Experimental Performance Evaluation of a MAC protocol for Cooperative ARQ Scenarios

Ch. Verikoukis*, A. I. Pérez-Neira**, J. Alonso-Zárate*, and Ch. Skiannis***
* Telecommunications Technological Center of Catalonia (CTTC)
e-mail: {cveri, jesus.alonso}@cttc.es
** Department of Signal Theory and Communications, Technical University of Catalonia (UPC)
e-mail: anuska@tsc.upc.edu
*** University of the Aegean
e-mail: cskianis@aegean.gr

Abstract – **This paper provides details and performance evaluation of the implementation of the Persistent Relay Carrier Sensing Multiple Access (PRCSMA) protocol for Cooperative ARQ (C-ARQ) scenarios using off-the-shelf wireless cards. The underlying idea of PRCSMA is to modify the basic rules of the IEEE 802.11 MAC protocol to execute a distributed C-ARQ scheme in wireless networks in order to enhance their performance and to extend coverage.**

For the implementation, we modify the HostAP driver for off-the-shelf air-interfaces. HostAP is a Linux based driver for 802.11b WLAN cards based on Intersil´s Prism 2.5. Performance evaluation tests provide proof-of-concept of the efficiency, in terms of mean delay, of the PRCSMA in realistic conditions.

Keywords – cooperative MAC, experimental performance evaluation, cooperative ARQ, PRCSMA.

I. Introduction

Cooperative communications are advanced techniques that take advantage of spatial diversity among neighboring users in order to significantly improve the efficiency of wireless systems. The improvement induced by exploiting cooperation in wireless networks can be attained in terms of higher transmission rate, lower transmission delay, more efficient power consumption, or even increased coverage range. In such schemes, neighboring stations that otherwise will not participating in the communication process are enabled to offer some type of collaboration. The fundamental theory behind the concept of cooperation has been widely studied over the last years [1]-[2], and now is one of the most challenging and active research topics. Even though cooperative communications were originally proposed as innovative techniques at the physical layer, it is necessary to transpose them to the higher layers of the protocol stack to bring the benefits of the cooperative diversity to the end user.

The work we present in this paper is focused on a specific kind of cooperative communications: Cooperative ARQ (C-ARQ) schemes. C-ARQ schemes exploit the broadcast nature of the wireless channel in the following manner: once a station receives a data packet with errors, it can request retransmissions from any of the users which overheard the original transmission and may act as spontaneous helpers. This kind of operation increases the reliability and efficiency of the communications by exploiting either space or time diversity.

The fundamental concepts and theoretical bounds behind C-ARQ have been already analyzed in the literature [3]-[5]. However, these works consider simplified network topologies with one transmitter, one receiver, and a single relay and they assume ideal scheduling strategies.

Up to the knowledge of the authors, very few contributions can be found in upper layers of the protocol stack [6]-[9]. For example, the Persistent Relay Carrier Sensing Multiple Access (PRCSMA) protocol was presented in [9] as an extension of the Medium Access Control (MAC) protocol of the IEEE 802.11 Standard to efficiently operate in C-ARQ schemes. However, most of these works are based on either theoretical analysis or computer simulation, while significant less effort has been dedicated to their implementation and testing. Korakis et al. in [10] present a first approach to implement a cooperative scheme in actual hardware by integrating the CoopMAC protocol in a testbed using the Linux Host AP wireless driver. That paper describes the assumptions and the implementation process that the authors followed in order to implement the protocol. They also provide some experimental results using 3 nodes with TCP traffic, showing that CoopMAC outperforms the MAC protocol of the 802.11 Standard. In more recent works, CoopMAC has been also tested both for UDP and TCP traffic in a medium size testbed (10 nodes) [11]. Makda [12] studied and implemented in the aforementioned testbed security implications with CoopMAC. Likewise, Bletsas in [13] implemented some of his previous proposed theoretic algorithms for cooperative diversity in the context of a wireless sensor network testbed by using low cost embedded Software Defined Radio (SDR). In a more recent work [14], a first attempt to implement packet combining schemes for off-the-shelf wireless sensor is presented. The authors propose, implement, and evaluate two packet combining schemes for cooperative communications using Tmote Sky motes based on the 802.15.4 Standard.

Previous works are either focused on cooperative communications schemes for Wireless Sensor Networks or implement Cooperative MAC protocols of other cooperative strategies different to C-ARQ. Therefore, we contribute in this paper with the proof-of-concept of a novel MAC protocol, the PRCSMA, within the context of a C-ARQ scheme. More

This work has been supported by LOOP (FIT-330215-2007-8), PERSEO (TEC2006-10459/TCM)

978-1-4244-4148-8/09/$25.00 ©2009

specifically, we report the experience gained along the implementation of PRCSMA in a testbed by modifying the Linux wireless driver of Host AP. Host AP is a driver for 802.11 wireless interfaces based on Intersil´s Prism 2/2.5/3 chipset that has been widely used in experimental research as it supports access functionality in the software and it allows some modifications in the MAC protocol. It has to be noted that no modifications can be done at the physical layer of these devices. We also present comprehensive performance evaluation results for different type of experiments.

The rest of the paper is organized as follows. We first provide an overview of PRCSMA in Section II. Then, we discuss the different possibilities to build testbeds for MAC protocols in Section III. In Section IV we report the experience gained along the implementation process, describing the main challenges and the solutions adopted. The main results of the performance evaluation carried out with the testbed are presented in Section V. Finally, Section VI concludes the paper and outlines future work.

II. PRCSMA Overview

PRCSMA is a MAC protocol for wireless networks designed to coordinate the retransmissions of the relays in a C-ARQ scheme [9]. The main idea is that whenever a destination receives a packet with errors, it requests retransmissions from any of the relays which overheard the original transmission from the source. Then, the destination may be able to reconstruct the original packet by combining the different received copies using either Maximum Ratio Combining or Majority Voting, among other possibilities. Therefore, any station which overhears a transmission from a source station becomes a potential helper to assist any station which receives a packet with errors.

PRCSMA is essentially based on the IEEE 802.11 Standard. The whole scheme with PRCSMA works as follows: all the stations listen to all the ongoing transmissions in order to cooperate in case it is required. They keep a copy of any overheard data packet until either it is acknowledged by the destination or a certain timeout expires (this copy might be stored in a dedicated queue or in the own data buffer). Any destination which receives a data packet with errors initiates a cooperative phase by broadcasting a Call for Cooperation (CFC) control packet. Note that the error detection could be implemented, for example, by adding a Cyclic Redundancy Code (CRC) to all data packets. The CFC is transmitted after a SIFS while regular data transmission in IEEE 802.11 is executed after a longer silence period (DIFS). As a consequence, cooperation processes get priority over regular data traffic. All the stations which receive the CFC and also received the original transmission from the source are potential relays. Those which fulfill some predefined relay selection criteria, not defined in the basic description of PRCSMA, become active relays and get ready to forward their information. It is worth noting that any relay selection criteria could be attached to the CFC packet. Regardless of the specific PHY cooperative strategies applied by the relays and the reconstructing mechanism implemented at the destination station, which are not within the scope of the basic definition of PRCSMA, the cooperative information gets the form of a data packet, henceforth referred to as cooperative packet.

Within a cooperative phase, every relay attempts to get access to the channel to forward the cooperative information persistently and following the basic rules of the IEEE 802.11 Standard. Therefore, during a cooperation phase, the network is set into saturation conditions with a certain number of stations (active relays) attempting to transmit data at the same time until the cooperation phase is finished. Accordingly, and in order to avoid a certain collision upon cooperation request, all the active relays initiate a random backoff deferral period before attempting to retransmit for the first time. To do so, they independently initiate their respective backoff counters to a random value within the interval [0, CW], where CW is the size of the initial contention window.

All the relays which already have a non-zero backoff counter upon the reception of the CFC packet use their current backoff counter value instead of resetting it to a new random value. The cooperation phase is completed whenever the destination station sends a positive (ACK) or negative (NACK) packet indicating the output of the decoding process.

In the light of the implementation of the protocol, it has to be taken into account that:

- There is no ACK associated to each retransmission in order to reduce overhead.
- The persistent behavior of the relays eliminates the possibility that the destination does not receive the required amount of retransmissions, as long as there is at least one active relay.
- The relays can execute either the basic access or the collision avoidance (COLAV) access mode (with RTS/CTS handshake) for a cooperation phase.

III. Types of Networking Testbeds

Two different approaches may be used to test MAC protocols in realistic wireless environments: *i)* off-the-shelf air-interfaces with open source wireless drivers or *ii)* SDR techniques.

In the first case, the wireless interfaces of the experimental nodes are commercial wireless cards based on the IEEE 802.11 Standard. However, the cards are bound with the operating system by open source Linux based wireless drivers. By partially modifying the functionality of the MAC layer, new protocols can be tested. It has to be noted that the implemented protocols must have a high degree of backwards compatibility with the legacy one and cannot be fundamentally different. This is because the physical layer of the cards and the lower part of the MAC layer cannot be modified as they are implemented as part of the firmware. In any case, such kind of solution offers the possibility to easily compare new protocols with the commercial 802.11 solutions.

On the other hand, with the SDR approach, software and

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

hardware can be combined to provide an efficient technique for building and testing wireless communication systems. The software executes the signal processing operations while the hardware is not modulation specific. SDR techniques offer flexibility in the design of the protocols and provide certain degree of reconfigurability. In that sense, the physical and the MAC layers can be built from scratch. This makes these techniques particularly suitable for experimentation with protocols that are fundamentally different to the 802.11 Standard, as well as for the experimentation with novel PHY-MAC cross-layer designs, cooperative schemes, or advanced coding techniques. However, implementing any new technique from scratch in an SDR is a very time-consuming task and can be very challenging due to the specific hardware limitations. Moreover, in order to compare the proposed solutions with the legacy ones (e.g., 802.11 MAC protocol), these protocols have to be implemented in the same hardware platform.

There exist some toolkits for designing SDR platforms. The GNU radio [15] is arguably the most popular toolkit and it helps in bridging the hardware with signal processing modules for different kinds of protocols. The WARP [16] solution at Rice University is an FPGA-based approach that offers four radio interfaces per node and it is mainly focused on PHY/MAC layer implementations.

Since PRCSMA is strongly based on the IEEE 802.11, we decided to go for the first alternative. In the next section, we describe the details of the implementation.

IV. Implementation

A. Testbed Platform

The experiment has been realized within the framework of the Self-managed Access Laboratory (SAL) testbed at the CTTC premises. SAL is a cluster of multiple PC boards equipped with basic components such as CPU, RAM memory, Ethernet interface, and both USB and PCI ports. The PCs communicate with each other through either a real or emulated channel and they are managed and interconnected through a centralized control bus based on Ethernet.

The SAL testbed is based on the Parallel-Knoppix customization of Linux Debian. Knoppix is a free-software initiative that aims at encapsulating the latest version of the Linux Debian distribution as a liveCD. This feature allows each PC of the platform to be booted up using any external memory, such as an USB device.

According to the operation of SAL, we first load the Operating System (OS) in one of the PCs of the cluster. This PC becomes the master and transfers the OS, through Ethernet, to the other PCs of the cluster, which become slaves. Therefore, all the PCs operate with the same drivers and kernel version. This characteristic is an asset when an experiment with many nodes needs to be tested.

The SAL testbed is structured in five layers with increasing level of abstraction, as shown in Figure 1. The first (lower) layer is the ***hardware manager*** and consists of a hardware database server responsible for maintaining the state of the available processors and network resources updated. The second layer contains the ***operating system manager*** that provides the distributed cluster of PCs (one master and several slaves) with low-latency inter-node communications using Message Passing Interface (MPI). The third layer is the ***scenario manager*** which is responsible for distributing, synchronizing, and monitoring the required testbed scenarios through a series of agents. The forth layer is the ***project manager*** which defines the Integrated Developer Environment (IDE) components. This layer converts abstract SAL user instructions to the necessary commands of the testbed. Finally, the fifth (upper) layer is the ***technology manager*** which defines the interfaces that the end-user of SAL uses to manage the hardware.

SAL is not related to any wireless technology in particular and thus it allows us to conduct experimental research of multiple technologies simultaneously. Any technology implementable in a wireless card with Linux drivers can be supported by SAL. This key feature provides SAL with more flexibility compared to other testbeds aimed to the experimentation of a particular technology. In addition, with the appropriate off-the-shelf radio interface, SAL nodes can implement SDR algorithms such as those described in the GNU Radio project.

In the next sections, we describe the implementation of PRCSMA in SAL.

B. Implementation Overview

Our testbed consists of a network formed by a number of wireless stations where a single source, a single destination, a monitor station, and a number of relays can be predefined for each experiment. In our experiments, the source generates UDP packets with a constant rate and the monitor captures the traffic (data and control packets) in the network.

We focus the interest on measuring the average delay perceived by the source when it transmits a data packet to the destination with the assistance of a given number of relays. This delay is defined as the time elapsed from the moment a packet is ready for transmission, until the destination is able to acknowledge the proper reception of the packet without errors. The relays have to contend to get access to the channel executing PRCSMA at the MAC layer in order to retransmit a predefined number of copies of the original packet.

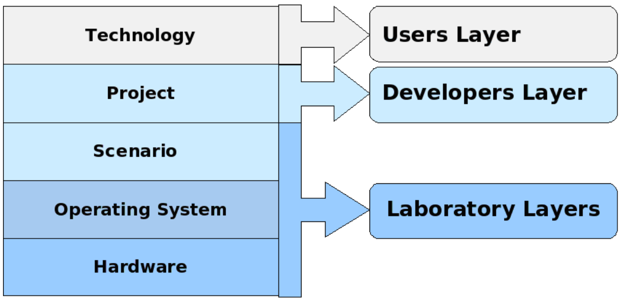

Figure 1 SAL Testbed Structure

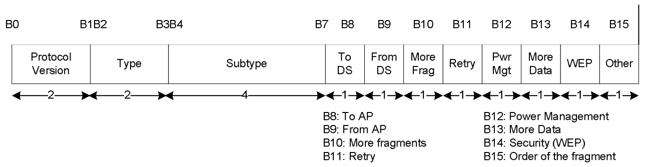

Figure 2 Control Frame of a MAC Header

C. Assumptions

For the execution of each experiment, some assumptions and simplifications have been made to complete the first implementation of PRCSMA:

- All the stations execute the basic access mode of the 802.11 (without RTS/CTS handshake).
- Since we have no access to the physical layer of the wireless card, we do not implement any diversity or combining mechanisms like Majority Voting or Maximum Ratio Combining. As a consequence, we measure in our experiment the time required to receive a predefined number of copies of the original packet.
- The number or relays is fixed and known.
- Time-sensitive tasks such as the transmission of control packets (RTS, CTS, and ACK), operate in the lower module of the wireless card as a part of the firmware. Since we cannot modify this part, CFC and ACK packets indicating the beginning and end of a cooperative phase, respectively, are implemented by using modified data packets.
- For the same reason, we cannot suppress the transmission of ACKs after a correct reception of each retransmission. Accordingly, the time spent in the transmissions of ACK packets is explicitly subtracted at the end of each experiment for the calculation of the actual transmission delay that would be perceived in a real network.

D. Description of the Implementation

In this section we describe some of the key challenges found out along the implementation of the protocol in the available drivers. The solutions adopted in each case are also discussed.

- **Identification and reception of the overheard packets**

The first problem we faced was related to the ability of the relays to receive and store packets in their buffers with a MAC address in the destination field of the packet different to their own address. By default, each card passes to the higher layers only those packets that have the same MAC address in the destination field. Otherwise, the driver drops the packets and, as a consequence, we were not able to use these packets for the cooperative phase.

In order to overcome this problem, we configured the wireless cards in promiscuous mode. This mode permits the reception of packets with different MAC addresses and passes them to the higher layers of the card. Although the monitor mode also allows for this functionality, it does not allow for the transmission of packets. Therefore, the promiscuous mode was the choice.

However, this promiscuous operation brought two more challenges. First, we had to apply a type of filtering based on the Basic Service Set Identifier (BSSID) of the received packets to guarantee that the received packets belonged to our network. Otherwise all the received packets would be stored by the nodes, regardless of the BSSID of origin.

Second, both UDP and TCP packets have the same type and subtype in the Frame Control (Figure 2), and thus it was necessary to apply some sort of filtering using information related to higher layers to subtract any TCP packet received from any other application. We decided to filter them by the field *protocol* of the IP header. Since we used UDP packets, this field has the value 17. The same type of filter may be easily adapted to TCP packets (value 6).

- **CFC Generation**

Since it is not possible to create a new type of control packet, CFC packets have been implemented by modifying data packets. As a consequence, the receiver would be unable to identify them. In order to sort them out from conventional data packets we modified one of the reserved fields of the subtype field of the MAC header of data packets.

The main drawback of this approach is that, since CFC packets are data packets, they have to compete for the channel. As a consequence, a CFC packet may not be transmitted immediately after the reception of the packet of interest (the one received with errors) since any other node may transmit a data packet. Therefore, it was necessary to devise a method to define for which data packet the CFC requests the retransmission. In order to solve this problem we attached the sequence number of the erroneous packet and the MAC address of the transmitting device to the header of the CFC, so that it was possible for the relays to identify the requested packet.

- **Transmission of a cooperative packet**

Whenever a relay receives a CFC, it has to verify that the requested packet is stored in its buffer. Then, it should retransmit an exact copy of the received packet to the destination. Ideally, the destination would send an ACK to the transmitter of this packet and not to the relay. However, we faced two major problems when forcing the ACK to be transmitted to the source and not the relay due to the limitations to get access to the firmware:

1) If a relay does not receive an ACK from the receiver it considers that its retransmission had failed and enters into backoff mode. Therefore, the backoff window is doubled up. When the backoff time expires, the relay will try to retransmit the packet with, in some cases, a lower transmission rate due to the adaptive rate capability of the firmware for previously failed transmissions.

2) During the retransmission, the firmware of the card automatically changes the sequence number of the packet. Therefore, any ACKs to the original destination that has different sequence number will be incoherent.

In order to face these problems, we substituted the transmitter address of the MAC header of the cooperative packets by the MAC address of the relay. Accordingly, the relays receive an

ACK for every successful transmission. Then, and for the computation of statistics, we explicitly subtract the time required for the transmission of these ACKs offline. Furthermore, in order to accomplish with the PRCSMA rules, we created a final ACK packet (to indicate the end of the cooperation phase) using a data packet. This type of ACK should compete for the channel and as a consequence may add some additional delay to the whole cooperation phase.

- **Identification of the cooperative packets**

Since the CFC packets have been created using a normal data packet, it was impossible to reserve the channel for the cooperation phase. Therefore, one of the problems we faced was the identification of the cooperative packets over other regular data packets. A possible solution would consist in identifying the address of the transmitter that appears in the MAC header, but this solution is not possible based on what we previously described due to the modification of this header. In order to keep the related overhead as low as possible we decided to modify the subtype field with a reserved one for these packets. Giving this field a unique number allowed us to identify these packets.

V. Performance Evaluation

A. *Platform Configuration*

For the experiments presented in this paper, each of the PCs in the SAL testbed was equipped with an INTEL Pentium D Processor 945+, at 3.40 GHz, 1 GB of RAM, and a wireless card bound with a Linux Wireless driver Host AP in Prism 2.5 chipset. All the nodes of SAL were configured to use channel 1 of the 2.4GHz ISM band. For the duration of the experiments, there were uncontrolled neighbor wireless networks in the channels 4, 5, and 6 of the ISM band that could interfere with our transmissions.

B. *Description of the Scenario*

In our experiments we used a set up of a variable number of relays with a source and destination. We ran different experiments changing the number of relays, the packet length, and the relays access method.

We repeated each experiment 30 times and we depicted plots with the average and the standard deviation values of the delay that the receiver perceives since the moment it transmits a CFC until it gets a pre-defined number of cooperative packets using a fixed number of active relays. As it has been previously explained, due to the limitations to get access to the firmware of the wireless cards, we did not perform any physical layer operation to combine the received packets.

The statistics generated in the experiment were measured with WinShark. We generated UDP packets that were transmitted in constant-duration time frames of 2 seconds. For every received packet from the transmitter, the receiver broadcasts a CFC and a cooperation phase is initiated. As it is a real experiment and some channel errors may occur in any transmissions, we have included in the plots the value of the standard deviation.

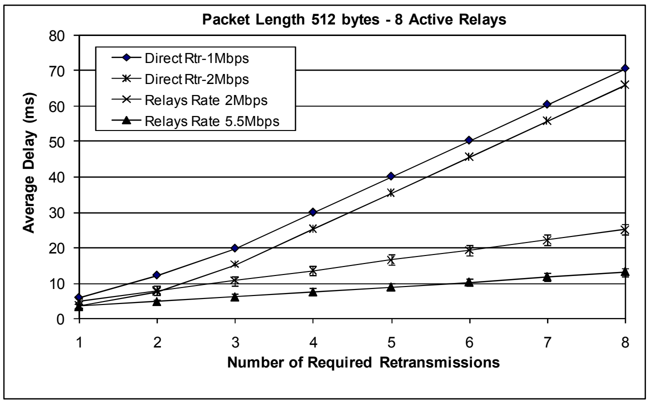

Figure 3 Average Delay vs. Number of Required Retransmissions

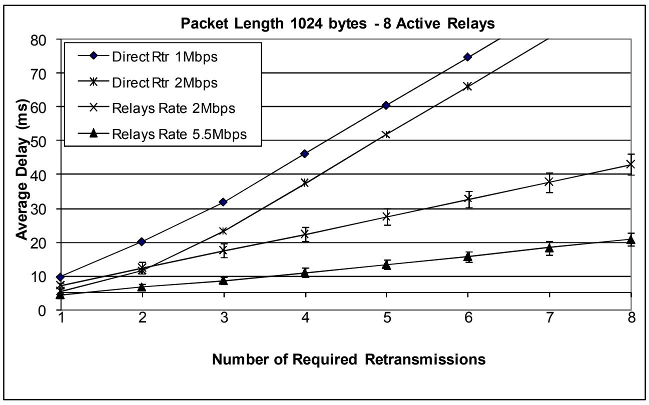

Figure 4 Average Delay vs. Number of Required Retransmissions

C. *Results*

We first compare the average transmission delay of a non-cooperative ARQ with a C-ARQ scheme. In the former case, retransmissions are performed directly from the original source at low data rates (1 or 2 Mbps) while in the later case all active relays retransmit the packets at fixed rate of either 2 or 5.5 Mbps, respectively, but being the same for all the relays. In both cases the rules of the 802.11 Standard are executed at the MAC layer. As discussed before, the time devoted to ACKs from the destination to the relays has been subtracted offline. The obtained results for a network with 8 active relays are depicted in Figure 3 and Figure 4 for data packets of 512 and 1024 bytes, respectively. It can be seen in these figures that the mean delay in the C-ARQ case is significantly lower than in the non-cooperative cases. However, for low number of required retransmissions and when both the transmission rates from the source and from the relays are the same, retransmissions from the source perform better. This is basically due to the overhead that the cooperative mechanism introduces in the whole process. These overheads include the time needed to broadcast the CFC packet and the wasted time due to both collisions and backoff deferral periods among the relays.

In any case, when the transmission rate from the relays is higher than the original transmission rate from the source, cooperation is always preferable. The gain from the coopera-

978-1-4244-4148-8/09/$25.00 ©2009

tion increases with the number of required retransmissions. It can also be seen from Figure 3 and Figure 4 that the gain is higher for smaller packet lengths which are more robust against channel errors.

So far, we have considered that all the relays had the same (high) data transmission rate. This can be seen as an ideal scenario. A more realistic scenario may include relays that adapt the rate of the cooperative packets according to the received signal strength from the CFC packet. Therefore, in the next experiment we consider a scenario wherein not all the relays may be able to transmit at the same maximum rate. In order to be aware of the exact transmission rates of the packets we evaluated a multirate scenario with 10 active relays: 5 of them retransmit the cooperative packets at 5.5 Mbps and the rest retransmit the cooperative packet at 2 Mbps.

The results are shown in Figure 5, where the average delay of the multirate scenario is compared to that of the first single rate experiment with 10 active relays transmitting at a common fixed rate. It can be seen that the average transmission delay of the multirate experiment for low number of required retransmissions (<5) shows similar performance to the single rate scenario with the lowest transmission rate (2 Mbps). However, for higher number of required retransmissions, the average delay performance lies in between the two curves with the single rate experiments, attaining values similar to the lower bounds of the standard deviation.

The results of the multirate configuration show that the relays at 2 Mbps are more prone to seize the channel than the relays at 5.5 Mbps. This can be verified by looking at the different slopes of the two curves for number of required retransmissions higher than 4. Moreover, the bigger values of standard deviation underline the random distribution of the transmissions order.

The results of the second experiment can be justified with the conclusions presented in [18] where the authors demonstrate that the presence of a low rate node in a network directly affects the performance of the rest of the nodes of the network in long term. Therefore, efficient relay selection criteria should be designed to improve the overall performance.

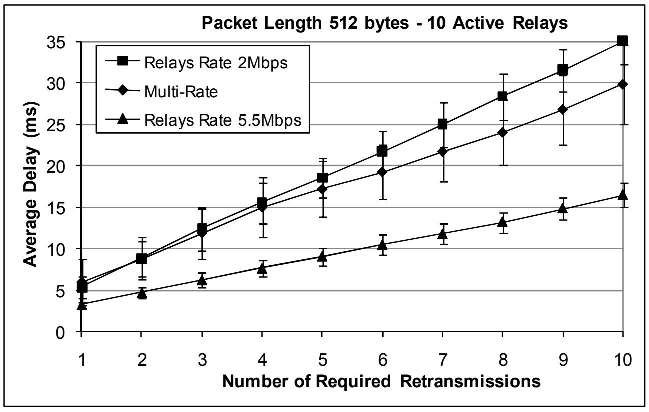

Figure 5 Average Delay vs. Number of Required Retransmissions

VI. CONCLUSIONS

In this paper we have described our experience when implementing the PRCSMA protocol for C-ARQ in a hardware testbed. We have presented all the challenges faced along the implementation process and we have described the different solutions adopted in each case. In addition, several experiments have been carried out to validate the testbed and to experimentally evaluate the performance of the protocol in a practical network. We have presented in this paper the most relevant results. Future work includes the design of simple relay selection mechanisms to further improve the efficiency of cooperation and the further analysis of more advanced scenarios for multi-radio topologies.

REFERENCES

[1] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior,” IEEE Trans. on Information Theory, vol. 50, no. 12, Dec. 2004.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, “User Cooperation Diversity – part I: system description,” IEEE Transactions on Communications, vol. 51, no. 11, pp. 1927-1938, Nov. 2003.

[3] E. Zimmermann, P. Herhold, and G. Fettweiss, “On the performance of cooperative relaying protocols in wireless networks,” European Trans. On Telecommunications, vol. 16, no. 1, pp. 5-16, Jan. 2005.

[4] P. Gupta, I. Cerruti, and A. Fumagalli, “Three transmission scheduling policies for a cooperative ARQ protocol in radio networks,” in proc. of the WNGG, Oct. 2004.

[5] I. Cerruti, P. Gupta, and A. Fumagalli, “Delay Models of Single-Source Single-Relay Cooperative ARQ protocols in Slotted Radio Networks with Poisson Frame Arrival,” in proc. of the INFOCOM 2007.

[6] P. Liu, Z. Tao and S. Panwar “A Cooperative MAC Protocol for Wireless Loacal Area Networks,” in Proc of the IEEE ICC 2005.

[7] J. Morillo-Pozo, J. García-Vidal, and A. I. Pérez-Neira, “Collaborative ARQ in Wireless Energy-Constrained Networks,” in proc. of the DIALPOM’05.

[8] N. Johansson, P. Larsson and M.Mayer, “Data packet forwarding method for multi-hop system, involves selecting receiving node based on data quality indicator representing degree of decodability of unsuccessfully decoded data packet, received from one of nodes,” Patent no. WO2006085801-(A1), Nov. 2007.

[9] J. Alonso-Zárate, E. Kartsakli, Ch. Verikoukis, and L.Alonso, “Persistent RCSMA: A MAC Protocol for a Distributed Cooperative ARQ Scheme in Wireless Networks,” EURASIP Signal Processing Magazine, vol. 2008, article ID 817401, pp. 13, May 2008.

[10] T. Korakis, S. Narayanan, A. Bagri, and S. Panwar, “Implementing a Cooperative MAC Protocol for Wireless LANs,” in proc. of the IEEE ICC 2006, Turkey, Jul. 2006.

[11] T. Korakis, Z. Tao, S. Makda, B. Gitelman, S. Panwar, “To Serve is to Receive – Implications of Cooperation in a Real Environment,” in proc. of Networking 2007, Atlanta, Georgia, USA, May 2007

[12] S. Makda, A. Choudhary, N. Raman, T. Korakis, Z. Tao, and S. Panwar, “Security Implications of Cooperative Communications in Wireless Networks,” in proc. of IEEE Sarnoff 2008, Princeton, Apr. 2008.

[13] A. Bletsas and A. Lippman, “Implementing Cooperative Diversity Antenna Arrays with Commodity Hardware,” IEEE Communications Magazine, vol. 44, no. 12, pp. 33-40, Dec. 2006.

[14] D. O'Rourke and C. Brennan, “A Practical Implementation of an Improved Packet Combining Scheme for Wireless Sensor Networks,” in proc. of the ICC 2008 Workshops.

[15] http://www.gnu.org/software/gnuradio/

[16] http://warp.rice.edu/

[17] IEEE, Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications, IEEE Std. 802.-11-99, Aug. 1999.

[18] M. Heusse, F. Rousseau, G. Berger-Sabatel, and A. Duda “Performance Anomaly of 802.11b,” in proc. of the INFOCOM 2003.