An Automatic Repeat Request Protocol for Cooperative Slotted Radio Networks

Isabella Cerutti, Andrea Fumagalli, and Puja Gupta

Technical Report UTD/EE-03/2004
July 2004

An Automatic Repeat Request Protocol for Cooperative Slotted Radio Networks *

Isabella Cerutti, Andrea Fumagalli, and Puja Gupta
OpNeAR Lab.
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
Email: {isabella,andreaf,pkg021000}@utdallas.edu

July 14, 2004

Abstract

In conventional (non-cooperative) radio networks, the data frames, that are corrupted by errors during the transmission in a fading environment, are retransmitted by the source. In cooperative radio networks, frame retransmission may be performed by a neighboring node that has successfully overheard the source's frame transmission. The advantage of the latter is the spatial diversity that is provided by the cooperative node. In addition, the retransmitted frame may have incremental redundancy for improved performance.

In this paper a simple automatic repeat request (ARQ) protocol is specified that takes advantage of coded cooperative communication in slotted, single-hop radio networks. An exact analytical formulation is presented to compute the delay experienced by Poisson arriving frames whose retransmission (when needed) is performed by one cooperative node. The study reports significant quantitative advantages of the coded cooperation ARQ protocol in terms of both throughput and latency, when compared to non-cooperative ARQ protocols.

1 Introduction

Wireless networks are reaching a widespread diffusion, thanks to a variety of solutions — e.g., cellular, ad-hoc, wireless LAN, and sensor networks — that are increasingly deployed in the field. The main drive behind this expansion is the virtually endless applications that come with it. They cover both commercial and military needs, including security, medical monitoring, machine diagnosis, chemical and biological detection [1].

One of the most peculiar characteristics of the radio medium is its inherent broadcast nature. When a source transmits, beside the intended destination, other nodes within earshot may receive the transmitted signal. Depending on various factors, the signal received at the nodes within earshot may be affected by different amount of fading, power attenuation and noise. In single-hop networks, this characteristic is traditionally treated as *interference* at the physical layer and as *collision* at the MAC layer. Both may prevent correct reception at the destination.

Reliable data delivery over a single-hop radio channel requires the retransmission of the *data frame* until its reception is successful at the destination. Traditional Automatic Repeat Request (ARQ) protocols specify the frame exchange that takes place between the source and destination to ensure correct and in-order data delivery to the destination [2]. As neighboring nodes do not actively participate and cooperate in the frame exchange, these ARQ protocols are here referred to as *non-cooperative*.

Cooperative ARQ protocols permit nodes, other than the source and destination, to actively help deliver the data frame to the destination correctly. The rationale is that nodes which are within earshot from the source and the destination may cooperate by making use of the received interference to improve the overall capacity of the source-destination wireless channel. In simple terms, when the source's data frame transmission is not successful, a neighboring node (referred to as the *relay*) that has overheard the source's transmission is invited to take part in the ARQ frame exchange. The destination can rely now on data frames that are coming from two distinct nodes — the source and the relay —

*This research was supported in part by NSF Grants No. ANI-0082085, ECS-0225528 and Italian Ministry of University (MIUR) under FIRB project "Enabling platforms for high-performance computational grids oriented to scalable virtual organizations" (contract n. RBNE01KNFP).

thus getting a better overall reception quality. The essence of the idea lies in that the destination benefits from data frames arriving via two statistically independent paths, a concept known as *spatial diversity*.

A complete survey on cooperative radio communication can be found in [10]. Initial work on cooperative communications on the Gaussian relay channel is reported in [3]. The role of the relay is solely to assist the source, i.e., *one-way cooperation*. More recent works [10, 7, 6, 12, 13, 17] have extended the concept of cooperative communications by taking into account fading of the wireless channel and allowing two nodes to cooperate with one another simultaneously, i.e., a *two-way cooperation* whereby each node both transmits its data frames and helps the other node carry out its own data frame transmissions. These works can be divided into three categories [10]. In *detect-and-forward* methods [12, 13], the relay detects and retransmits the frame whenever possible. In *amplify-and-forward* methods [7], the relay amplifies the received noisy signal and retransmits it. Both of these methods use retransmission of the exact copy of the data frame. In *coded cooperation* methods [5, 6], in contrast, cooperation is achieved in the framework of channel coding. The work in [5, 6] shows the feasibility of the coded cooperation and evaluates the benefits, in terms of reduction of bit or frame error probability, when using different codes and cooperation levels. Most of these results focus on the physical layer aspects of cooperation. Only a very recent work focuses on the related ARQ protocol aspects [17]. In this work the authors study the performance of a one-way cooperative ARQ protocol through the evaluation of the Signal-to-Noise Ratio (SNR) gain and average number of retransmissions by means of event-driven simulation.

In this paper, a simple two-way cooperative coded ARQ ($CC-ARQ$) protocol is described for the slotted wireless channel. The $CC-ARQ$ is designed for two sources that — while they are simultaneously transmitting data to a common destination, e.g., a base station — cooperate with one another. At each source, priority is always given to the retransmission of the other source's data encoded in a *incremental redundancy frame*, over its own data frames. The advantage of this policy is to limit the overall number of outstanding data frames, i.e., frames awaiting positive acknowledgment from the destination. It also minimizes the average number of data frames that must be stored at the destination to guarantee efficient and in-order delivery of the received data frames to the upper layer.

Moreover, the paper derives an analytical model to compute the expected data frame latency and number of frames stored at the source. Due to the complexity of this model derivation, the study is limited to the case whereby only one source has data frames to transmit, i.e., one-way cooperation. The model is exact when the arrival process of the data frames at the source is Poisson, and the retransmission probabilities are known.

The performance gain of the $CC-ARQ$ protocol is analyzed in the final part of the paper, when compared to two non-cooperative ARQ protocols, — i.e., conventional ARQ [2] and Hybrid-ARQ ($H-ARQ$) [8, 15]. The comparison includes saturation throughput, expected frame latency and expected queue length at the source, obtained under various scenarios of offered load, channel attenuation, and node location. The study is based on both simulation and numerical results which are obtained from the analytical model presented for the one-way cooperation case. The results indicate that the $CC-ARQ$ protocol yields improved network performance when the channel quality (SNR) between the cooperating nodes is sufficiently good. When this is not the case, cooperation may not be beneficial and non-cooperative protocols might achieve better network performance.

2 $CC-ARQ$ Protocol Description

In this section, the $CC-ARQ$ protocol is described. The network is composed of two sources that are transmitting data to a third node, the destination. The nodes represent typical wireless nodes, such as cellular phones, or sensors. Sources are provided with a buffer of unlimited capacity.

Each source transmits on a reserved orthogonal channel and is provided with the capability of "eavesdropping" on both channels. The transmissions on both the orthogonal channels are organized in time-slots. During a time-slot, each source may transmit a data frame and the destination acknowledges, positively with an `ACK` control frame, or negatively with `NAK` control frame, the reception of the data frame. The destination transmits acknowledgments (`ACK` or `NAK`) and receives data frames on both channels.

Before describing the $CC-ARQ$ protocol, few additional details on the concept of coded cooperation are discussed here. In order to take advantage of the coded cooperation, the sources can act as relays too, for one another. The source, that has a data frame to transmit, encodes the data into a codeword, which is partitioned into two frames, containing N_1 bits and N_2 bits, respectively. Partitioning can be obtained through different techniques, the one presented here is based on puncturing [4]. If the original codeword has $N_1 + N_2$ bits, it is possible to puncture this codeword down to the N_1 bits for the data frame, which itself is a valid (weaker) codeword. The remaining N_2 bits are the punctured bits

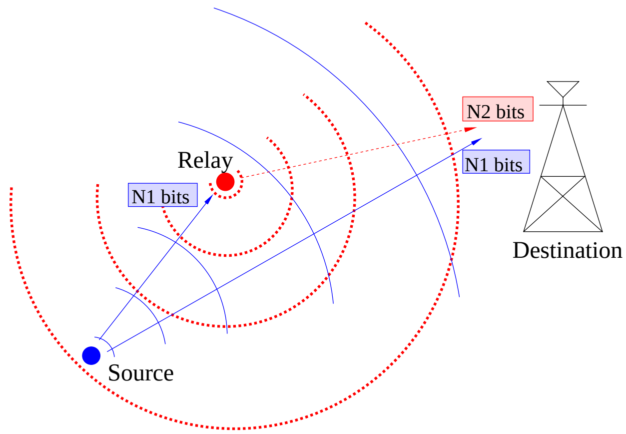

Figure 1: Coded cooperation concept

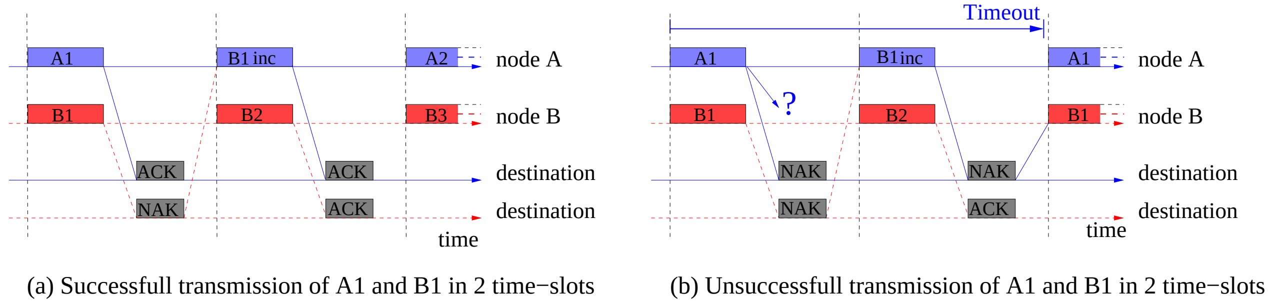

Figure 2: Time chart of the $CC - ARQ$ protocol

that form the incremental redundancy frame and can be transmitted by the relay node, as shown in Fig. 1. To be able to do so, the relay node must receive the data frame transmitted by the source correctly. The percentage of the total bits that the relay transmits for the other source, i.e., $N_2/(N_1 + N_2)$, indicates the chosen level of cooperation. In the remaining of the paper, the level of cooperation is fixed to 50% for each relay.

The version of the $CC - ARQ$ protocol that is going to be presented and studied assumes that the relay nodes act altruistically, i.e., they give priority to the incremental redundancy frames over their own data frames. Other policies are possible within the $CC - ARQ$ protocol, e.g., FCFS or egoistic cooperation (i.e., priority to data frames over incremental redundancy frames).

Four different cases need to be considered when using the two-way $CC - ARQ$ protocol. For simplicity, assume that the buffers at both nodes are empty.

1. When a data frame is generated at a relay node, the node synchronizes the transmission of the data frame. If the data frame is successfully decoded (determined by checking the CRC code) at the destination, the destination replies with an `ACK`. This is the scenario depicted in Fig. 2.a for node A. In the first time-slot, the data frame `A1` transmitted by node A is acknowledged by the destination in the same frame. In the next time-slot, node A can transmit other data frames or incremental redundancy frames for node B.

2. When the destination is not able to decode the data frame, it replies with a negative acknowledgment (`NAK`). The `NAK` is received by both nodes, thus in the next frame the other relay node is invited to cooperate. In Fig. 2.a, the data frame `B1` sent by node B in the first time-slot has been decoded successfully by node A, but not by the destination that sends a `NAK`. In the second time-slot, node A transmits the incremental redundancy frame `B1inc`, different from the original `B1`. Then the destination combines together `B1` and `B1inc` and at this time, it is able to receive successfully (`ACK`) the original data from node B. Note that node A altruistically gives priority to frame `B1inc` over the other data frames that could be stored in its own buffer.

3. Assume that the destination is not able to decode the data frame. If the other node is also unable to decode the

data frame, it cannot cooperate in the next time-slot. Thus the timeout expires and the source has to retransmit the original data frame. A good choice for the timeout duration is of two time-slots. In Fig. 2.b, the data frame A1 sent by node A in the first time-slot is NAK by the destination in the same frame. Since node B cannot cooperate, node A retransmits a new A1 in the third time-slot. With this new retransmission the four different cases can happen again until the original data frame is going to be received successfully at the destination.

4. When the destination receives a data frame and an incremental redundancy frame, but it is unable to decode successfully the combined frames, a NAK is transmitted. Thus, a new retransmission of the original data frame from the source is required in the next time-slot. This case is shown in Fig. 2.b, where the frame B1, after being negatively acknowledged (NAK) for two consecutive time-slots, is retransmitted again in the third time-slot by the node B and the four different cases can repeat until the original data frame is going to be received successfully at the destination.

So far, the presentation of the $CC-ARQ$ protocol has assumed that only a node at a time is transmitting. Since the nodes are operating on orthogonal channels, they can transmit simultaneously without interfering. Thus the four cases mentioned above can be intercorrelated, as shown in Fig. 2 in which cases 1 and 2 are happening simultaneously in Fig. 2.a and cases 2 and 3 in Fig. 2.b.

The same protocol can be used for one-way $CC-ARQ$, with the difference that only one node is transmitting data frames, while the other can only transmit incremental redundancy frames.

Note that a buffer able to store only two data frames for each channel is required at the destination. It can be demonstrated that this is also the minimum required window size that allows continuous frame transmission at both of the sources.

3 Analytical Framework

This section describes an analytical model for the one-way $CC-ARQ$ protocol, based on a queueing system. A node, referred to as S, acts as source, while the other, referred to as R, acts as a relay. Each node is represented by a queue with slotted service time and a single server. Data frames are generated at node S following a continuous-time Poisson process with average arrival rate λ_S. The service (i.e., transmission) time is deterministic and equal to the time-slot duration T.

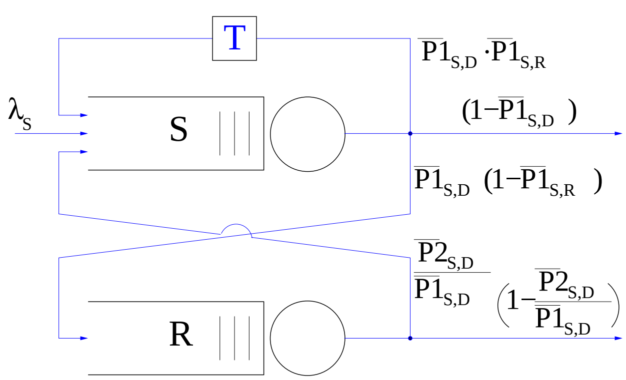

Figure 3: One-way $CC-ARQ$ protocol queueing system

Fig. 3 depicts the queueing systems. After being served, a data frame can exit from the system or being feedbacked in the same queue with a delay or move to the other queue with some fixed probability. Under the assumption that the timeout duration is two time-slots, the delay duration is one time-slot. Similarly, with some probability an incremental redundancy frame can move in the other queue or exit from the system. The probabilities to feedback or to move from one queue to another are given by the frame error probabilities. Error probabilities of ACK and NAK have been considered negligible. The meaning of the frame error probabilities represented in the figure is the following:

- $\overline{P1}_{S,j}$: average frame error probability, i.e., probability that node j (either relay node R or destination D) has detected errors in the frame sent from S;

- $\overline{P2}_{S,D}$: average error probability of combined frames, i.e., probability that the destination D has detected errors in the data frame sent from S combined with the incremental redundancy frame sent from R.

The calculation for obtaining the error probabilities for convolutional codes are reported in Section 4.1.

The objective is to evaluate the expected latency experienced by the frames entering in queue S before exiting from the system, also referred to as *sojourn time*, and the expected number of frames in queue S. In order to solve the model in Fig. 3, the queueing model with slotted time and delayed feedback is studied in the next section.

3.1 Queueing Model with Slotted Service Time and Delayed Feedback

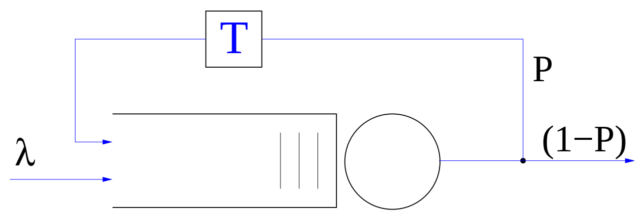

Figure 4: Queue with slotted service time and delayed feedback

Consider a system composed of a queue with slotted service time and a delay element in the feedback, as depicted in Fig. 4. In such a system, the frames, that have been transmitted (or served), repeatedly return at the end of the queue after a deterministic delay with a fixed probability P and are discharged with probability $(1 - P)$. The arrival rate is a continuous-time Poisson variable with average λ and the service time is deterministic and equal to T. It is assumed that the delay introduced by the feedback is deterministic and equal to the service time, i.e., T. The utilization factor is:

$$\rho = \frac{\lambda T}{1 - P}. \tag{1}$$

The Markov chain is ergodic if the stability condition is satisfied, i.e., $\rho < 1$.

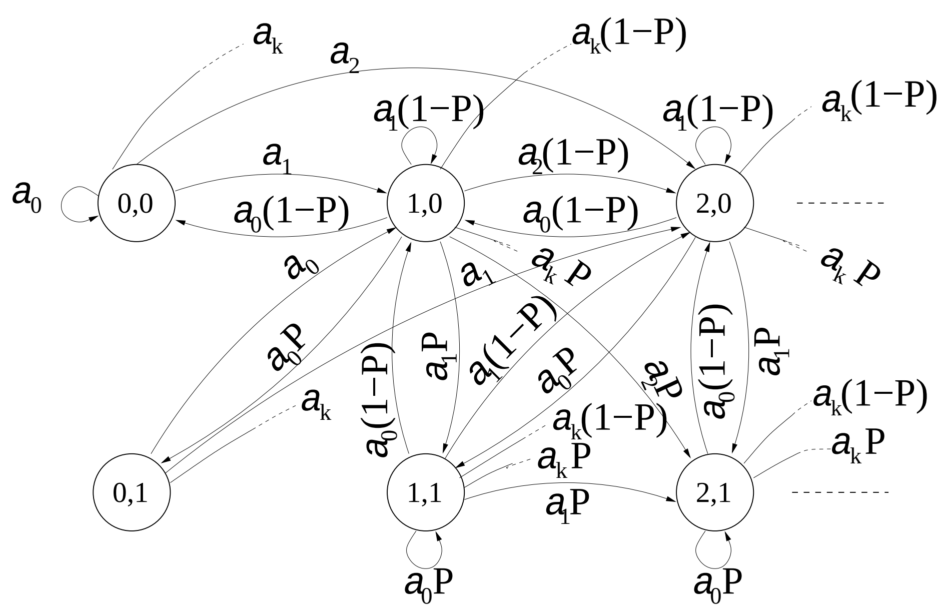

Figure 5: Markov chain for the queue with the delayed feedback

To evaluate the sojourn time in a queue with delayed feedback, it is first necessary to consider the discrete-time Markov chain associated with the process. Fig. 5 shows the discrete-time Markov chain of the system. The state $S_{i,j}$ represents the case of i frames in the queue and j frames in the delay element. Since the delay is equal to the service time, the value of j is limited to $\{0, 1\}$. The probabilities of transitions between the states are as indicated in the figure, with a_k indicating the probability of k Poisson arrivals during a time-slot, i.e., $a_k = e^{-\lambda T}(\lambda T)^k/k!$.

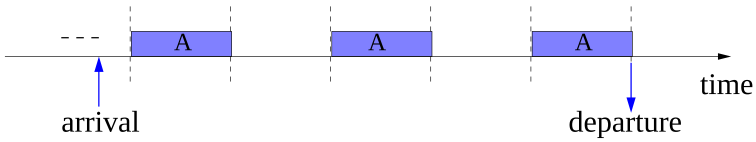

Figure 6: Frame transmissions in states $S_{0,1}$ and $S_{1,0}$

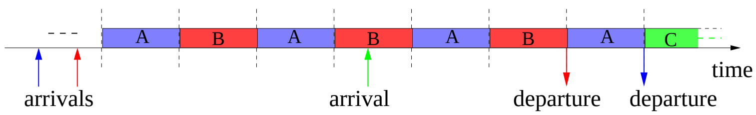

Figure 7: Frame transmissions in states $S_{i,j}$, with $i + j \geq 2$

The expected sojourn time is not affected by the service policy, thus it is possible to study the system under the assumption that the feedbacked frames have higher priority over the waiting frames (i.e., feedback at the head of the queue), without affecting the final result. By analyzing the output of the system, two distinct cases can be observed. If in the system there is only one frame, i.e., $S_{0,1}$ and $S_{1,0}$, frame transmissions are spaced by an idle time-slot (Fig. 6), during which the frame is in the delay element, If in the system there are 2 or more frames, i.e., $S_{i,j}$ with $i + j \geq 2$, in each time-slot there is a frame transmission and the transmission of two frames is alternating (i.e., one frame is transmitted while the other is in the delay element and vice-versa) until one of the two exits from the system, as shown in Fig. 7. Since the objective is the evaluation of the sojourn time, it is possible to swap the transmission without affecting the average. This would be equivalent to a system in which each frame is feedbacked directly at the head of the queue without any delay. This principle can be applied straightforwardly in the latter case (2 or more frames in

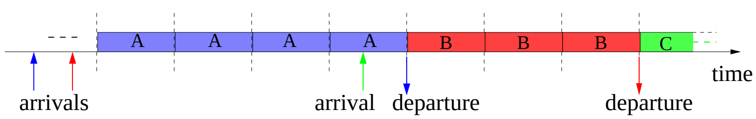

Figure 8: Frame transmissions in states S'_i, with $i \geq 2$

the system), as shown in Fig. 8, but an additional correction is required for the former case. Indeed in the former case the overall time that the frame spends in the delay element should be accounted, as shown in Fig. 9, while this is not necessary in the latter case because it is already incorporated in the waiting time of the following frame in the queue.

With these observations in mind, we would like to reduce the number of states of the Markov chain in Fig. 5 and evaluate the probability of being in one of the two above mentioned cases. By observing the Markov model, it is possible to notice that each state $S_{i-1,1}$ can be joined together with the corresponding state $S_{i,0}$, for $i \geq 2$, to form the aggregated state S'_i, i.e., i frames in the system. This is consistent with the idea of feedback directly at the head of the queue without passing through any delay element. The case for state $S_{0,1}$ is handled differently. When the system is either in state $S_{0,0}$ or $S_{0,1}$ the server is idle and their joint probability is $\pi_{0,0} + \pi_{0,1} = 1 - \rho$. State $S_{0,1}$ is aggregated with $S_{0,0}$ to form S'_0. The simplified Markov chain is represented in Fig. 10.

We are interested in solving the Markov chain of Fig. 10 and evaluating the steady state probability π'_1 of state S'_1.

Solving the equations:

$$\begin{cases} \pi'_0 = 1 - \rho \\ \pi'_0 = \pi'_0 \cdot a_0 + \pi'_1 \cdot a_0(1 - P) \end{cases} \tag{2}$$

one can find:

$$\pi'_1 = \frac{(1-\rho)(1-a_o)}{(1-P)a_o}. \tag{3}$$

With the steady state probability π'_1 it is possible to evaluate the sojourn time. Before being served, a frame entering in the system will have to wait on average half time-slot for synchronization, the residual service time of the frame

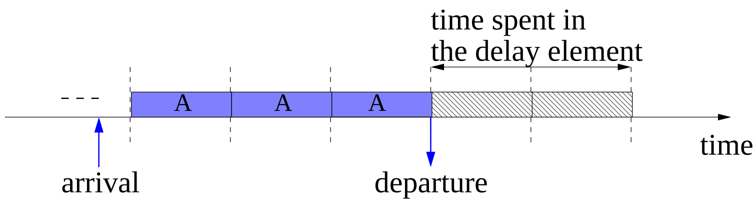

Figure 9: Frame transmissions in state S_1'

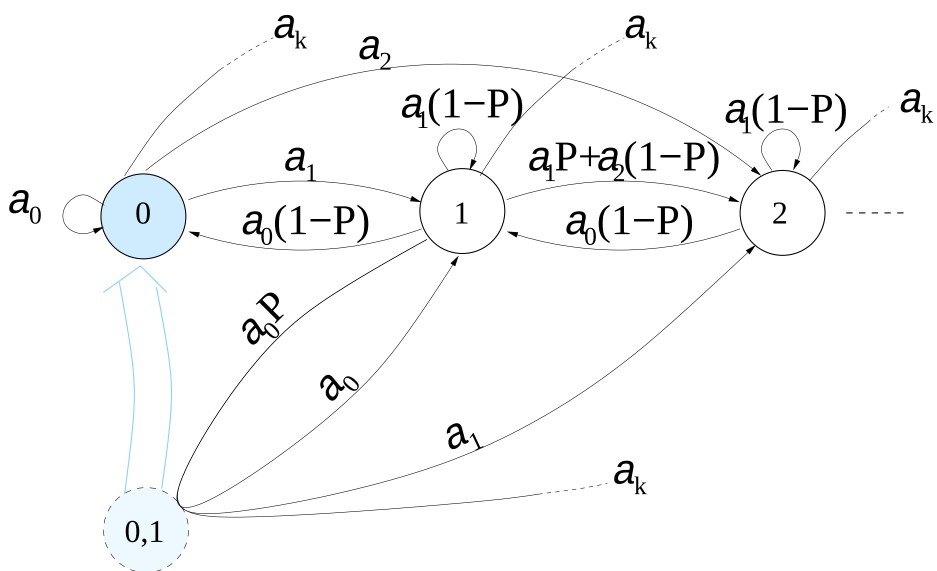

Figure 10: Markov chain for the queue feedback at the head of the queue

already in service, the residual time for the frame in the delay element to come back to service when in state S_1', and the service time for the average number of frames already waiting in queue (N_q). The expected waiting time W is:

$$W = \frac{T}{2} + \rho\frac{PT}{1-P} + \pi_1' a_0 \frac{PT}{1-P} + N_q \frac{T}{1-P}. \tag{4}$$

Using Little's theorem ($N_q = \lambda W$) and substituting Eq. 3, it is possible to derive W:

$$W = \frac{T}{2(1-\rho)} + \frac{\rho}{(1-\rho)}\frac{PT}{(1-P)} + \frac{PT}{(1-P)^2}(1-a_0). \tag{5}$$

The sojourn time T_s can be found by adding the service time to W and is:

$$T_s = W + \frac{T}{1-P} + \frac{T}{1-P}\frac{\pi_1' P a_0}{1-\pi_0'} = \frac{T}{2(1-\rho)} + \frac{\rho}{(1-\rho)}\frac{PT}{(1-P)} + \frac{T}{1-P} + \frac{P}{(1-P)^2}(1-a_0)T. \tag{6}$$

In the final expression of T_s the first three terms evaluates the sojourn time when the feedback is not delayed, while the last term accounts for the additional delay in the feedback.

The average number of frames in the queue at node S, or average queue length, seen by Poisson arrivals can be derived using Little's theorem and by removing the expected number of frames in the delay element:

$$N_q^S = \lambda W - P\rho. \tag{7}$$

The expected number of frames in the queue, or expected queue length, seen by feedbacked frames (i.e., at the time-slot start or end), N_q^s, is:

$$N_{q,sl}^S = N_q + \frac{\rho}{2}. \tag{8}$$

3.2 One-way Cooperative ARQ Protocol

The results presented in Section 3.1 can be applied to the queueing model of one-way $CC-ARQ$ protocol (Fig. 3) with arrival rate $\lambda = \lambda_S$. It is possible to consider queue R and the delay element as a unique delay element of duration T. Thus, the probability of feedback, P, is equivalent to the probability of passing through the other queue (i.e., node R cooperates) plus the probability of passing through the delay element, i.e.,

$$P = \overline{P1}_{S,D}(1-\overline{P1}_{S,R})\frac{\overline{P2}_{S,D}}{\overline{P1}_{S,D}} + \overline{P1}_{S,D}\overline{P1}_{S,R}. \tag{9}$$

The expressions for W, N_q^S, and $N_{q,sl}^S$ are given in Eqs. 5, 7, and 8, respectively. The expected frame latency, or sojourn time, requires an additional term to account for the service time of the frames that exit from the system after being served by queue R, as follows:

$$T_s = W + \frac{T}{1-P} + \overline{P1}_{S,D}(1-\overline{P1}_{S,R}) \cdot \left(1 - \frac{\overline{P2}_{S,D}}{\overline{P1}_{S,D}}\right)\frac{T}{1-P}. \tag{10}$$

3.3 Non Cooperative ARQ Protocols

The equations derived in Section 3.1 are simplified in the case of non-cooperative ARQ protocols.

For the conventional ARQ protocol, the probability of feedback is given Eq. 9, by setting $\overline{P1}_{S,R} = 1$. Unsuccessful data frames are immediately retransmitted, thus removing the delay in the feedback. If the stability condition holds, the sojourn time can be obtained from the final expression of Eq. 6 by removing the last term:

$$T_s = \frac{T}{2(1-\rho)} + \frac{\rho}{1-\rho}\frac{PT}{1-P} + \frac{T}{1-P}. \tag{11}$$

The expected queue length in node S can be obtained by applying Little's theorem:

$$N_q^S = \lambda W. \tag{12}$$

For $H-ARQ$ protocol, the probability of feedback is given by Eq. 9, by setting $\overline{P1}_{S,R} = 0$. With this value of P, if the stability condition holds, the expected sojourn time and the expected queue length at node S can be obtained from Eqs. 10 and 7 respectively.

4 Performance Evaluation

This section reports various network performance results achievable by one-way and two-way $CC-ARQ$ protocol in a number of scenarios. Results are presented after a short description of the assumptions made on the wireless channels.

4.1 Wireless Channel Characterization

Frame transmissions are assumed to be affected by path loss and flat Rayleigh fading. Rayleigh fading is assumed to be slow and constant over a single time-slot and statistically independent of the fading experienced by any other frame transmissions. The instantaneous SNR for a transmission from relay node i to node j (i.e., the other relay node or the destination) is given by:

$$\gamma_{ij} = \frac{E_{b_i}}{N_0} \cdot K \cdot l_{ij}^{\beta} \cdot \alpha_{ij}^2 \tag{13}$$

where:

- E_{b_i}: transmitted energy per bit at node i,
- N_0: noise spectral density of the Additive White Gaussian Noise (AWGN), proportional to the Boltzmann constant and the absolute temperature,
- K: path loss for an arbitrary reference distance;

- l_{ij}: distance between node i and node j;
- β: path loss exponent;
- α_{ij}: a Rayleigh distributed random variable used to model the Rayleigh fading magnitude between source i and destination j, $E[\alpha_{ij}^2] = 1\ \forall i$.

On the channel from any source to destination, the probability of receiving a frame incorrectly (error probability) is a function of both γ_{ij} and the code used to add redundancy to the transmitted data. In this paper, the relay nodes are assumed to decode data using a rate-compatible punctured convolutional code (RCPC) [4]. The error probability of a data frame, encoded with a convolutional code, sent from source i to destination j can be evaluated using the union bound technique [9, 6]. The union bound technique conservatively evaluates the error probability as follows:

$$P1_{i,j} \leq 1 - \left(1 - \min\left\{1, \sum_{d=d_f}^{\infty} \mathrm{a}_d \cdot P(d|\gamma_{ij})\right\}\right)^B, \tag{14}$$

where:

- B: number of data and CRC bits in each data frame, i.e., number of trellis branches in the codeword,
- d_f: free distance of the code [16],
- a_d: spectrum of the code [4], i.e., number of codewords of weight d,
- $P\left(d|\gamma_{ij}\right)$: probability that a wrong path at distance d is selected.

The expected error probability can be obtained by averaging on the probability density function of the instantaneous SNR, $f(\gamma_{ij})$, as:

$$\overline{P1}_{i,j} \leq \int_0^{\infty} P1_{i,j} \cdot f(\gamma_{ij})\, \partial\gamma_{ij}. \tag{15}$$

It is assumed that binary PSK with soft decoding is employed, in which case

$$P\left(d|\gamma_{ij}\right) = Q\left(\sqrt{2 \cdot d \cdot \gamma_{ij}}\right) \tag{16}$$

where $Q(\cdot)$ is the Marcum Q function [11] and d is the weight of the codeword.

When the destination D combines the frame transmitted from a source i with the incremental redundancy frame sent by the relay j (cases 2 and 4 of $CC-ARQ$ protocol in Section 2), the error probability for a given instantaneous SNR and the average error probability can be upper bounded as follows:

$$P2_{i,D} \leq 1 - \left(1 - \min\left\{1, \sum_{d_i=d_f}^{\infty} \sum_{d_j=d_{f2}-d_i}^{\infty} \mathrm{a}_{d_i,d_j} \cdot P\left(d_i + d_j|\gamma_{iD}, \gamma_{jD}\right)\right\}\right)^B \tag{17}$$

$$\overline{P2}_{i,D} \leq \int_0^{\infty}\int_0^{\infty} P2_{i,D} \cdot f(\gamma_{iD})\, f(\gamma_{jD})\, \partial\gamma_{iD}\, \partial\gamma_{jD} \tag{18}$$

where

- d_{f2}: free distance of the parent code [16];
- a_{d_i,d_j}: spectrum of the code, i.e., number of codewords of weight d_i in the first N_1 bits and weight d_j in the other N_2 bits;
- $P\left(d_i + d_j|\gamma_{iD}, \gamma_{jD}\right)$: probability that a wrong path at distance $d_i + d_j$ is selected, which is:

$$P\left(d_i + d_j|\gamma_{iD}, \gamma_{jD}\right) = Q\left(\sqrt{2d_i\gamma_{iD} + 2d_j\gamma_{jD}}\right). \tag{19}$$

4.2 Numerical Results

Numerical results have been obtained under the assumption that the wireless channel is affected by a path loss of $K = 60$ dB and that the path loss exponent is $\beta = -4$. Frames have fixed size and carry $B = 128$ bits of data and CRC, that are encoded into 256 bit codewords using a rate-compatible punctured convolutional code (RCPC) with rate 1/2, parent code rate of 1/4, puncturing period of 8, memory of 4 and generator polynomials G(23,35,27,33)(octal) [4].

Data frame arrivals follow a Poisson process. The time-slot duration has been fixed to $T = 1$. Propagation time and error probabilities of ACK and NAK have been considered negligible.

Simulation results have been obtained by running an event-driven simulator written in C++. Statistics have been collected by generating $5 \cdot 10^6$ data frames. The frame error probabilities have been evaluated using Eqs. 14 and 17 and using the instantaneous value of Rayleigh fading.

1. Validation of the Analytical Framework

The analytical results have been obtained using the framework presented in Section 3 with the average error probabilities (Eqs. 15 and 18).

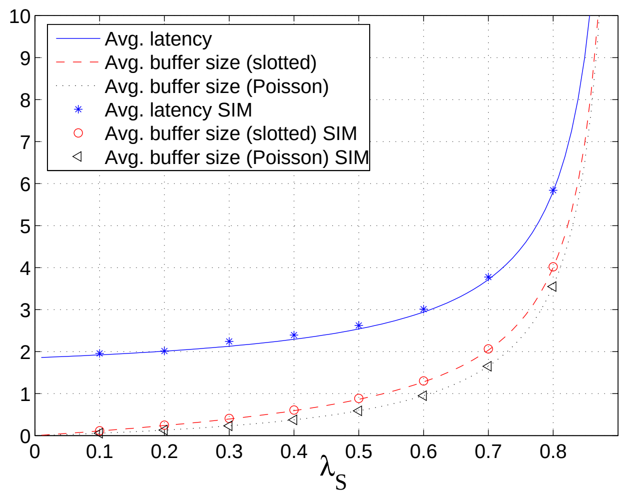

Figure 11: Expected frame latency and queue length of one-way $CC - ARQ$

Fig. 11 compares the analytical results against the simulation results for one-way $CC - ARQ$. The network parameters studied are the expected frame latency (Eq. 10) and the expected queue length as seen by Poisson arrivals (Eq. 7) and by feedbacked arrivals (Eq. 8), when the arrival rate at source S, λ_S, increases. The match of the analytical results and the simulation results confirms the validity of the model.

2. Two-way $CC - ARQ$ Performance

Figs. 12 to 17 study the effect of the positions of the two relay nodes on network parameters such as latency, queue length and throughput in two-way $CC - ARQ$. In these studies the positions of the destination D and the node $R1$ are fixed as indicated in the figures and the position of the second node, $R2$, is changed. The average received channel SNR between $R1$ and D, $E[\gamma_{R1,R2}]$, is 0 dB and that between $R2$ and D varies depending upon the position of $R2$. The arrival rate at both nodes $R1$ and $R2$ is $\lambda = 0.4$.

Fig. 12 shows the variation of the latency of frames at node $R1$ as the position of node $R2$ changes. Although the position of $R1$ is fixed the latency experienced by frames generated at $R1$ changes depending on the position of node $R2$. The latency is minimum when $R2$ is in the region that is centered around the midpoint between $R1$ and D. In this region $R1$ has to provide minimum cooperation for $R2$ and can request cooperation from $R2$. As $R2$ moves away from this region, the SNR of the channel $R2$-D deteriorates and $R2$ requires more cooperation from $R1$. In addition to that $R2$ is not in a favorable position to cooperate with $R1$ and therefore latency at $R1$ increases as it has to provide cooperation for $R2$ in addition to retransmit its own data. At some positions of $R2$, $R2$ requires cooperation from $R1$ to the extent that it causes the queue at $R1$ to become unstable. This region is marked as the ”unstable region”.

Fig. 13 shows the variation of the latency of frames at node $R2$. It is observed that when $R2$ is in the region centered around D, the channel SNR between $R2$-D is greater than that of $R1$-D. In these positions, $R2$ requires minimum cooperation from $R1$ and, at the same time, can provide cooperation for $R1$, therefore in this region data

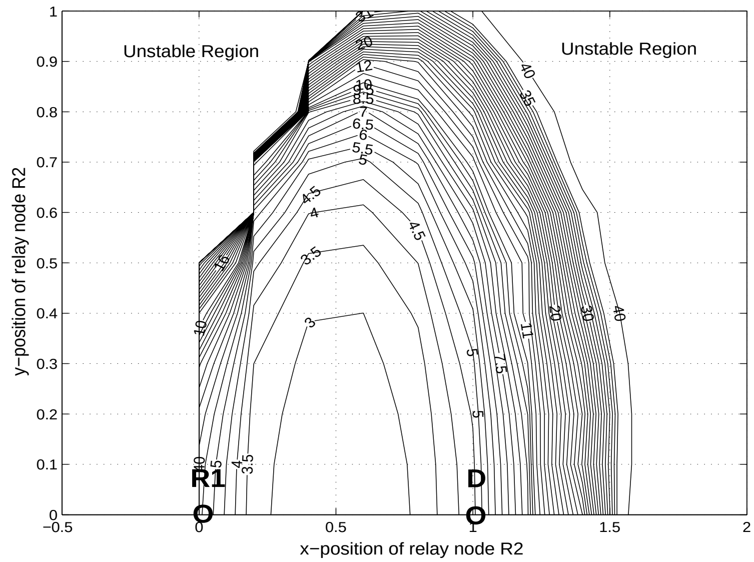

Figure 12: Expected latency of data frames generated at node $R1$ as a function of the position of node $R2$

frames generated at $R2$ will experience minimum latency. The latency of $R2$ is less than that of $R1$, implying that although $R2$ provides cooperation for $R1$, its latency is not affected significantly. As $R2$ moves away from this region, the SNR of the channel from $R2$-D reduces. Thus $R2$ has to rely on $R1$ to provide the incremental redundancy and therefore the latency experienced by the frames generated at $R2$ increases. In some positions of $R2$ the channel SNR deteriorates to an extent that several retransmissions may be required causing the queue at $R2$ to become unstable. It can also be seen from Figs. 12 and 13 that when $R2$ is positioned behind $R1$ (negative side of x-axis), the queue at $R1$ becomes unstable, however the queue at $R2$ is still stable. This happens because the queue at $R2$ is stable only due to cooperation from $R1$, i.e., by cooperating $R1$ is increasing the latency of its own frames but greatly reducing the latency of frames generated at $R2$.

Fig. 14 shows the variation of the average of the latency of frames generated at node $R1$ and $R2$. The average of the latency of node $R1$ and $R2$ is seen to be increasing as node $R2$ moves away from the destination. When $R2$ is positioned close to the destination, the expected latency is higher than the latency of node $R2$ and lower than that of $R1$. Thus, in these positions, it is $R2$ which is providing cooperation to $R1$. In the region midway between $R1$ and D, the expected latency has a similar pattern as the latency of node $R1$, which shows that in this region, $R1$ and $R2$ have almost equal latency. In the other regions the latency of node $R1$ is greater than the expected latency, i.e., $R1$ is providing cooperation for $R2$.

In summary, these three figures demonstrate clearly how the coded cooperation affects the network performance. For a channel with a received SNR of 0 dB, the frame error probability is approximately 58%, thus $R1$ requests cooperation from $R2$ approximately 58% of the times irrespective of the position of $R2$. The channel SNR of $R2$ can be greater or less than 0 dB depending upon its position, and therefore the amount of cooperation requested by $R2$ varies. If $R2$ is in a position where its channel SNR is better than that of $R1$, $R2$ provides cooperation for $R1$, increasing its own latency slightly. However if $R2$ is in a position such that its channel SNR is worse than that of $R1$, it not only takes cooperation from $R1$ but is also unable to provide cooperation to $R1$ and therefore the latency of $R1$ increases by a larger amount.

Fig. 15 and Fig. 16 show the variation of the expected queue length at node $R1$ and $R2$ respectively as the position of $R2$ varies. The expected queue length is that seen at the beginning of a time-slot and follows a pattern similar to the latency.

Fig. 17 shows the variation of the network throughput as the position of $R2$ varies. The network throughput is defined as the the ratio of the total number of frames that have been received successfully to the total number of frames transmitted. Throughput is maximum in the region around midway between $R1$ and D because $R2$ is in a favorable position to transmit its own data frames as well as incremental redundancy frames for $R1$. As $R2$ moves out of this region and away from the destination, $R1$ has to transmit greater number of incremental redundancy frames as well as greater number of data frames, leading to reduced throughput.

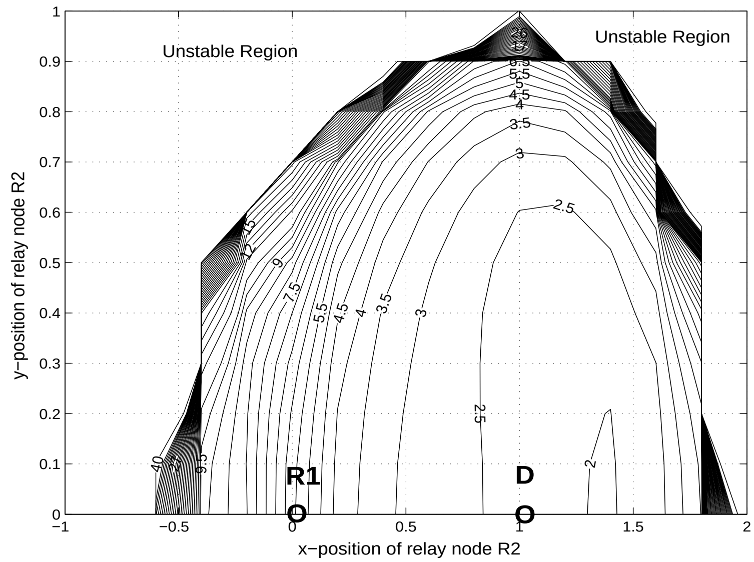

Figure 13: Expected latency of data frames generated at node $R2$ as a function of the position of node $R2$

In Fig. 18, the arrival rate at the two nodes is fixed to $\lambda_N = \lambda_{R1} + \lambda_{R2} = 0.8$. The channel SNR from $R1$ - D is 0 dB, with $R2$ placed half-way between $R1$ and D. The figure shows the variation of latency at node $R1$ and $R2$ as the ratio λ_{R1}/λ_N varies between $(0, 1)$. The minimum latency of $R2$ is higher than the minimum latency of $R1$ because node $R2$ is in a more favorable position to cooperate with $R1$, than vice-versa. As λ_{R1} increases (and λ_{R2} decreases), latency at $R1$ also increases and latency at $R2$ decreases gradually. Thus, although $R2$ is cooperating with $R1$, the latency of $R2$ is driven by the increase of data frame arrival rate at $R2$, rather than by cooperation. The latency at $R1$ and $R2$ is equal, when the ratio of λ_{R1}/λ_N is around 0.6, i.e., the arrival rate at node $R2$ should be less than λ_{R1} to compensate for the requested cooperation.

3. Comparison of Cooperative and Non-cooperative ARQ Protocols

In Figs, 19-20, the advantage of using $CC-ARQ$ over non-cooperative ARQ, namely (conventional) ARQ, and $H-ARQ$, is studied by two performance metrics, the network throughput and expected latency. Nodes $R1$ and $R2$ have the same arrival rate, i.e., $\lambda_{R1} = \lambda_{R2} = 0.5$.

Fig. 19 shows the network saturation throughput as a function of the channel SNR for $CC-ARQ$, $H-ARQ$ and ARQ, when $R2$ is placed half-way between $R1$ and D. It is seen that at low values of channel SNR, e.g., -5 dB, the throughput of $CC-ARQ$ is almost double that of $H-ARQ$ and there is a gain of 75% over ARQ. As the channel SNR improves, the gain of $CC-ARQ$ over $H-ARQ$ and ARQ reduces and the throughput values of the three protocols converge at high channel SNR. The reason is that for low values of channel SNR, a large number of retransmissions are required and therefore $CC-ARQ$ can greatly outperform $H-ARQ$ and ARQ as it incorporates diversity. At higher values of channel SNR, the error probability is very small and therefore very few retransmissions are required. Thus, the selected protocol has a small impact on the throughput performance.

Fig. 20 shows the expected latency of the frames generated at node $R1$ and $R2$ plotted against the arrival rate at each node, for $CC-ARQ$, $H-ARQ$ and ARQ protocols. Node $R2$ is placed half-way between $R1$ and D. The channel SNR between $R1$-D is 0 dB. It is observed that the expected latency is always minimum in the case of $CC-ARQ$ as compared to $H-ARQ$ and ARQ. As the arrival rate increases the expected latency also increases, up to instability. The instability is reached by $CC-ARQ$ at higher arrival rates than in ARQ and $H-ARQ$ protocols. Similar to the case of throughput, the position of the relay node will also affect the expected latency.

Fig. 21 compares the throughput gain of $CC-ARQ$ over $H-ARQ$ as the position of the node $R2$ changes. For this study $R1$ and D are placed as indicated in Fig. 21 and the channel SNR is 0 dB. It can be observed that, when $R2$ is positioned in the region behind D, the throughput achieved by $H-ARQ$ is higher than $CC-ARQ$. In these positions the channel from $R2$ to D is better than that from $R2$ to $R1$, and therefore $R1$ and $R2$ cannot rely on each other for cooperation. However if $R2$ is positioned around midway between $R1$ and D, $R1$ and $R2$ can cooperate with each other and $CC-ARQ$ achieves a higher throughput. Also when $R2$ is in the region behind $R1$, $CC-ARQ$ performs much better than $H-ARQ$ since the channel SNR from $R2$ to $R1$ is higher than the channel SNR from $R2$

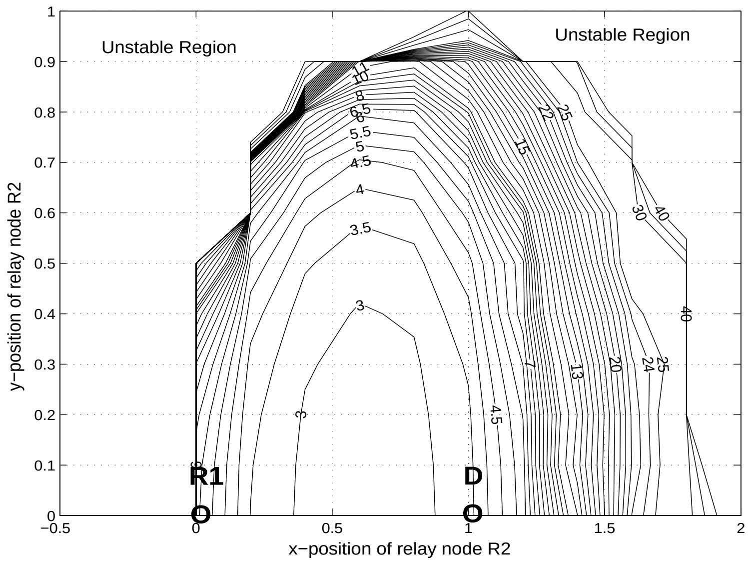

Figure 14: Expected latency of data frames generated at node $R1$ and $R2$ as a function of the position of node $R2$

to D. For a more degraded channel SNR, the throughput gain is expected to be even higher, as indicated by Fig. 19.

Similarly, Fig. 22 shows the difference in throughput of $CC-ARQ$ and (conventional) ARQ. The setup is similar to the above case. In this case it is observed that irrespective of the position of the node $R2$, $CC-ARQ$ achieves a higher throughput than ARQ. The throughput gain in some positions is even higher than that shown in Fig. 19.

Fig. 23 studies one-way $CC-ARQ$, i.e., only $R1$ is transmitting data while $R2$ provides cooperation for $R1$. The network throughput is plotted as a function of the position of $R2$, while $R1$ and D are fixed. It can be seen that the throughput is maximum when $R2$ is in the region centered around the midpoint between $R1$ and D. As $R2$ moves away from this region the throughput decreases as the number of retransmissions is higher. It is interesting to compare the performance of one-way cooperation, versus two-way cooperation. In Fig. 17 it is observed that the throughput gradually reduces when moving away from the destination, but in Fig. 23 there is a region behind $R1$ where the throughput reduces and then increases again. This region of low throughput is because the cooperating node $R2$ is in a position such that it is able to cooperate with $R1$, however due to the channel quality the incremental redundancy frames from $R2$ are not successful and finally $R1$ has to retransmit the same data resulting in reduced throughput. To the left of this region the throughput becomes higher as in this region $R2$ may not be able to cooperate with $R1$ as it has not received the frame from $R1$ correctly, and therefore $R1$ retransmits the data without the transmission of an additional incremental frame from $R2$. In two-way $CC-ARQ$, this effect gets evened out, because when $R2$ is not transmitting incremental frames for $R1$ it is transmitting its own data frames.

5 Summary

A simple ARQ protocol was presented and studied, in which the radio transmission of a source may be enhanced by performing frame retransmission at a cooperative node. Both one-way and two-way cooperation scenarios were investigated by means of simulation and analytical results.

It was confirmed that the cooperative ARQ protocol helps cope with the radio channel noise and fading, thus improving both throughput and latency, when compared to non-cooperative ARQ solutions. Equivalently, the cooperative ARQ protocol may reduce the SNR that is required to meet the desired QoS, when compared to non-cooperative ARQ solutions. The latter option may be specially appealing in radio networks where the signal power is strongly limited, e.g., some types of wireless sensor networks [14].

Further investigation is required on the subject. For example, the policy of giving priority to the incremental redundancy frames — while helping minimize the number of outstanding data frames — may lead to to "starvation" problems. This is the case for example, when one of the sources continuously transmits new data frames and forces the cooperative node to continuously retransmit its own unsuccessfully transmitted frames. Efficient admission control

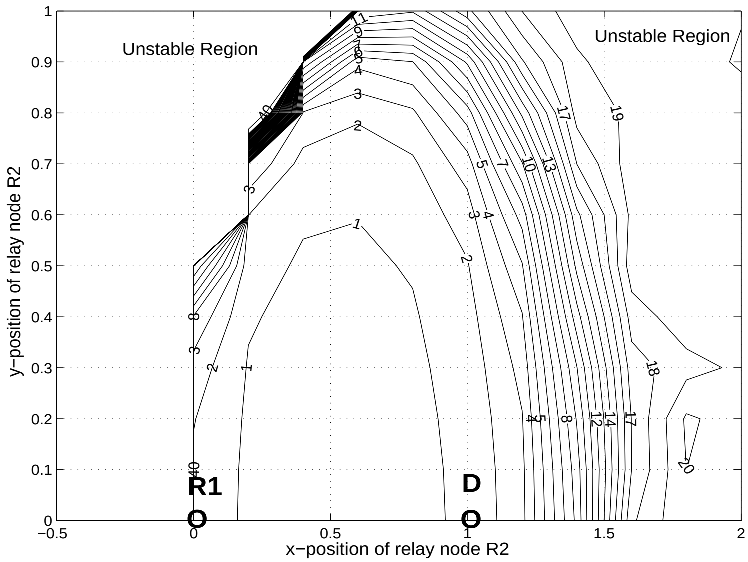

Figure 15: Expected queue length of node $R1$ as a function of the position of node $R2$

solutions to prevent these potential problems must be identified.

Acknowledgment

The authors would like to express their gratitude to Aria Nosratinia, Todd Hunt, and Harsh Shah for their valuable technical input on coded cooperation.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Elsevier Computer Networks*, 38(4):393–442, March 2002.

[2] D. Bertsekas and R. Gallager. *Data Networks (2nd ed.)*. Prentice-Hall, Inc, 1992.

[3] T. M. Cover and A. A. El Gamal. Capacity theorems for the relay channel. *IEEE Trans. Inform. Theory*, 25(5):572–584, 1979.

[4] J. Hagenauer. Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications. *IEEE Trans. on Comm.*, 36(4):389–400, 1988.

[5] T. E. Hunter and A. Nosratinia. Cooperative diversity through coding. In *Proc. IEEE ISIT*, page 220, Laussane, 2002.

[6] M. Janani, A. Hedyat, T. Hunter, and A. Nosratinia. Coded Cooperation in Wireless Communications: Space-Time Transmission and Iterative Decoding. *IEEE Trans. on Signal Processing*, 52(2):362–371, Feb. 2004.

[7] 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*, page 294, Washington, 2001.

[8] Shu Lin and P. Yu. A hybrid ARQ scheme with parity retransmission for error control of satellite channels. *IEEE Trans. on Comm.*, 30(7):1701–1719, 1982.

[9] E. Malkamaki and H. Leib. Evaluating the Performance of Convolutional Codes Over Block Fading Channels. *IEEE Trans. on Infor. Theory*, 45(5):1643–1646, 1999.

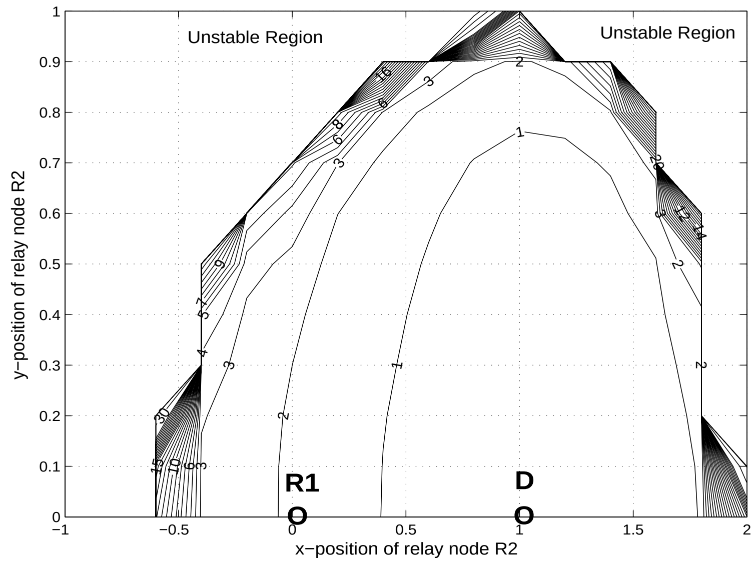

Figure 16: Expected queue length of $R2$ as a function of position of node $R2$

[10] A. Nosratinia, T.E. Hunter, and A. Hedayat. Cooperative communication in wireless networks. *IEEE Communications Magazine*, 2004. Accepted for publication.

[11] John G. Proakis. *Digital Communications (Fourth Edition)*. McGraw-Hill International Edition, 2001.

[12] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity–Part I: System description. *IEEE Trans. Commun.*, 51(11):1927–1938, 2003.

[13] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity–Part II: Implementation aspects and performance analysis. *IEEE Trans. Commun.*, 51(11):1939–1948, 2003.

[14] Marco Tacca, Paolo Monti, and Andrea Fumagalli. Cooperative and non-cooperative ARQ protocols for microwave recharged sensor nodes. Technical report, The University of Texas at Dallas, 2004. UTD/EE-02/2004.

[15] Yu-Ming Wang and Shu Lin;. A modified selective-repeat type-II hybrid ARQ system and its performance analysis. *IEEE Trans. on Comm.*, 31(5):593–608, 1983.

[16] Stephen B. Wicker. *Error Control Systems for Digital Communication and Storage*. Prentice-Hall, 1995.

[17] E. Zimmermann, P. Herhold, and G. Fettweis. The impact of cooperation on diversity-exploiting protocols. In *Proc. of 59th IEEE Vehicular Technology Conference (VTC Spring)*, 2004.

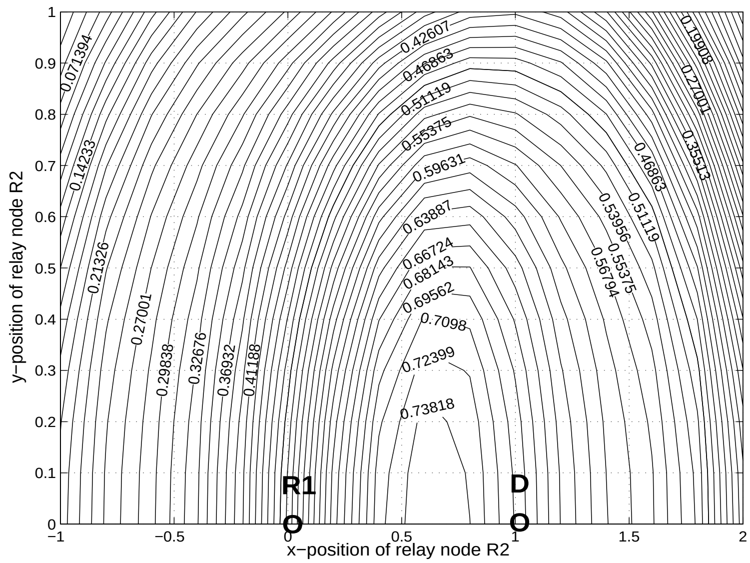

Figure 17: Saturation throughput as a function of the position of node $R2$

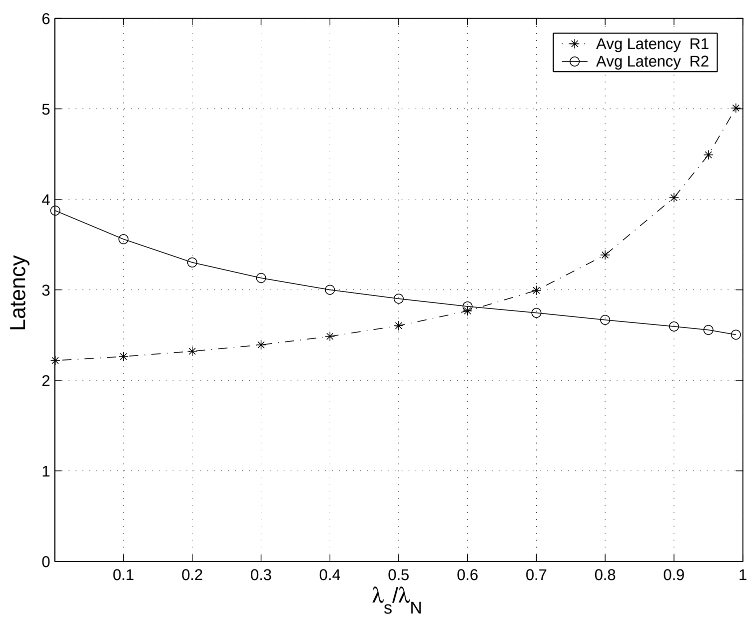

Figure 18: Expected latency of data frames generated at node $R1$ and $R2$ for different arrival rates

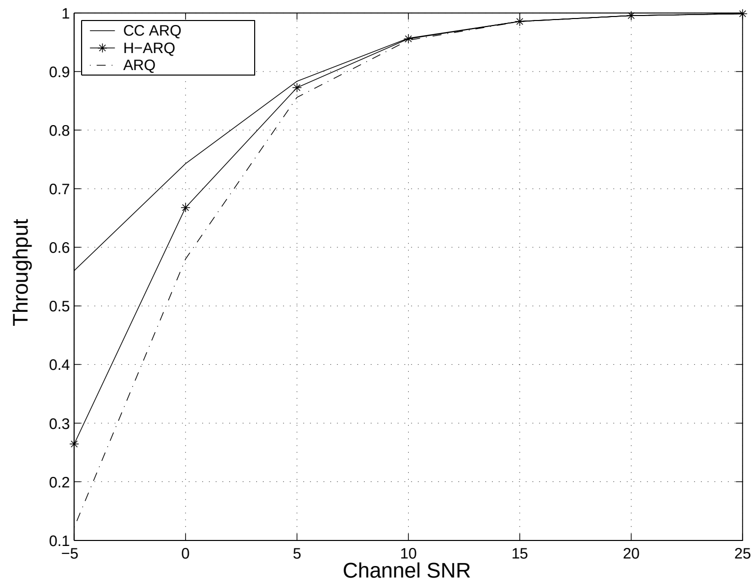

Figure 19: Saturation throughput as a function of the received SNR

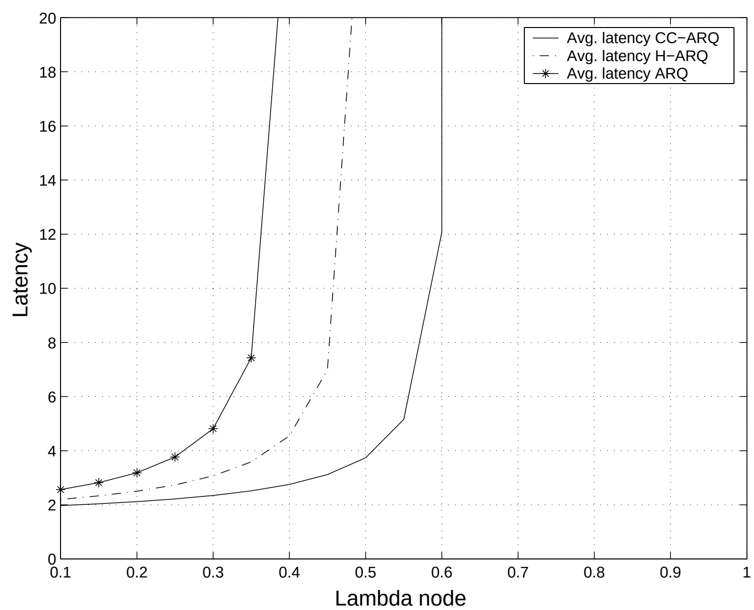

Figure 20: Expected latency of data frames as a function of the arrival rate at node for ARQ, $H-ARQ$ and $CC-ARQ$

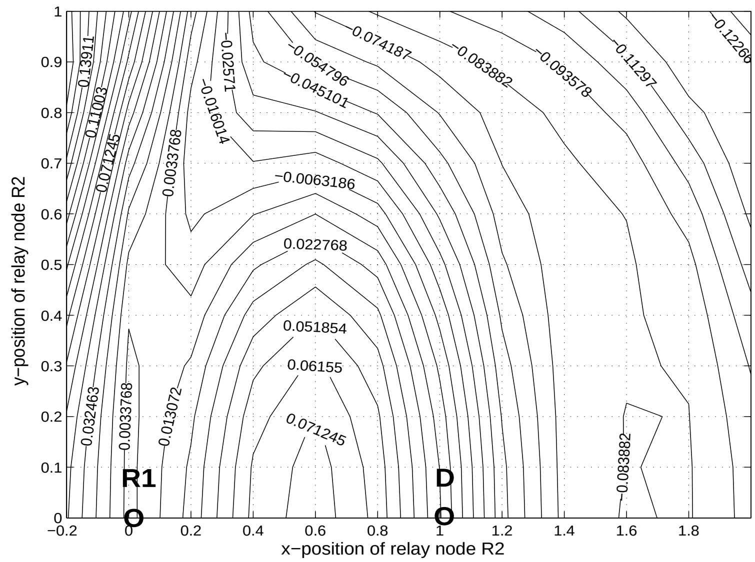

Figure 21: Difference of the saturation throughput of $CC-ARQ$ and $H-ARQ$ as a function of the position of node $R2$

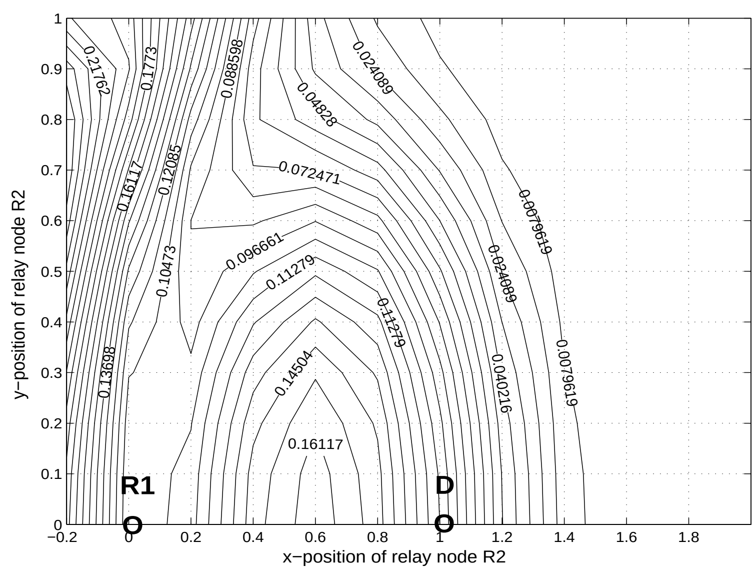

Figure 22: Difference of the saturation throughput of $CC-ARQ$ and ARQ as a function of the position of node $R2$

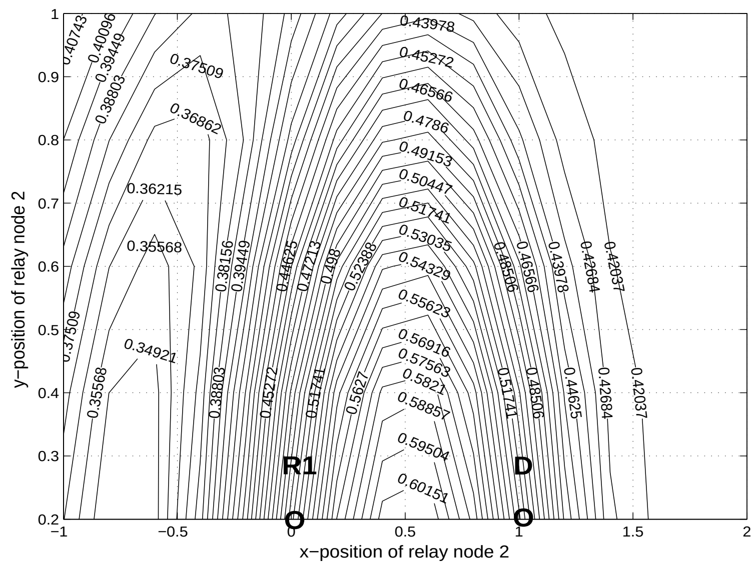

Figure 23: Saturation throughput as a function of the position of node $R2$ for one-way cooperation