As an alternative, CoopMAC II is proposed to overcome this problem. This second version of CoopMAC uses available empty fields in regular IEEE 802.11 control frames and eliminates the handshake between destination and helpers. Although the implementation is simpler, version II is more vulnerable to a change in the availability of a helping station caused by mobility. Computer simulations in (Liu et al., 2007) demonstrate the improved performance achieved with either CoopMAC I or II. Moreover, Korakis et al. implemented the protocol in actual WLAN cards, as reported in (Korakis et al., 2006). The main contribution of their work is the description of the overall implementation process and the limitations found when attempting to actually implement the protocol. These limitations were mainly due to the constraints imposed by the time sensitive tasks performed by the firmware of the wireless cards. In addition, the CoopMAC has been also adapted to wireless networks using directional antennas in (Tau et al., 2007).

On the other hand, both the Cooperative-MAC (CMAC) and FEC CMAC (FCMAC) protocols were presented in (Shankar et al., 2005) within the context of 802.11e networks to improve the overall performance and to ensure a certain QoS. In CMAC, a station detecting

an erroneous packet transmission between any other pair of source-destination stations decides to cooperate by retransmitting a copy of the overheard transmission as long as the received packet has no errors. A random backoff mechanism with a constant backoff window is applied to avoid collisions among different helpers. The size of the contention window of the helpers has to be very small in comparison to the contention window of the source in order to ensure that helpers retransmit their copy before the original source retransmits on its own the failed packet. Each helper transmits the copy of the packet at most once, to ensure that all available helpers cooperate and thus the benefits of diversity are obtained. On the other hand, FCMAC extends the operation of CMAC by fragmenting data packets into smaller blocks. Each block contains its own inner FEC field and the whole packet contains an outer FEC. Upon error detection of a whole packet, only a predefined number of randomly selected blocks among those received without errors are retransmitted. If the retransmitted blocks are those that were received with errors at destination, then the performance is improved. Otherwise, the increased overhead becomes useless. A possible solution consists in adding a negative acknowledgement (NACK) sent out by the destination upon error detection, indicating which are the blocks received with errors. However, the use of NACK in CMAC would imply higher overhead and again, it would require hardware modifications, thus breaking with the claimed backwards compatibility. The main limitation of CMAC and FCMAC is that they rely on the fact that helpers can learn whether other transmissions between any pair of source and destination are successful or not only by overhearing the radio channel.

In (Wang & Yang, 2005), the Cooperative Diversity Medium Access with Collision Avoidance (CD-MACA) protocol is proposed within the context of wireless ad hoc networks operating over the CSMA/CA protocol. Whenever a source terminal fails to receive the CTS packet, all those stations that had properly received it, take the place of the source terminal and retransmit the data packet. An analytical model based on Markov chain theory is proposed to obtain the achievable throughput of the system considering cooperation. Although the general idea of CD-MACA is rather interesting, the definition in (Wang & Yang, 2005) is quite general and several implementation details are not considered.

From an energy-efficient perspective, another cooperative MAC protocol is also presented within the context of ad hoc networks in (Azing et al., 2005). This proposal integrates cooperative diversity into two different wireless routing protocols by embedding a distributed cooperative MAC. The initial path establishment performed by the routing protocol can be done either considering cooperation or not. Cooperation is then achieved by forcing all the stations to act as a distributed virtual antenna, through which simultaneous transmissions are separated with CDMA.

In (Sadek et al., 2006) a cooperative MAC protocol was presented within the context of a mesh network formed by an access point, a number of regular stations, and one fixed wireless router (relay). A fixed TDMA scheme is applied and empty slots are used for cooperative relaying. The relay station keeps a copy of all those packets that are not properly received by the Access Point (AP). At the beginning of each time slot, the relay listens to the channel. If the channel is idle, it retransmits the packet at the head of its queue. Based on this main idea, two specific algorithms are proposed to exploit the benefits of cross-layer design between the PHY and MAC layers.

All these MAC protocols have been designed to achieve an improvement in the network performance by transmitting through faster multi-hop routes. However, none of them takes

into account the unique characteristics of the C-ARQ schemes for their implementation in on-demand cooperative schemes. DQCOOP is presented in the next section as a novel MAC protocol that has been tailored to meet the requirements of the C-ARQ scenario. It constitutes the adaptation of the high-performance DQMAN protocol (Alonso-Zárate et al., 2008a) to this kind of scenarios.

3. DQMAN for C-ARQ: DQCOOP

The aim of this section is to present DQCOOP as an extension and adaptation of the high-performance DQMAN (Alonso-Zárate et al., 2008a) to match the unique requirements posed by the C-ARQ schemes. DQMAN, in its turn, is the extension of the infrastructure-based DQCA protocol (Alonso-Zárate et al., 2008b) for wireless ad hoc networks. The new resultant protocol is called DQCOOP. The rules of DQMAN and DQCA will not be described into detail in this chapter as they can be found in (Alonso-Zárate et al., 2008a) and (Alonso-Zárate et al., 2008b), respectively.

In short, the basic idea of DQMAN is that any idle station with data to transmit listens to the channel for a randomized period of time before establishing its cluster. This Clear Channel Assessment (CCA) period gets the name of Master Selection Phase (MSP). If the channel is idle for the whole MSP, then a cluster is established. The station becomes master and starts broadcasting a periodical clustering beacon (CB) that allows neighbor stations to get synchronized and become slaves. The master operates as such for as long as there is data activity within its cluster. Therefore, the cluster structure changes along time as a function of the aggregate traffic load of the network. Once the cluster is established, the master station transmits its own data and it acts as the AP of a WLAN wherein DQCA can be executed. For completeness, we review the basic protocol rules of DQCA in the next section.

3.1 DQCA Overview

The purpose of this section is to highlight the basic features of DQCA. As demonstrated in (Alonso-Zárate et al., 2008b), DQCA outperforms the widely commercially spread Distributed Coordination Function (DCF) of the IEEE 802.11 Standard and remains stable even when the traffic load occasionally exceeds the channel capacity.

DQCA is a MAC protocol designed to manage the access to the channel in the uplink of an infrastructure WLAN. Time is divided into MAC frames, and each frame is divided in three parts separated by a Short Inter Frame Space (SIFS) necessary to tolerate propagation delays, turnaround times, and processing delays. The three parts, depicted in Fig. 2, are:

i) A Contention Window (CW) further divided into *m* access minislots wherein the nodes can send a short chip sequence named Access Request Sequence (ARS) to request access to the channel. An ARS is a short chip sequence that contains no explicit information but has a specific and predefined pattern that allows the AP to distinguish between an idle minislot, the presence of just one ARS, or the occurrence of a collision between two or more simultaneous ARS.

ii) A data slot reserved for the transmission of data packets.

iii) A feedback part wherein the AP broadcasts a Feedback Packet (FBP) that contains the data acknowledgment, the state of the each of the minislots of the CW for contention resolution algorithm, and a 'final message bit' that is enabled (set to one) by the AP to identify the last data packet (fragment) of a message. Of course, nodes must also include a

'final message bit' in their data packet transmissions in order to advertise the transmission of the final fragment of each message.

All the nodes execute three sets of simple rules at the end of each MAC frame. By simply using the feedback information attached to the FBP, they can update the state of two distributed queues (explained below) to execute the access algorithm. According to the protocol rules, DQCA operates as a random access protocol when the traffic load is low (an immediate access rule of the protocol allows a station to get access to the channel immediately if the distributed queues are empty), and it switches smoothly and automatically to a reservation protocol as the traffic load increases. Therefore, it attains the better of the access methods.

The protocol operation is based on two concatenated distributed queues, the Collision Resolution Queue (CRQ) and the Data Transmission Queue (DTQ). The CRQ is responsible for the resolution of collisions among ARS and the DTQ handles the transmission of data. The number of occupied positions (or elements) in each queue is represented by an integer counter (RQ and TQ for the CRQ and the DTQ, respectively). Both counters have the same value for all the nodes in the system and are updated according to a set of rules at the end of each frame. Each node must also maintain and update another set of counters that reveal its position in the queue (pRQ and pTQ for the CRQ and the DTQ, respectively). By the term "position" it is meant the relative order of arrival (or age) of the node in the respective queue. In the CRQ, each position (or element) is occupied by a set of nodes that suffered an ARS collision (i.e. attempted an ARS transmission in the same access minislot of the same CW). The DTQ contains the nodes that successfully reserved the channel through an ARS and therefore each queue element corresponds to exactly one node.
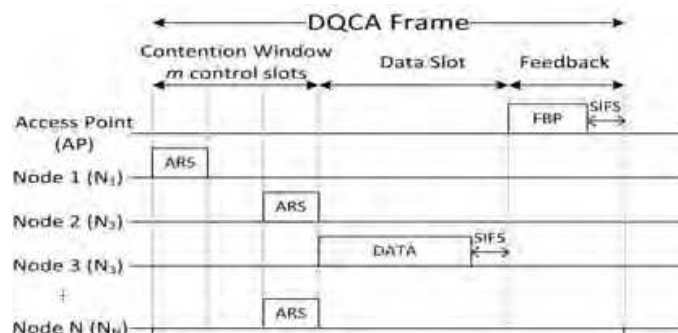

Fig. 2. DQCA Frame Structure

3.2 Motivation and Problem Statement

The intuitive idea behind DQCOOP is that the *destination* asking for cooperation gets the role of *master* and coordinates the retransmissions from the relays, which become *slaves*, as in DQMAN. Then, a temporary cluster is established around the destination and a variation of DQCA can be executed. This is represented in Fig. 3.
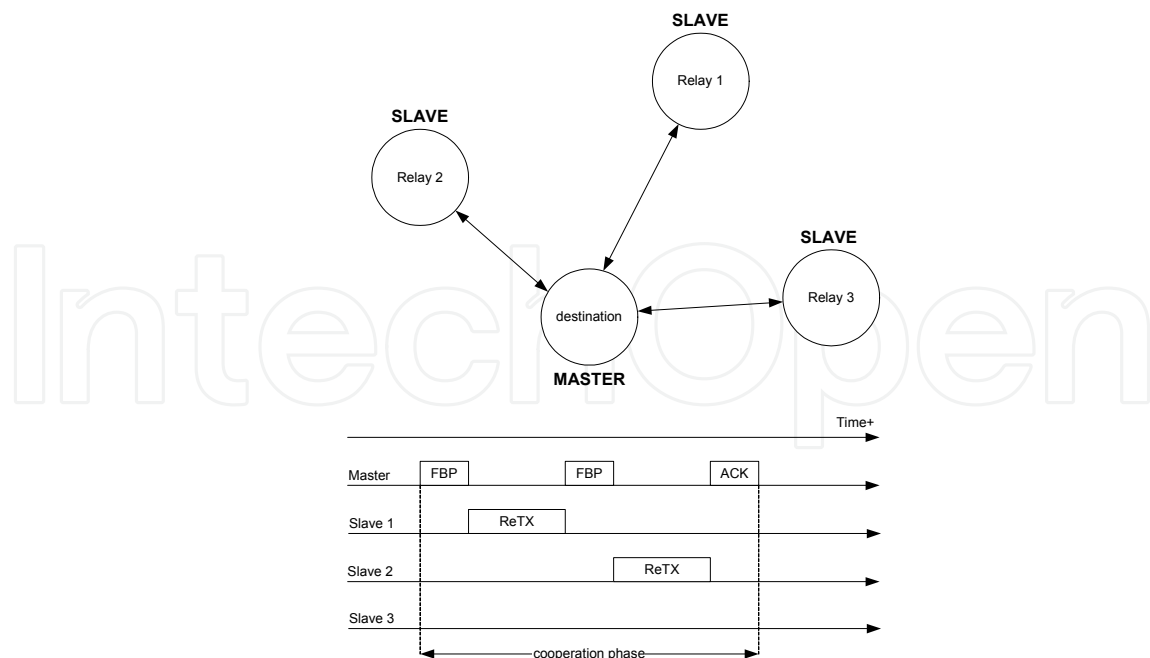
Fig. 3. Master-Slave Architecture of DQCOOP (simplified example)

The master, i.e., the destination, initiates the periodic broadcast of the FBP and creates a temporary cluster. A cooperation phase is initiated. The slaves, i.e., the relays, request access to the channel to retransmit their cooperative packet (retransmissions of the original source transmissions) by executing a variation of the DQCA rules. It is assumed that the relays attempt to retransmit persistently until the cooperation phase is finished. Whenever the cooperation phase is finished either an ACK or a NACK packet is transmitted, indicating either the successful or unsuccessful recovery of the data packet originally received with errors, respectively.

However, DQMAN, as defined in (Alonso-Zárate et al., 2008a), would be inefficient in managing the access to the channel in a C-ARQ scheme. This is mainly due to the fact that upon cooperation request (broadcast by the destination), the group of active relays forms an ad hoc network wherein all the active stations suddenly have a data packet ready to be transmitted. This turns temporarily the network from idle to saturation conditions. This idle-to-saturation sharp transition would cause DQMAN to spend a non-negligible start-up time before attaining its high performance, mainly due to:

1) The simultaneous channel access requests from the active relays in the first frame immediately after the transmission of the CFC would have a high probability of collision. Therefore, some empty frames would be needed until the first collision could be solved and data retransmissions could actually start.

2) Upon the transmission of the first FBP, all the active relays (slaves) would retransmit in the following frame by executing the *immediate access rule* of DQCA (Slotted ALOHA access for low traffic loads). All these transmissions would collide, causing a waste of resources for the duration of a complete MAC frame.

3) Even with the immediate access rule disabled, an empty frame would be present when the collision resolution process starts due to the MAC frame structure with the feedback broadcast at the end of the frame.

Therefore, it is necessary to expand and adapt the DQMAN operation to take into consideration the aforementioned issues that may potentially degrade its performance in C-ARQ schemes. DQCOOP is presented in the next section with the goal of attaining the near-optimum performance of DQMAN within the context of the considered C-ARQ scheme.

3.3 Protocol Description
The core operation of DQCOOP is highly based on DQMAN. However, the *clustering algorithm* and the *MAC protocol* (frame structure and protocol rules) are modified to meet the requirements of the C-ARQ scheme. Their descriptions are presented in the next two sections.

3.3.1 Clustering Algorithm
In DQCOOP, the clustering algorithm of DQMAN is modified as it follows:

1) The destination, and not the transmitter as in DQMAN, takes the *master* role when a cooperation phase is initiated with the transmission of a CFC packet. Some of the relays which received the original data packet (received with errors by the destination to trigger a cooperation phase) and also receive the CFC transmitted by the destination become active relays. These active relays get the role of *slaves*. A cluster is then established. The master periodically broadcasts a FBP, in the same way as in DQMAN, to provide the slaves with the minimum feedback information necessary to execute the protocol rules at the end of each frame.

2) There is no CCA prior to the establishment of the cluster. This means that the destination station does not have to contend with other users to get access to the channel. Therefore the contention within the MSP associated with DQMAN is avoided with DQCOOP. This can be actually performed as the CFC is transmitted instead of the ACK when receiving a packet with errors. ACK packets are usually given priority over all kind of traffic (in wireless networks), and thus there is no need for contention in this case.

3) The cluster is broken up whenever the master either manages to decode the original packet or discards the packet. The cooperation phase is ended with the transmission of the ACK packet. Otherwise, if a maximum time-out expires and the original packet cannot be decoded, a NACK packet is transmitted and the cluster is broken up as well. That is, in fact all the stations become idle upon the transmission of either the ACK or the NACK by the master.

3.3.2 The MAC Protocol: Frame Structure and Protocol Rules
When a cooperation phase is initiated, time is divided into five parts as represented in Fig. 4. Upon the transmission/reception of each FBP, all the stations execute the protocol rules of DQCA. The five parts of a cooperation phase within the context of DQCOOP are:

1) A **CFC transmission**. The cooperation phase is initiated when a CFC is broadcast by the destination station upon the reception of a data packet with errors. This CFC takes the form of a special FBP and indicates that **immediate access is forbidden**.

2) An **initial contention window composed of m_0 minislots** follows the CFC transmission wherein every active relay station randomly selects (with equal probability) one out of the m_0 minislots where to send an Access Request Sequence (ARS).

3) A **FBP transmission.** A FBP is broadcast by the master station with the **feedback**

information regarding the state of each of the m_0 **previous minislots**. As in DQMAN, for each minislot, this information can have one out of three values. It can be empty (E), i.e., no ARS transmitted, success (S), i.e., exactly one ARS transmitted, or collision (C), i.e., more than one ARS transmitted in the same minislot (no matter how many).

4) **A number of regular DQMAN consecutive MAC frames** follow this first FBP until the cooperation phase is ended. The rules of DQMAN, with the exception of the immediate access rule, are executed to manage the data retransmissions and the resolution of the collisions. **The contention window of these frames has *m* minislots**, where in general $m<m_0$, although this is not a mandatory condition.

5) An **ACK or NACK transmission**. Whenever the destination is able to **successfully decode** the original packet, it broadcasts an **ACK** packet indicating the end of the cooperation phase. A **NACK** is transmitted if the packet cannot be decoded at some point in time.

Short Inter Frame Spaces (SIFS) are left between each of the parts of the cooperation phase to compensate for non-negligible propagation and data processing delays and turnaround times to switch the radio transceiver from receiving to transmitting mode.

It is worth mentioning that the value of m_0 must be tuned according to the expected number of active relays. The higher the number of active relays, the higher the value of m_0 in order to reduce the probability that all the access requests collide in the first frame. However, a high value for m_0 has a cost in terms of control overhead. On the other hand, as long as at least one access request is successful, the data transmission process can be initiated from the first MAC frame, avoiding thus the loss of resources.
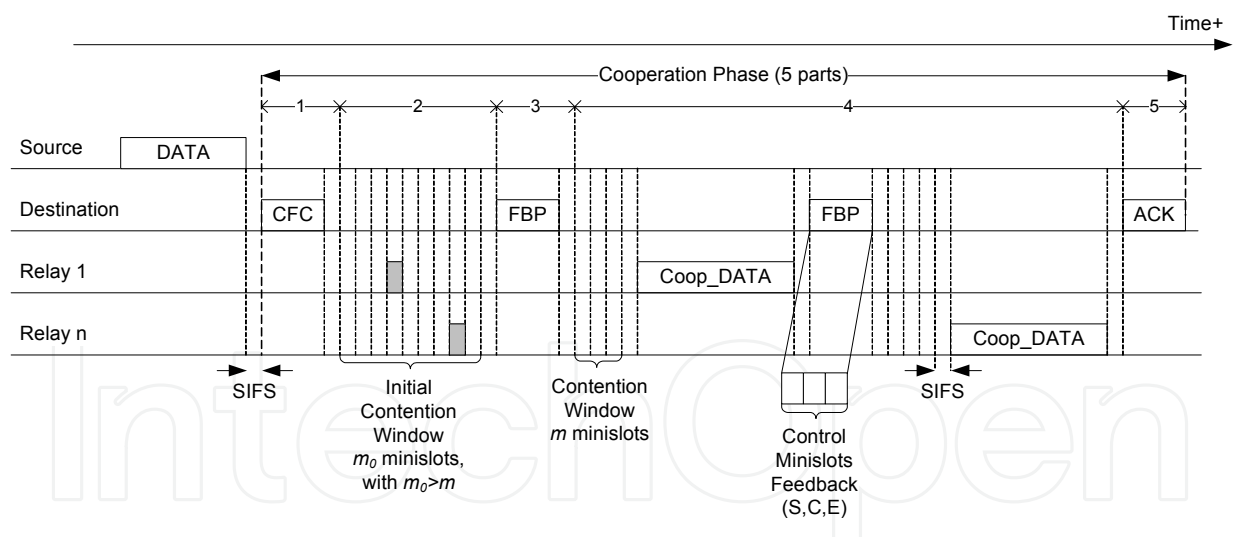

Fig. 4. DQCOOP MAC Frame Structure

3.3.3 Operational Example

A simple network layout with six stations is considered, all of them in the transmission range of each other. A source station (*S*) transmits to a destination station (*D*) with the support of relays *R1*, *R2*, *R3*, and *R4*. TQ and RQ represent the size of the DTQ and the CRQ, respectively, and pTQ_i and pCR_i represent the position of the ith user in the DTQ and CRQ, respectively.

The cooperation phase is represented in Fig. 5 and explained as follows:

1) Upon the reception of the data packet with errors, D initiates a cooperation phase by broadcasting a CFC. This packet sets the start of frame 0.

2) Frame 0 contains 5 access minislots (m_0=5). The set of relays {$R1, R2, R3, R4$} select the set of minislots {3, 1, 5, 5}.

3) At the end of frame 0, D broadcasts the FBP with the following feedback information regarding the state of the minislots, i.e., {*Success, Empty, Success, Empty, Collision*}.

4) Upon the execution of the protocol rules, $R2$ gets the first position of DTQ, $R1$ gets the second position of DTQ, and both $R3$ and $R4$ get the first position of CRQ. In terms of the four integer number representing the queues, this can be written as {$pTQ1, pTQ2, pTQ3, pTQ4$}={2,1,0,0} and {$pRQ1, pRQ2, pRQ3, pRQ4$}={0, 0, 1, 1}. On the other hand, TQ=2 and RQ=1.

5) During frame 1, both the data transmission and the collision resolution work in parallel. At the beginning of the frame, containing 2 access minislots (m=2), $R3$ and $R4$ attempt to solve their collision. They reselect an access minislot where to send an ARS. In this case, they select minislots 1 and 2 respectively, and thus they successfully solve their collision.

6) On the other hand, $R2$, which is at the first position of DTQ, transmits data (a retransmission of the original packet).

7) At the end of frame 1, the FBP broadcast by D indicates that a transmission has been successful and the next station in DTQ should transmit in the following frame. In addition, the feedback information on the state of the minislots allows $R3$ and $R4$ to queue, orderly in time, in DTQ.
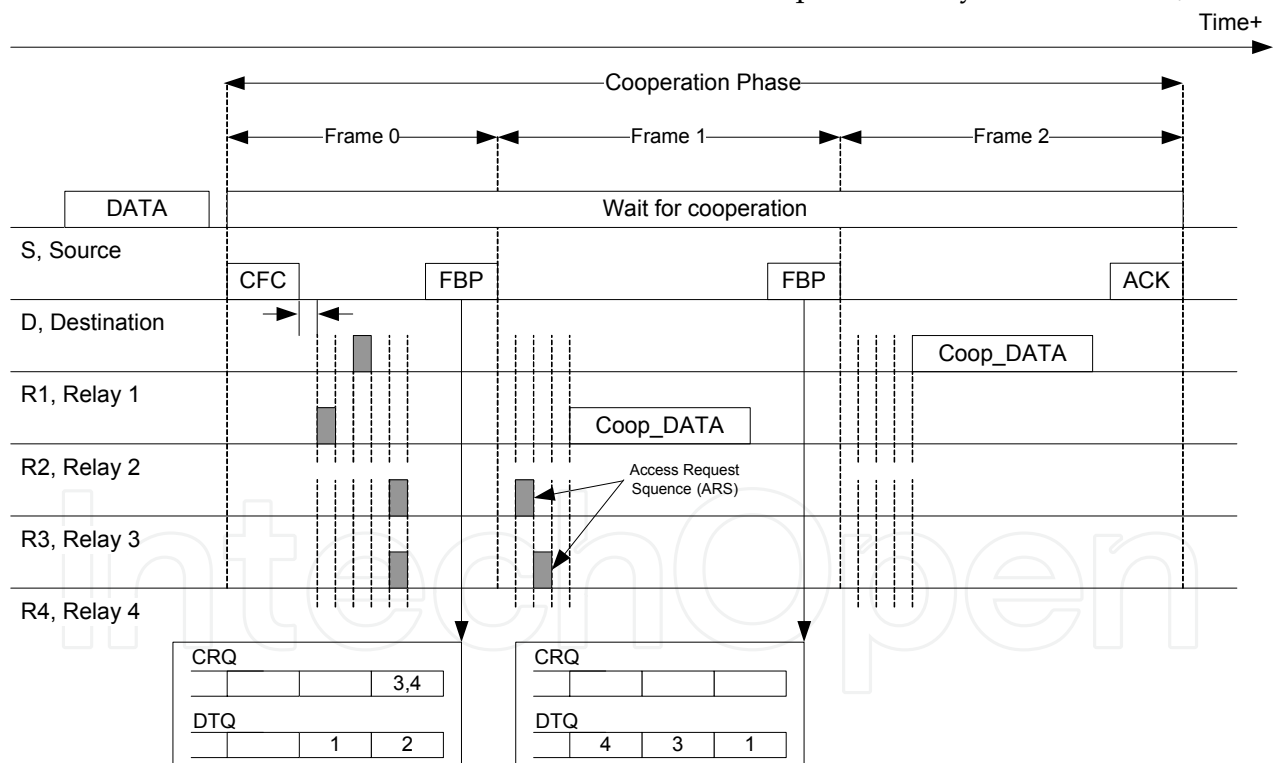

Fig. 5. DQCOOP Example of Operation

8) In frame 2, there are no collisions to be solved and thus the minislots are empty. $R1$ transmits data.

9) Upon the reception of the retransmission from $R1$, D is able to successfully decode the original packet. Therefore, it transmits an ACK packet indicating the end of the cooperation phase. All the relays discard the buffered cooperative packet.

3.4 Performance Evaluation

The performance of DQCOOP is evaluated in this section through a C++ custom-made simulator. In order to focus on the evaluation of the cooperation phases, a single-hop network wherein all the data transmissions from a fixed source to a fixed destination are received with errors is considered. That is, the destination always broadcasts a CFC packet upon the reception of every original data packet received from the source station. Moreover, the source has always a packet ready to be transmitted to the destination.

In this performance evaluation, we will measure the average packet transmission delay defined as the period of time elapsed from the moment that a packet is first transmitted from the source until it can be decoded at destination after receiving K retransmissions. For all the experiments, we assume that a constant number of relays are activated within each cooperation phase. Furthermore, and without loss of generality, the destination is considered to require a constant number K of retransmissions from the relay set to decode the original packet.

The simulation parameters are summarized in Table 1.

Parameter	Value	Parameter	Value
Control Rate	6 Mbps	MAC header	34 bytes
Data Rate (Source)	24 Mbps	PHY preamble	96 µs
Data Rate (Relays)	54 Mbps	ACK, CFC, FBP length	14 bytes
Packet Length	1500 bytes	*SlotTime*	10 µs
ARS	10 µs	SIFS	10 µs

Table 1. Simulation Parameters

3.4.1 Number of Minislots in the Cooperation Phase

The average packet transmission delay as a function of the number of active relays in a cooperative phase is represented in Fig. 6 when $K=3$ and for different values of m_0 and m, which in addition accomplish that $m_0=m$. Each curve represents the results obtained with different number of access minislots (m).

For low values of m, the average packet transmission delay gets lower as the value of m increases thanks to the faster collision resolution process. However, increasing the number of access minislots also increases the MAC overhead. The addition of an extra minislot entails an extension of the frame duration (devoted to overhead) and also enlarges the size of the FBP that contains the state of each one of the minislots. Therefore, as it can be seen in the figure, for high values of m, e.g., $m=10$, the fact that the collision resolution becomes shorter in time does not pay off the increase in the protocol overhead when the number of active relays is low and thus the average packet transmission delay gets higher. This can be better appreciated in Fig. 7 where the average packet transmission delay for the scenario with 5 relays is plotted as a function of the number of access minislots $m=m_0$. In this curve it is easier to see that, for low number of access minislots, an increase in the number of minislots leads to lower average packet transmission delays. However, over a given threshold, the faster resolution of collisions due to the longer contention window does not compensate for the MAC overhead and the average packet transmission delay increases

with the number of access minislots. For this reason, it is necessary to find a good compromise between the faster collision resolution and the protocol overhead. This tradeoff will be further discussed later in the next section.
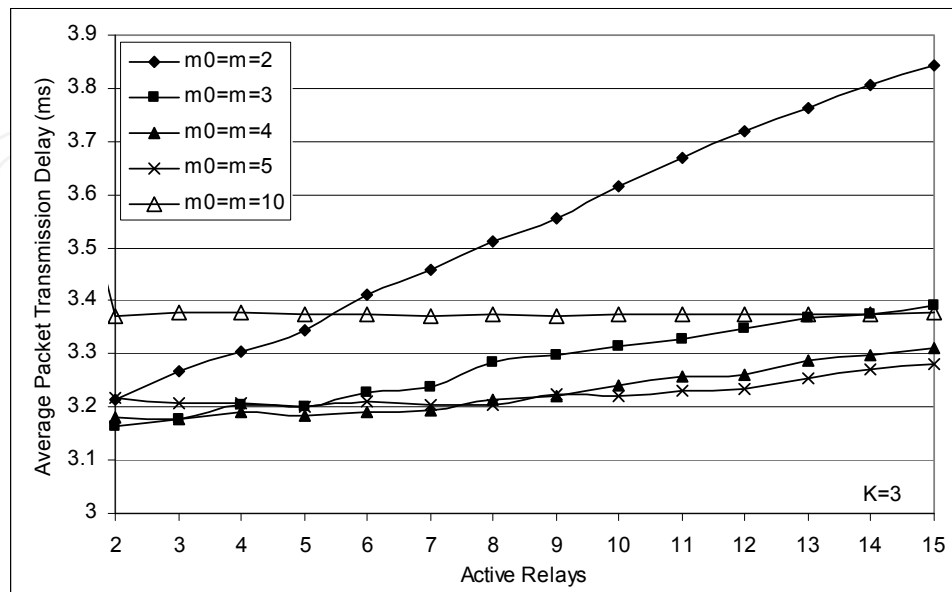

Fig. 6. Average Packet Transmission Delays for Different Values of $m_0=m$
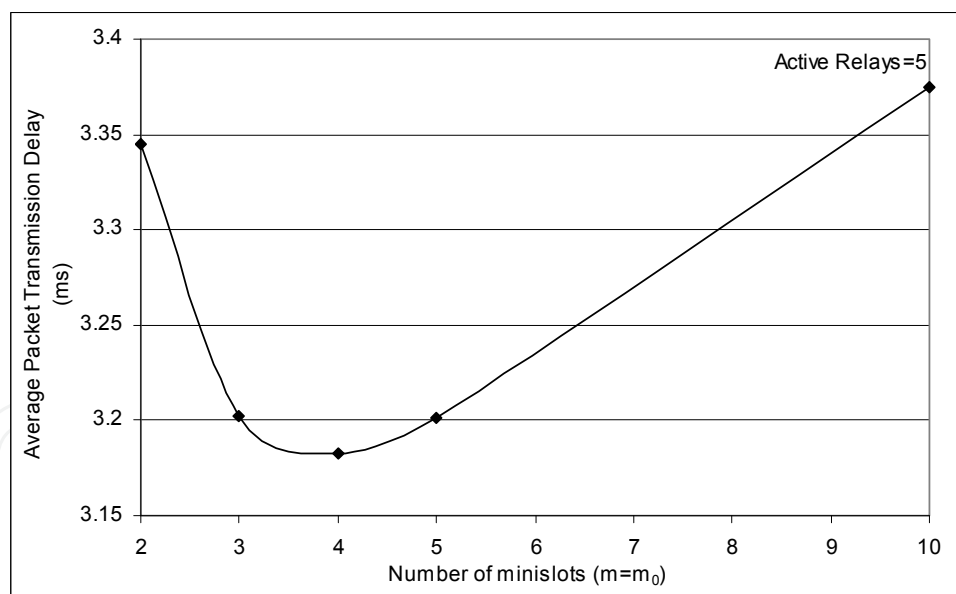

Fig. 7. Average Packet Transmission Delay for Different Values of $m_{0=m}$

Getting back to the results in Fig. 6, they show that the average packet transmission delay drops remarkably when the number of access minislots is at least equal to 3. Higher values of m do not result in any substantial reduction of this time. Therefore, as it happens with the DQCA protocol (Alonso-Zárate et al., 2008b), a good operational point for DQCOOP is to set $m=3$.

It is interesting to evaluate whether this discussion is still valid for any arbitrary number of required retransmissions (K). The average packet transmission delay is plotted in Fig. 8 as a function of the value of K and for different values of $m_0=m$ when the number of active relays is 15. In all cases, there is a considerable reduction of the average packet transmission delay when shifting from 2 to 3 minislots. However, there is no much interest in increasing the number of access minislots to higher values than 3, at least in terms of packet transmission delay. Therefore, it is important to reinforce the already known argument that the number of access minislots should be set to 3 in any DQCA-like protocol.

However, it seems reasonable to think that the value of m_0 (the number of access minislots within the very first frame after the transmission of the CFC) could be set to a higher value than m in order to *absorb* the first multiple access request arrival from all the active relays. Note that the first frame is the one that receives the maximum number of simultaneous access requests. In subsequent frames, the requests are split into smaller groups according to the m-ary tree-splitting collision resolution operation of DQMAN.

In the next section, m is set to 3 and the performance of the protocol is evaluated for different values of m_0. The aim is to evaluate the reduction of the average packet transmission delay for $m_0>m$.
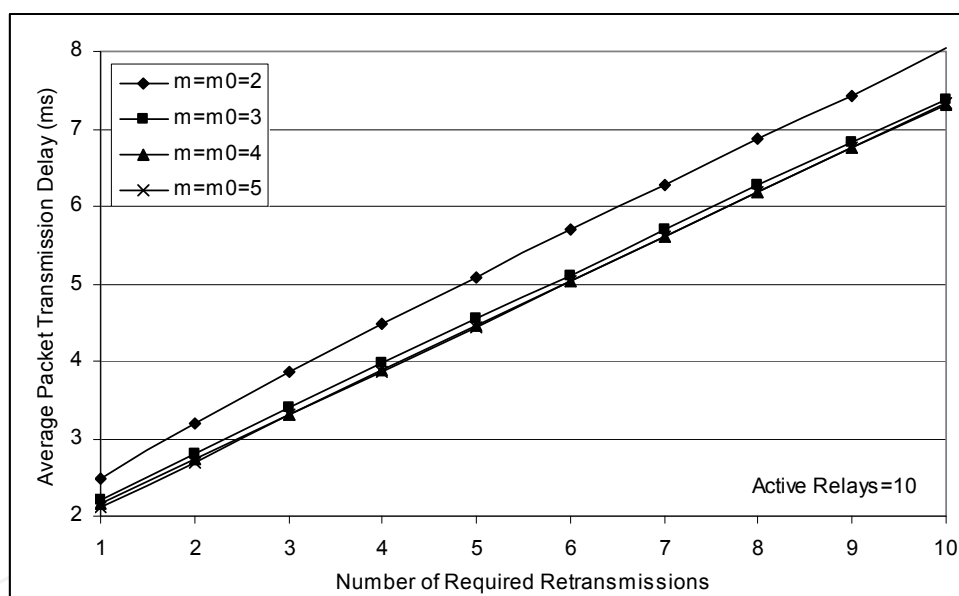

Fig. 8. Average Packet Transmission Times for Different Values of K

3.4.2 Number of Minislots in the Start-up Phase (m_0)

The performance of DQCOOP for different values of m_0 is evaluated in this section. As discussed before, an increase in the number of minislots of the first frame reduces the probability of collision in the first access requests upon initialization of a cooperation phase and, therefore, it should yield a lower average packet transmission delay. However, it also entails an increase of the protocol overhead (frame length and amount of required feedback information).

In order to quantify this tradeoff, first note that the duration of a cooperation phase can be decomposed as the sum of time devoted to the transmission of data and the overhead due to

the necessary MAC protocol. This overhead time includes silent intervals as well as the time devoted to the transmission of control packets. Considering this, the *relative overhead* is defined as the ratio between the overhead time in the cases that $m_0>1$ and the overhead time when $m_0=1$ (this latter case is the worst case in terms of overhead since all the relays collide in the first access request with probability one). This definition allows plotting the curves with different values of K in the same vertical axis and also makes the results independent of the absolute values of the transmission rates used for the simulation and the numerical evaluation.

The relative overhead is plotted in Fig. 9 as a function of the value of m_0, for different number of required retransmissions (K), and considering a total number of 5 active relays. The first observation is that there is a close relationship between the overhead of the protocol and the value of m_0. The value of the relative overhead is very sensitive to the value of m_0 if the number of required retransmissions is low. This means that if the value of K is low, the accurate tuning of the value of m_0 has a remarkable effect on the performance of the C-ARQ scheme. All the curves show a local minimum of the relative overhead for any pair of values of m_0 and K. However, on the other hand, the higher the values of K, the more flat the curves become. This means that if the number of required retransmission is high, the value of m_0 becomes a non-critical parameter on the performance of DQCOOP.

The main reason for this behavior is that when the number of required retransmissions is high and thus the duration of the cooperation phase is long, the impact of the overhead of the first frame on the performance of DQCOOP is low. Note that if K retransmissions are needed, at least K frames are necessary.

On the other hand, it seems reasonable to believe that the selection of the value of m_0 should depend on the number of active relays (which request access simultaneously in the first frame). In order to evaluate this relationship, the average packet transmission delay is plotted in Fig. 10 for $K=3$. Different curves are plotted for different number of active relays and as a function of the value of m_0.
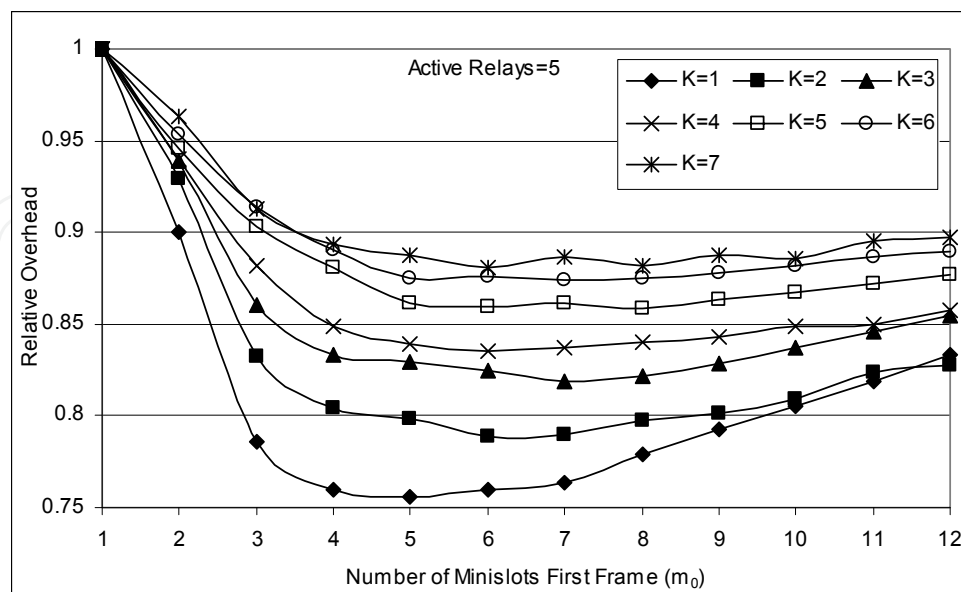

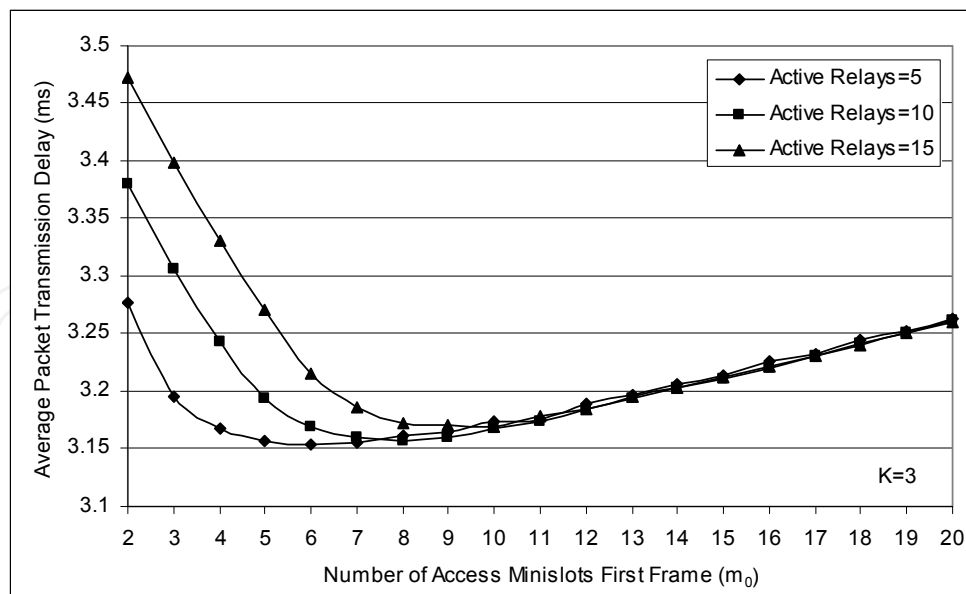Fig. 9. Protocol Relative Overhead (DQCOOP)

Fig. 10. Average Packet Transmission Delay as a Function of m_0

It is worth noting that when $m_0 \geq 10$ the three curves almost overlap. This means that, if this condition is fulfilled, the average packet transmission delay is almost equal and independent of the number of active relays. In addition, the value of the average packet transmission delay at m_0=10 is not substantially bigger than the one at the respective minimum values that can be found for m_0=6 (for 5 active relays), m_0=7 (for 10 active relays), and m_0=10 (for 15 active relays). This constitutes a worthwhile design guideline since by setting m_0=10 the average packet transmission delay for any value of K can be predicted with reliable accuracy regardless of the number of active relays in each cooperation phase (considering a practical situation with no more than 15 active relays). In addition, this fact relaxes the configuration requirements of the network, which is of remarkable interest when operating in fully decentralized and spontaneous networks.

4. Conclusions

In this chapter we have highlighted the important role of the MAC layer in the performance of C-ARQ schemes. Typically, these kinds of schemes have been evaluated from fundamental points of view and assuming perfect scheduling among the relays. However, we have shown that efficient MAC protocols are necessary to fulfill the specific requirements posed by C-ARQ schemes and to get the most of their potential to increase the efficient of wireless communications.

In addition, we have presented the DQCOOP protocol as an extension and adaptation of DQMAN to efficiently coordinate the contention among the relays in a C-ARQ scheme. It has been necessary to redesign the initialization phase of a DQMAN cluster so as to manage the idle-to-sharp traffic transition that takes place upon the transmission of a CFC. Since the active relays attempt to help simultaneously, the first contention window of DQMAN has to be resized. In addition, the protocol frame structure and the protocol rules have been also modified to optimize the performance of DQMAN in the context of C-ARQ schemes.

The performance of the protocol has been evaluated with computer simulations. Results show that the performance of DQCOOP can be independent of number of active relays and the number of access minislots. This is a desirable characteristic in fully decentralized networks, as is the case of ad hoc networks, where there might be no previous knowledge of the network topology and configuration. Results also show that this independency can be simply accomplished by setting the number of access minislots to 3 (attaining a faster resolution of collisions compared to the transmission of data) and properly dimensioning the number of access minislots in the very first frame, which is also modified to avoid an otherwise certain empty data field. This last modification aims at absorbing the first simultaneous access request by all the active relays. In fact, results show that the number of access minislots in the very first frame can be overdimensioned at almost no cost, and thus the performance of DQCOOP can be independent of the number of relays. The cost of increasing by one unit the number of access minislots in terms of overhead pays off the reduced probability of collision in the first access request.

5. References

Alonso-Zárate J., Gómez J., Verikoukis C., Alonso L., & Pérez-Neira A. (2006). Performance Evaluation of a Cooperative Scheme for Wireless Networks, *Proceedings of the IEEE PIMRC*, pp. 1-5, ISBN 1-4244-0329-4, Helsinki, Finland, September 2006

Alonso-Zárate J., Kartsakli E., Skianis C., Verikoukis C., & Alonso L. (2008a), Saturation Throughput Analysis of a Cluster-based Medium Access Control Protocol for Single-hop Ad Hoc Wireless Networks, *Simulation: Transactions of the Society for Modeling and Simulation International*, Vol. 84, No. 12, 619-633, ISSN 0037-5497

Alonso-Zárate J., Kartsakli E., Cateura A., Verikoukis C., & Alonso L. (2008b). A Near-Optimum Cross-Layered Distributed Queueing Protocol for Wireless LAN, *IEEE Wireless Communications Magazine, Special Issue on MAC protocols for WLAN*, Vol. 15, No. 2, (February 2008) 48-55, ISSN 1536-1284

Azgin A., Altunbasak Y., & AlRebig G. (2005). Cooperative MAC and Routing Protocols for Wireless Ad Hoc Networks, *Proceedings of the IEEE GLOBECOM 2005*, ISBN: 0-7803-9414-3, December 2005

Biswas S. & Morris R. (2005). ExOR: Opportunistic Multi-hop Routing for Wireless Networks, *Proceedings of the SIGCOMM '05*, pp. 133-144, ISBN 1-59593-009-4 , New York, NY, USA, 2005.

Campbell G. et al. (2002), Method and apparatus for detecting collisions and controlling access to a communications channel, *US Patent no. US6408009 B1*, June 2002.

Cerruti I., Fumagalli A., & Gupta P. (2008). Delay Model of Single-Relay Cooperative ARQ Protocols in Slotted Radio Networks Poisson Frames Arrivals, *IEEE/ACM Trans. on Networking*, Vol. 16, No. 2, (April 2008) 371-382, ISSN 1063-6692

Chakraborty S. S., Liinaharja M., & Ruttik K. (2005). Diversity and packet combining in Rayleigh fading channels, *IEE Proceedings-Communications*, Vol. 152, No. 3, (June 2005) 353 – 356, ISSN 1350-2425

Dianati, M., Ling X., Naik K., & Shen X. (2006). A Node-Cooperative ARQ Scheme for Wireless Ad Hoc Networks, *IEEE Trans. on Vehicular Technology*, Vol. 55, No. 3, (May 2006) 1032-1044, ISSN 00189545

Fitzek F. H. P. & Katz M. D. (2006). *Cooperation in Wireless Networks: Principles and Applications*, Ed. Springer, ISBN 978-1-4020-4710-7, The Netherlands

García-Vidal J., Guerrero-Zapata M., Morillo J., & Fusté D. (2007). A Protocol Stack for Cooperative Wireless Networks, In: *Wireless Systems and Mobility in Next Generation Internet, in Lecture Notes in Computer Science (LNCS)*, 62-73, Springer, ISBN: 978-3-540-70968-8, Springer, (2007)

Goldsmith A. (2005). *Wireless Communications*, Cambridge University Press, ISBN 0521837162

Gómez J. & Pérez-Neira A. (2006). Average Rate Behavior for Cooperative Diversity in Wireless Networks, *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 5309-5402, ISBN 0-7803-9389-9, Kos, Greece, May 2006

Gómez J., Alonso-Zárate J., Verikoukis C., Pérez-Neira A., & Alonso L. (2007). Cooperation On Demand Protocols for Wireless Networks, *Proceedings of the IEEE PIMRC*, pp. 5, ISBN: 978-1-4244-1144-3, Athens, Greece, September 2007

Gupta P., Cerruti I., & Fumagalli A. (2004). Three Transmission Scheduling Policies for a Cooperative ARQ Protocol in Radio Networks, *Proceedings of the WNCG Conference*, Austin, October 2004

IEEE 802.11 Standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.-11-99, Revision 2007

Korakis T., Natayanan S., Bagri A., & Panwar S. (2006). Implementing a Cooperative MAC Protocol for Wireless LAN, *Proceedings of the IEEE International Conference on Communications (ICC'06)*, pp. 4805-4810, ISBN: 1-4244-0355-3, June 2006

Liu P., Tao Z., Lin Z., Erkip E., & Panwar S. (2006), Cooperative Wireless Communications: A Cross-Layer Approach, *IEEE Wireless Communications Magazine, Special Issue on Advances in Smart Antennas*, Vol. 13, No. 4, (August 2006) 84-92, , ISSN 1536-1284

Liu P., Tao Z., & Panwar S. (2007). CoopMAC: A Cooperative MAC for Wireless LANs, *IEEE Journal on Selected Areas on Communications*, Vol. 25, No.2, (February 2007) 340-354, ISSN 0733-8716

Morillo-Pozo J., García-Vidal J., & Pérez-Neira A. I. (2005). Collaborative ARQ in Wireless Energy-Constrained Networks, *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing 2005 (DIAL-POM'05)*, ISBN 1-59593-092-2

Morillo-Pozo J. D., & García-Vidal J. (2007). A Low Coordination Overhead C-ARQ Protocol with Frame Combining, *Proceedings of the IEEE PIMRC*, pp. 1-5, ISBN 978-1-4244-1144-3, Athens, Greece, September 2007

Nosratinia, A., Hunter, T. E., & Hedayat A. (2004). Cooperative Communications in Wireless Networks, *IEEE Communications Magazine*, Vol. 42, No. 10, (October 2004) 74-80, ISSN 0163-6804

Pfletschinger S. & Navarro M. (2008). Link Adaptation with Retransmissions for Partial Channel State Information, *Proceedings of the IEEE GLOBECOM 2008*, pp. 1-6, ISBN 978-1-4244-2324-8, New Orleans, Louisiana, USA, November 2008

Tao Z., Korakis T., Slutskiy Y., Panwar S., & Tassiulas L. (2007). Cooperation and Directionality: A Co-opdirectional MAC for Wireless Ad Hoc Networks, *Proceedings of the WiOpt 2007*, pp. 1-8, ISBN: 978-1-4244-0960-0, April 2007

Shankar S., Chou C., & Ghosh M. (2005). Cooperative Communication MAC (CMAC) – A new MAC protocol for Next Generation Wireless LANs, *Proceedings of the IEEE International Conference on Wireless Networks, Communications, and Mobile Computing 2005*, pp. 1-6, ISBN: 0-7803-9305-8, June 2005

Sadek B., Ray Liu K. J., & Ephremides A. (2006). Collaborative Multiple-Access Protocols for Wireless Networks, *Proceedings of the IEEE International Conference on Communications (ICC'06)*, pp. 4495-4500, ISBN: 1-4244-0355-3, June 2006

Wang X. & Yang C. (2005). A MAC Protocol Supporting Cooperative Diversity for Distributed Wireless Ad Hoc Networks, *Proceedings of the IEEE PIMRC*, pp. 1396-1400, ISBN: 9783800729098, Berlin, Germany, September 2005

Zimmermann M. D., Herhold P., & Fettweis G. (2004). The Impact of Cooperation on Diversity-Exploiting Protocols, *Proceedings of the 59th IEEE Vehicular Technology Conference*, pp. 410-414, ISBN 0-7803-8255-2, Milan (Italy), May 2004

Zimmermann E., Herhold P., & Fettweis G. (2005). On the Performance of Cooperative Relaying Protocols in Wireless Networks, *European Trans. on Telecommunications*, Vol. 16, No. 1, (January 2005) 5-16, ISSN 1541-8251

Zorzi M., Rao R. R., & Milstein L.B. (1997). ARQ error control for fading mobile radio channels, *IEEE Trans. on Vehicular Technology*, vol. 46, no. 2, (May 1997) 445-455, ISSN 00189545
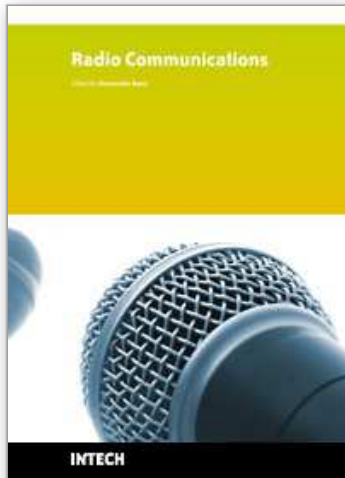
Radio Communications

Edited by Alessandro Bazzi

In the last decades the restless evolution of information and communication technologies (ICT) brought to a deep transformation of our habits. The growth of the Internet and the advances in hardware and software implementations modified our way to communicate and to share information. In this book, an overview of the major issues faced today by researchers in the field of radio communications is given through 35 high quality chapters written by specialists working in universities and research centers all over the world. Various aspects will be deeply discussed: channel modeling, beamforming, multiple antennas, cooperative networks, opportunistic scheduling, advanced admission control, handover management, systems performance assessment, routing issues in mobility conditions, localization, web security. Advanced techniques for the radio resource management will be discussed both in single and multiple radio technologies; either in infrastructure, mesh or ad hoc networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds