Cooperative ARQ Protocols in Slotted Radio Networks

Isabella Cerutti, Andrea Fumagalli, and Puja Gupta

Technical Report UTD/EE/12/2005
August 2005

Cooperative ARQ Protocols in Slotted Radio Networks *

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

August 4, 2005

Abstract

In conventional (non-cooperative) automatic repeat request (ARQ) protocols for radio networks, the corrupted data frames that cannot be correctly decoded at the destination are retransmitted by the source. In cooperative ARQ protocols, data frame retransmissions may be performed by a neighboring node (the relay) that has successfully overheard the source's frame transmission. One advantage of the latter group of ARQ protocols is the spatial diversity provided by the relay.

The first delay model for cooperative ARQ protocols is presented in this paper. The model is analytically derived for a simple set of retransmission rules that make use of both uncoded and coded cooperative communications in slotted radio network. The model estimates the delay experienced by Poisson arriving frames, whose retransmissions (when required) are performed also by a single relay. Saturation throughput, frame latency and buffer occupancy at the source, and relay are quantified and compared against two non-cooperative ARQ protocols.

1 Introduction

Wireless networks are enjoying a widespread diffusion, thanks to a variety of solutions that are increasingly deployed in the field, e.g., cellular, ad-hoc, wireless LAN, and sensor networks. The main drive behind this expansion is the virtually endless list of enabled applications. They address both commercial and military needs, including security, medical monitoring, machine diagnosis, chemical and biological detection [1, 2].

One peculiar characteristic of the radio medium is its inherent broadcast nature. Beside the intended destination, a signal transmitted by a source may be received also by other neighboring nodes that are within earshot. Traditionally, this phenomenon is treated as interference, i.e., the received signal is discarded by the nodes that are not the intended destination.

The quality of the signal received by the destination (and other nodes) depends on various factors, e.g., path loss, fading, and noise. Automatic Repeat Request (ARQ) protocols are used to guarantee reliable delivery over the radio channel. The ARQ protocol specifies how data frames that are not correctly received and detected by the destination must be retransmitted until they are successfully delivered. Most of the available ARQ protocols require the source to retransmit the frames unsuccessfully delivered at the destination [3]. As other neighboring nodes do not take part in the frame retransmission process, these ARQ protocols are referred to as *non-cooperative*.

Cooperative ARQ ($C-ARQ$) protocols permit nodes, other than the source and the destination, to actively help deliver the data frame correctly. The rationale is that a node(s) which is within earshot from the source and the destination may cooperate. This node is referred to as the *relay*. The relay makes use of the received signal (or interference) from the source to improve the overall capacity of the source-destination radio channel. In simple terms, when the source's data frame transmission is not successful, the relay is

*This research was supported in part by NSF Grants No. ANI-0082085, ECS-0225528, CNS-0435429. and the Italian Ministry of University (MIUR) under FIRB project "Enabling platforms for high-performance computational grids oriented to scalable virtual organizations" (contract n. RBNE01KNFP).

invited to take part in the frame retransmission process. By doing so, the destination can rely on data frames that are transmitted by both the source and the relay, possibly yielding 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, i.e., *spatial diversity*.

Focusing on cooperative communications, it must be noted that a number of results has been published on this promising topic. A recent survey on cooperative radio communications can be found in [4]. Initial work on cooperative communications on the Gaussian relay channel is reported in [5]. The relay role is to assist the source, i.e., *single-source cooperation*. More recent works [4, 6, 7, 8, 9, 10] have extended the concept of cooperative communications by taking into account fading and allowing two sources to cooperate with one another at the same time. This case is referred to as *double-source cooperation*, whereby each source interleaves the transmission of its own data frames with the retransmission of the other source's frames. These works can be divided into three categories, according to the method used to realize cooperation communications [4]. In *detect-and-forward* methods [8, 9], the relay detects and retransmits the frame whenever possible. In *amplify-and-forward* methods [6], the relay amplifies the received signal and retransmits it. Both of these methods use retransmission of the exact copy of the data frame. In *coded cooperation* methods [11, 7], cooperation is achieved in the framework of channel coding. The approach in [11, 7] shows the feasibility of coded cooperation and evaluates the benefits, in terms of reduction of bit or frame error probability, when using various codes and cooperation levels. Most of these results focus on the physical layer aspects of cooperation. Only few works have considered related ARQ protocol aspects [10, 12, 13, 14]. In [10] the Signal-to-Noise Ratio (SNR) gain and average number of retransmissions of a single-source cooperative ARQ protocol is studied. In [12], the performance of different cooperative protocols is derived in terms of outage probability and SNR gain and compared against non-cooperative protocol performance. In [14] the saturation throughput and latency of three double-source cooperative ARQ protocols are studied. In [13], a relaying protocol for multiple relays, operating over orthogonal time slots, is proposed as a generalization of hybrid ARQ protocols. Throughput, energy consumption, and outage probability of the proposed protocol are compared against multihop protocol performance. In these studies, network performance of relaying protocols are based on event-driven simulations. Analytically derived delay models for cooperative ARQ protocols are not available.

The objective of this paper is to present the first delay model of four single-source and single-relay Cooperative ARQ ($C-ARQ$) protocols. A simple set of retransmission rules are used. The aim is to reduce the signaling and control overhead in the network, the hardware and algorithm complexity at the nodes, and the changes required in the existing ARQ protocols to introduce cooperation. The model is derived assuming a slotted radio network in which nodes transmit on orthogonal channels to avoid collisions during transmission. It is assumed that for a given source-destination pair the relay is already chosen, and the channel characteristics are fully known, i.e., bit and frame error probability. Frames are generated at the source using a Poisson arrival process. First, the delay model is derived using the second moment of the number of frames stored at the source. Second, the delay model is derived using the frame mean residual transmission and retransmission time. Both the derivations make use of three state stationary probabilities of the embedded Markov chain, that is obtained by sampling the system state at every time slot.

With the proposed delay model some of the advantages and disadvantages of cooperative ARQ protocols may be quantified. A study case is presented in Section 6. Two single-source single-relay cooperative ARQ protocols are considered. In the first protocol, *Stop and Wait* $C-ARQ$, the source can transmits a new frame only when the previous one has been successfully acknowledged by the destination. In the second protocol, *Selective Repeat* $C-ARQ$, the source can transmit a new frame only in the absence of frames requiring retransmission because of timeout expiration. In addition, two coding strategies for single-source single-relay cooperative ARQ protocols are considered. In the first protocol, *type I* $C-ARQ$, the relay transmits an exact replica of the data frame, as it was transmitted by the source. In the second protocol, *type II* $C-ARQ$, the relay transmits a frame, which contains incremental redundancy bits. These bits are computed by the relay and used by the destination by means of a *code combining* strategy for decoding. The latter case is based on the coded cooperation framework discussed in [4]. The performance of the cooperative ARQ protocols is compared against two non-cooperative ARQ protocols, i.e., type I Hybrid-ARQ ($H-ARQ$) [15] and type II $H-ARQ$ [16, 17]. Saturation throughput, expected frame latency, and expected buffer occupancy at the source and relay are evaluated and compared. Various scenarios of offered load, radio channel attenuation, and geographical distribution of the nodes are considered. The study makes use of both simulation and

numerical results obtained from the analytical delay model. The numerical results are shown to match the simulation results under a variety of conditions. The results help understand under what conditions the cooperative ARQ protocols yield superior network performance.

2 Single-Source Single-Relay Cooperative ARQ Protocols

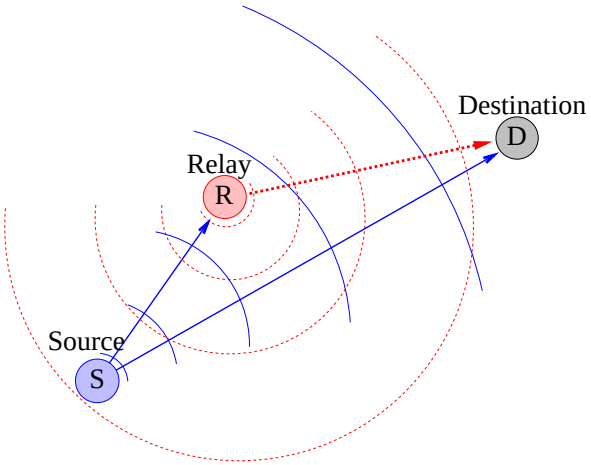
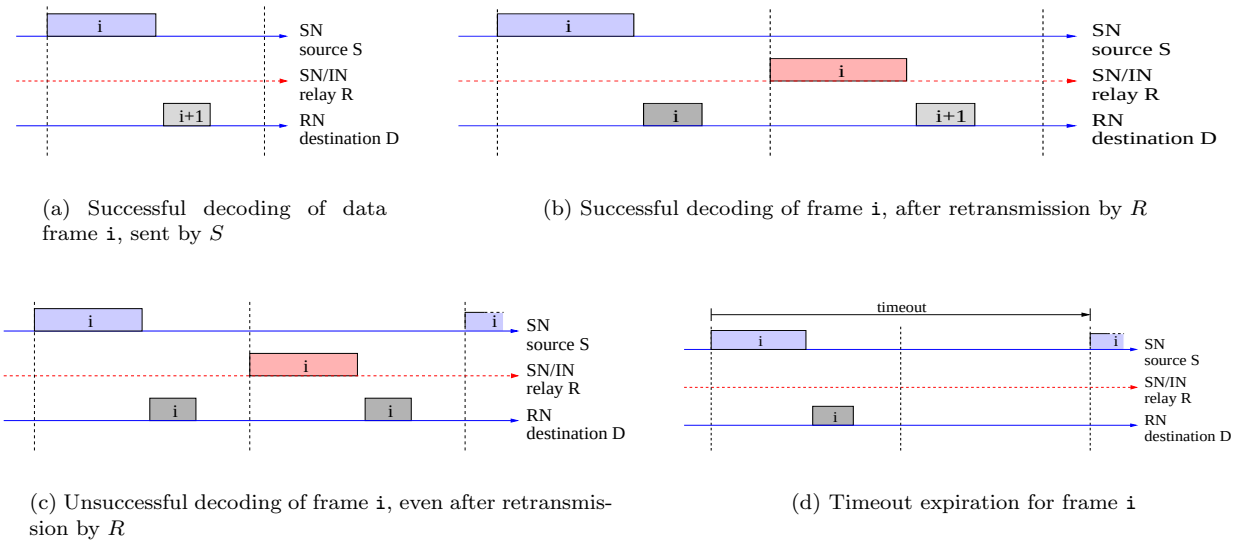


Figure 1: Coded cooperation concept



(a) Successful decoding of data frame `i`, sent by S

(b) Successful decoding of frame `i`, after retransmission by R

(c) Unsuccessful decoding of frame `i`, even after retransmission by R

(d) Timeout expiration for frame `i`

Figure 2: Time chart

The set of retransmission rules for the two single-source and single-relay ARQ protocols is described in this section. Simple rules are chosen to contain protocol and signaling complexity. In addition, this choice facilitates the assessment of the resulting benefits.

Assume that a source S, a destination D, and a relay R are already chosen. Assume that the three nodes have unlimited buffer capacity.

S, D, and R transmit on distinct orthogonal channels. D may receive on both channels simultaneously. Data frames carry some degree of redundancy to perform error detection and correction after transmission.

Time is divided into slots. Time slots on the orthogonal channels are synchronized. During a time slot one data frame and its acknowledgment control frame are transmitted by S (R) and D, respectively. Propagation time is considered to be negligible when compared to frame transmission time. R is provided with the capability of "eavesdropping" on S's channel. In describing and studying the protocols, few additional assumptions are made. Data frames are always received at D and R. However, the payload of some data frames may not be decoded correctly due to transmission errors. Relaxation of these assumptions does not alter the fundamental behavior of the protocols and is omitted in this paper to maintain the protocols description simple.

Four single-source and single-relay ARQ protocols are described next.

2.1 Type I $C-ARQ$ Protocols

In type I Cooperative ARQ ($C-ARQ$) protocols, R transmits an exact replica of the data frame transmitted by S that was unsuccessfully decoded at D. The following four sequences of frame exchange are possible. Each sequence is described with the help of a figure. The data frame Sequence Number is indicated by `SN`. The control frame Request Number is indicated by `RN`.

a) Fig. 2(a): D successfully receives the data frame transmitted by S. Data frame `SN=i` is transmitted by S and acknowledged by D with the transmission of control frame `RN=i+1`. In the next time slot, S may transmit data frame `SN=i+1`.

b) Fig. 2(b): D successfully receives the data frame with the help of R. Data frame `SN=i` is transmitted by S. It is not successfully decoded by D. However, it is correctly received and decoded by R. D sends a (re)transmission request to R using control frame `RN=i`. In the next time slot, R transmits data frame `SN=i`. The frame is correctly received by D, which sends control frame `RN=i+1` to S. In the next time slot, S may begin a new sequence and transmit data frame `SN=i+1`.

c) Fig. 2(c): D does not receive successfully the data frame due to some transmission error(s) detected in the frame from R. This sequence begins in a way similar to the previous one. This time, however, the frame transmitted by R is not correctly received by D. D sends control frame `RN=i` to S which begins a new transmission sequence of data frame `SN=i`.

d) Fig. 2(d): timeout expires. For various reasons, S may not receive the next control frame from D. In this case, a timeout is used at S to avoid deadlock. In the example shown, data frame `SN=i` is transmitted by S. It is not successfully decoded by D. D sends a (re)transmission request to R using control frame `RN=i`. However, neither R was able to decode successfully the data frame transmitted by S. Thus, it cannot cooperate, and the request from D is discarded. Upon expiration of the timeout, S begins a new transmission sequence of data frame `SN=i`. Observing that the longest transmission sequence lasts 2 time slots, a timeout of two time slots provides the lowest latency without triggering unnecessary retransmission.

Type I $C-ARQ$ protocol may be implemented as either *Stop and Wait* (SW) or *Selective Repeat* (SR).
Stop and Wait. In type I $C-ARQ$ Stop and Wait ($C-ARQ\ SW$), node S can transmit data frame `SN=i+1` only after receiving control frame `RN=i+1` from D, i.e., S must wait that data frame i is correctly received by D before attempting the transmission of the successive data frame. Note that in each time slot S and R transmissions are mutually exclusive.
Selective Repeat. In type I $C-ARQ$ Selective Repeat ($C-ARQ\ SR$), node S can transmit data frame `SN=i+1` in the time slot successive to the transmission of data frame `SN=i`, even when D has not received data frame `SN=i` successfully. By using a timeout of 2 time slots, it is possible that S keeps transmitting two distinct data frames in consecutive time slots, until at least one of the two is successfully received by D. Note that S and R may transmit simultaneously during the same time slot.

The relay needs not keep a copy of the data frame for more than one slot-time. Depending on the implemented type I $C-ARQ$ protocol, at most two data frames may be outstanding at the same time, awaiting acknowledgment from D, one transmitted by S, the other by R, respectively. For the same reason, up to two data frames may be acknowledged during the same time slot. For example, at the end of sequence 2, D must inform S that R successfully delivered a data frame. During the same time slot, D may have to inform S that the data frame transmitted by S was successfully received by D too (sequence 1).

2.2 Type II $C-ARQ$ Protocols

The type II Cooperative ARQ ($C-ARQ$) protocol follows the same retransmission rules of type I $C-ARQ$ protocol that are shown in Figs. 2(a)-2(d). The only difference is that the frame transmitted by R contains incremental redundancy bits, instead of the exact replica of the data frame transmitted by S. The Sequence Number of the incremental redundancy frame is indicated by `IN`. Incremental redundancy frame `IN=i` is computed at R, after receiving and decoding data frame `SN=i` from S correctly. When data frame `SN=i` from S is not decoded correctly at R, R cannot cooperate (sequence 3). A special feature of type II $C-ARQ$ protocol is the decoding procedure at D. When data frame `SN=i` from S is not decoded successfully, it is stored at D. When incremental redundancy frame `IN=i` is received, D will attempt to jointly decode the two frames combined, i.e., `SN=i` and `IN=i`, as their combination produces a stronger redundancy code than each individual frame does. At the end of a transmission sequence, irrespective of its success or unsuccess, all frames stored at D are discarded.

Type II $C-ARQ$ protocol may be implemented as either SW or SR. These two options are similar to those already described for type I $C-ARQ$.

3 Non-Cooperative ARQ Protocols

In this section, two non-cooperative protocols are briefly described, i.e., type I and type II Hybrid-ARQ ($H-ARQ$). The two non-cooperative protocols are used to assess the performance gain of the cooperative ARQ protocols in Section 6.

In non-cooperative protocols, the relay is not required, and only the source-destination channels are used. Under the assumption made on the radio channel, i.e., the propagation latency is negligible, the SW and SR implementations of the same non-cooperative ARQ protocol yield the same performance. Thus, only the SW option is considered.

3.1 Type I $H-ARQ$ Protocol

Type I $H-ARQ$ protocol makes use of both error detection and error correction capabilities. (Note that when only error detection capability is used, this is the conventional SW ARQ protocol [15].) The following retransmission rules are used.

S sends the next data frame, e.g., `SN=i`. Three cases are possible. If data frame `SN=i` is received and decoded correctly at D, control frame `RN=i+1` is sent. At this point, a new data frame may be transmitted. If data frame `SN=i` is received but not decoded correctly at D, control frame `RN=i` is sent, requesting S to retransmit data frame `SN=i`. If data frame `SN=i` is not received by D a timeout is used at S to retransmit data frame `SN=i`. The timeout value is chosen to be one time slot.

3.2 Type II $H-ARQ$ Protocol

Type II $H-ARQ$ protocol follows the same retransmission rules of type I $H-ARQ$ protocol. The only difference is that a frame containing incremental redundancy bits is transmitted by S when the data frame is not received correctly by D. At D, the data frame and the incremental redundancy frame are jointly decoded for improved performance [15]. The following retransmission rules are used.

S sends the next data frame, e.g., `SN=i`. If data frame `SN=i` is received and decoded correctly at D, control frame `RN=i+1` is sent. At this point, a new data frame may be transmitted. If data frame `SN=i` is received but not decoded correctly at D, control frame `RN=i` is sent, requesting retransmission. In response, S sends incremental redundancy frame `IN=i`. If D is now able to jointly decode frames `SN=i` and `IN=i`, control frame `RN=i+1` is sent, and a new data frame may be transmitted. If D is not able to jointly decode the two frames, control frame `RN=i` is sent once again, and copies of the two received frames at D are discarded. S alternates the transmission of frames `SN=i` and `IN=i` until control frame `RN=i+1` is received. A retransmission timeout of one time slot is used at S to avoid deadlock.

4 Analytical Framework

This section describes the analytical framework that is used to derive the delay models of the described ARQ protocols, both cooperative and non-cooperative.

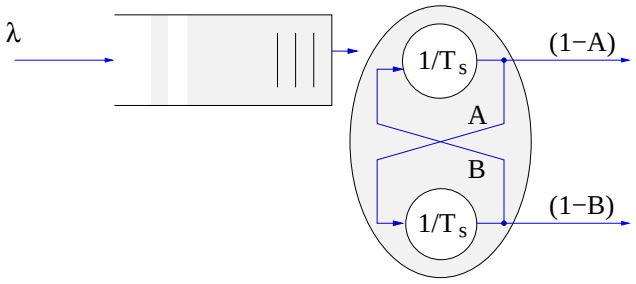


Figure 3: Queueing model with a two-stage service facility

Consider the queueing system in Fig. 3, which consists of a queue of unlimited capacity and a service facility of two cascaded servers with feedback. The job (data frame) arrivals constitute a continuous-time Poisson process of rate λ. Service time is slotted, i.e., service may initiate only at the beginning of a time slot. The service time of both servers is deterministic and equal to one time slot, i.e., T_s. After being served (i.e., transmitted) by the top server, the frame either leaves the system with probability $(1-A)$, or moves to the bottom server with probability A. After being served by the bottom server the frame either leaves the system with probability $(1-B)$ or returns to the top server with probability B. The frame may circulate between the two servers several times, before being released from the system. Probabilities A and B are time invariant.

Two service policies are considered, i.e., *mutually exclusive servers* and *independent servers*.

4.1 Mutually Exclusive Servers

With this policy, at most one frame can be in the service facility, i.e., only one of the two servers may be busy at one time. The next frame waiting in the queue is allowed to enter service only after the frame currently in the service facility leaves the system. The time spent by the frame in the service facility is a random variable X, whose first and second moments are, respectively,

$$\begin{aligned}\overline{X} &= T_s \sum_{k=0}^{\infty}(2k+1)(1-A)(A\cdot B)^k + \\ &+ T_s \sum_{k=0}^{\infty}(2k)A(1-B)(A\cdot B)^{k-1} = \\ &= T_s 2 \sum_{k=0}^{\infty} k(A\cdot B)^{k-1}\Big[(1-A)(A\cdot B) + A(1-B)\Big] + \\ &+ T_s \sum_{k=0}^{\infty}(1-A)(A\cdot B)^k = T_s \frac{1+A}{1-A\cdot B} \end{aligned} \tag{1}$$

$$\begin{aligned}\overline{X^2} &= T_s^2 \sum_{k=0}^{\infty}(2k+1)^2(1-A)(A\cdot B)^k + \\ &+ T_s^2 \sum_{k=0}^{\infty}(2k)^2 A(1-B)(A\cdot B)^{k-1} = \\ &= T_s^2 \frac{1+3A+A\cdot B(3+A)}{(1-A\cdot B)^2}. \end{aligned} \tag{2}$$

The waiting time, i.e., the time spent by a frame in the queueing system, can be found using the Pollaczek-Khinchin formula for M/G/1 queues with vacations:

$$W = \frac{\lambda \overline{X^2}}{2(1-\rho)} + \frac{T_s}{2} \tag{3}$$

where ρ is the utilization factor:

$$\rho = \lambda \overline{X}. \tag{4}$$

The sojourn time, T, can be found by adding the service time to W:

$$T = W + \overline{X}. \tag{5}$$

The utilization of the top and bottom servers, N_{1s} and N_{2s} respectively, are related to ρ as follows:

$$N_{1s} = \rho \tag{6}$$

$$N_{2s} = A \cdot \rho. \tag{7}$$

Using Little's theorem, the average number of frames in the system, i.e., in the queue and service facility, is:

$$N = \lambda \cdot T. \tag{8}$$

The average number of frames waiting in the queue and in the top server, N_{q1s}, is:

$$N_{q1s} = N - N_{2s} = \lambda \cdot T - A \cdot \rho. \tag{9}$$

These average values are also seen by the Poisson arriving frames.

Sampled at the beginning of each time slot, the expected number of frames in the system and the expected number of frames waiting in the queue and in the top server are, respectively,

$$N^{sl} = N - \frac{\lambda T_s}{2} \tag{10}$$

$$N^{sl}_{q1s} = N_{q1s} - \frac{\lambda T_s}{2}. \tag{11}$$

4.2 Independent Servers

When the servers are *independent* — i.e., both servers may be in service at the same time — two distinct frames may be in the service facility at the same time.

Because two frames may be served in tandem, the Pollaczek-Khinchin formula cannot be used in this case. The solution is found by computing and making use of part of the stationary distribution of an embedded Markov chain. Let P be defined as:

$$P = A \cdot B. \tag{12}$$

For simplicity, assume that $B = 1$ and replace A with P in Fig. 3. Note that the expected waiting time of the independent server queueing system is not affected by the value chosen for B. On the contrary, the expected service time will need to be corrected with an additional term when $B < 1$.

4.2.1 Embedded Markov Chain

The (discrete-time) embedded Markov chain is derived by sampling the state of the system shown in Fig. 3 at the beginning of each time slot. The state of the chain is the tuple $S_{i,j}$, whereby $i \geq 0$ indicates the number of frames waiting in queue and in the top server, and $j = \{0, 1\}$ indicates the number of frames in the bottom server. Fig. 4 illustrates the state transition diagram and the transition probabilities of the chain. Note that $a_k = e^{-\lambda T}(\lambda T)^k / k!$ is the probability of k Poisson arrivals during one time slot.

The utilization of the top server is:

$$\rho = \frac{\lambda T_s}{1 - P}. \tag{13}$$

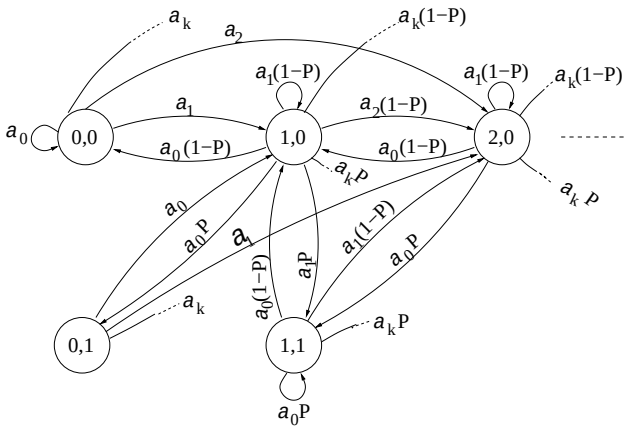


Figure 4: Embedded Markov chain for independent servers queueing model

The Markov chain is ergodic if the stability condition is satisfied, i.e., $\rho < 1$. Under this condition, three steady state probabilities, i.e., $\pi_{i,j} = P_r\{\text{chain is in state } S_{i,j}\}\ \forall i + j \leq 1$, can be derived easily:

$$\begin{cases} \pi_{0,0} = \frac{(1-\rho)(1-P)}{(1-a_0P)} \\ \pi_{0,1} = \frac{P(1-\rho)(1-a_0)}{(1-a_0P)} \\ \pi_{1,0} = \frac{(1-\rho)(1-a_0)}{a_0(1-a_0P)}, \end{cases} \tag{14}$$

by solving the system of equations:

$$\begin{cases} \pi_{0,0} = a_0\pi_{0,0} + a_0(1-P)\pi_{1,0} \\ \pi_{0,1} = a_0P\pi_{1,0} \\ \pi_{0,0} + \pi_{0,1} = 1 - \rho. \end{cases} \tag{15}$$

4.2.2 Deriving the Expected Frame Latency

With just the three state probabilities in (14), it is possible to evaluate the expected frame latency in the system and the buffer occupancy at S and R.

For this scope two alternate approaches are presented.

Approach I: Second Moment of the Number of Frames in the System

The first approach makes use of the second moment of the number of frames awaiting in the queue and in the top server. (This technique is similar to the M/G/1 derivation presented in [18].) Let v_n be the number of frames arriving during time slot n. Let q_n be the number of frames awaiting in the queue and in the top server at the beginning of time slot n. Let $\triangle_{q_n}$ be the shifted unit step function in q_n:

$$\triangle_{q_n} = \begin{cases} 1 & \text{if } q_n > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Let $\triangle_{Pq_{n-1}}$ be the shifted unit step function in q_n with probability P:

$$\triangle_{Pq_{n-1}} = \begin{cases} 1 & \text{with probability } P \text{ if } q_n > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The value of q_{n+1} is given by:

$$q_{n+1} = q_n + v_n - \triangle_{q_n} + \triangle_{Pq_{n-1}}. \tag{18}$$

Note that (18) takes into account q_n, i.e., the number of frames in the system at time n, v_n, i.e., the number of newly generated frames arrived during time slot n, $\triangle_{q_n}$, i.e., the departure of a frame from the top server,

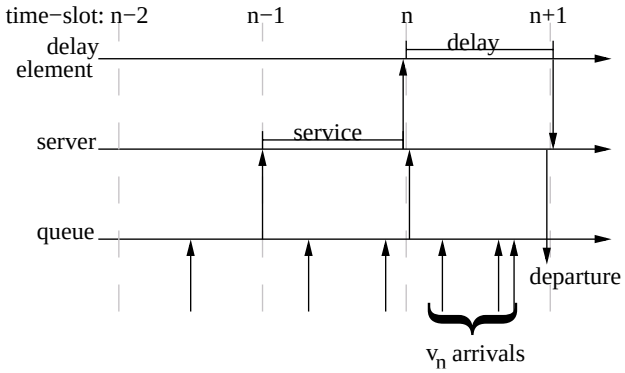


Figure 5: Time diagram of transitions in the system

and $\triangle_{Pq_{n-1}}$, i.e., the return of a previously served frame to the top server. A graphical representation of the transitions is displayed in Fig. 5.

The objective now is the evaluation of the first moment of q_n, when the embedded Markov chain is ergodic, i.e., when

$$\lim_{n\to\infty} E[q_n^j] = E[\tilde{q}^j] \quad \forall j = 1, 2, \ldots \tag{19}$$

$E[\tilde{q}]$ may be computed from the expectation of the square of (18), i.e.,

$$\begin{aligned} q_{n+1}^2 = q_n^2 + v_n^2 + \triangle_{q_n}^2 + \triangle_{Pq_{n-1}}^2 + 2q_n v_n - 2q_n \triangle_{q_n} + 2q_n \triangle_{Pq_{n-1}} - 2v_n \triangle_{q_n} + \\ + 2v_n \triangle_{Pq_{n-1}} - 2 \triangle_{q_n} \triangle_{Pq_{n-1}}. \end{aligned} \tag{20}$$

Keeping in mind that:

$$\triangle_{q_n}^2 = \triangle_{q_n} \tag{21}$$

$$\triangle_{Pq_{n-1}}^2 = \triangle_{Pq_{n-1}} \tag{22}$$

$$q_n \triangle_{q_n} = q_n. \tag{23}$$

The expectation of (20) becomes:

$$\begin{aligned} E[q_{n+1}^2] = E[q_n^2] + E[v_n^2] + E[\triangle_{q_n}] + E[\triangle_{Pq_{n-1}}] + 2E[q_n v_n] - 2E[q_n] + 2E[q_n \triangle_{Pq_{n-1}}] + \\ - 2E[v_n \triangle_{q_n}] + 2E[v_n \triangle_{Pq_{n-1}}] - 2E[\triangle_{q_n} \triangle_{Pq_{n-1}}]. \end{aligned} \tag{24}$$

Noting that the arrival process is independent of n, three averages in (24) may be rewritten in the product form, i.e., $E[q_n v_n]$, $E[v_n \triangle_{q_n}]$, and $E[v_n \triangle_{Pq_{n-1}}]$. In the presence of ergodicity, taking the limit as $n \to \infty$ yields:

$$\begin{aligned} E[\tilde{q}^2] = E[\tilde{q}^2] + E[\tilde{v}^2] + E[\triangle_{\tilde{q}}] + E[\triangle_{P\tilde{q}}] + 2E[\tilde{q}]E[\tilde{v}] - 2E[\tilde{q}] + 2 \lim_{n\to\infty} E[q_n \triangle_{Pq_{n-1}}] + \\ - 2E[\tilde{v}]E[\triangle_{\tilde{q}}] + 2E[\tilde{v}]E[\triangle_{P\tilde{q}}] - 2 \lim_{n\to\infty} E[\triangle_{q_n} \triangle_{Pq_{n-1}}]. \end{aligned} \tag{25}$$

For Poisson arrivals, the first and second moment of v_n are, respectively [18]:

$$E[\tilde{v}] = \lim_{n\to\infty} E[v_n] = \lambda T_s = \rho(1 - P) \tag{26}$$

$$\begin{aligned} E[\tilde{v}^2] &= \lim_{n\to\infty} E[v_n^2] = (\lambda T_s)^2 + (\lambda T_s) \\ &= \rho^2(1 - P)^2 + \rho(1 - P) \end{aligned} \tag{27}$$

The first moment of $\triangle_{q_n}$ and $\triangle_{Pq_n}$ are respectively:

$$E[\triangle_{\tilde{q}}] = \lim_{n\to\infty} E[\triangle_{q_n}] = 0 \cdot P_r\{\tilde{q} = 0\} + 1 \cdot P_r\{\tilde{q} > 0\} = P_r\{\tilde{q} > 0\} = P_r\{\text{server busy}\} = \rho \tag{28}$$

$$E[\triangle_{P\tilde{q}}] = \lim_{n\to\infty} E[\triangle_{Pq_n}] = 1 \cdot P \cdot P_r\{\tilde{q} > 0\} = P \cdot P_r\{\tilde{q} > 0\} = P \cdot P_r\{\text{server busy}\} = \mathrm{P}\rho. \tag{29}$$

Replacing (26)-(29) and subtracting $E[\tilde{q}^2]$ in (25) yields:

$$0 = \Big[\rho^2(1-P)^2 + \rho(1-P)\Big] + \rho + P\rho + 2E[\tilde{q}] + \rho(1-P) - 2E[\tilde{q}] + 2\lim_{n\to\infty} E[q_n \triangle_{Pq_{n-1}}] + \\ -2\rho(1-P)\rho + 2\rho(1-P)P\rho - 2\lim_{n\to\infty} E[\triangle_{q_n}\triangle_{Pq_{n-1}}]. \tag{30}$$

As derived in Appendix:

$$\lim_{n\to\infty} E[q_n \triangle_{Pq_{n-1}}] = PE[\tilde{q}] - P\pi_{0,0}\rho(1-P) - P\pi_{0,1}(1+\rho(1-P)), \tag{31}$$

and

$$\lim_{n\to\infty} E[\triangle_{q_n} - \triangle_{Pq_{n-1}}] = P\left(\rho - a_0\pi_{1,0}\right), \tag{32}$$

where $\pi_{i,j}$ is given in (14).

By replacing (31) and (32) in (30), one can solve for $E[\tilde{q}]$:

$$E[\tilde{q}] = \frac{1}{2(1-\rho)(1-P)}\Bigg[2\rho(1-P) - \rho^2(1-P)^2 - 2P\Big(\rho(1-P)\pi_{0,0} + (1+\rho(1-P))\pi_{0,1} - a_0\pi_{1,0}\Big)\Bigg]. \tag{33}$$

Finally, the following expressions are obtained:

$$N_{q1s}^{sl} = E[\tilde{q}] \tag{34}$$

$$N_{1s} = \rho \tag{35}$$

$$N_{2s} = P\rho \tag{36}$$

$$N^{sl} = N_{q1s}^{sl} + N_{2s} = N_{q1s}^{sl} + P\rho \tag{37}$$

$$N = N^{sl} + \frac{\lambda T}{2} = N_{q1s}^{sl} + P\rho + \frac{\lambda T_s}{2} \tag{38}$$

$$N_{q1s} = N_{q1s}^{sl} + \frac{\lambda T_s}{2} \tag{39}$$

$$\tag{40}$$

Applying Little's theorem, the expected latency experienced by the frame in the system is:

$$T = \frac{N}{\lambda} = \frac{E[\tilde{q}]}{\lambda} + \frac{P}{1-P}T_s + \frac{T_s}{2}. \tag{41}$$

Approach II: Mean Residual Time

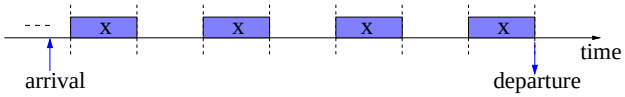


Figure 6: Frame transmission sequence at the top server when the system is in state $S_{0,1}$ and $S_{1,0}$

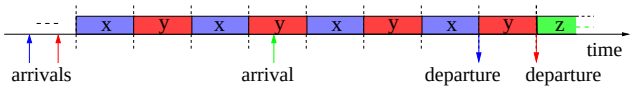


Figure 7: Frame transmission sequence at the top server when the system is in states $S_{i,j}$, with $i+j \geq 2$

The approach makes use of the mean residual time.

Two distinct system behaviors can be observed. In the first behavior, there is only one frame in the system, i.e., $S_{0,1}$ and $S_{1,0}$, and just one server is busy during each time slot (as it happens in the case of

mutually exclusive servers). Fig. 6 shows multiple transmission attempts of the same frame `x` at the top server while the system is in states $S_{0,1}$ and $S_{1,0}$. In the second behavior, there are 2 or more frames in the system, i.e., $S_{i,j}$ with $i+j \geq 2$, and both servers are busy. Fig. 7 shows multiple transmission attempts of frames `x` and `y` at the top server while the system is in states $S_{i,j}$ with $i+j \geq 2$. The sequence continues until one of the two frames is successfully transmitted, i.e., leaves the system.

In the latter behavior, notice that it is possible to swap identities of frames `x` and `y` without altering the resulting expected frame latency. For example, the sequence of Fig. 7 can be equivalently replaced with

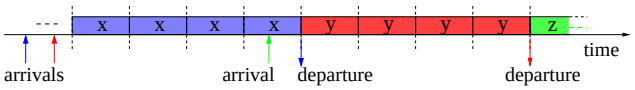


Figure 8: Frame transmission equivalent sequence in states $S_{i,j}$, with $i+j \geq 2$

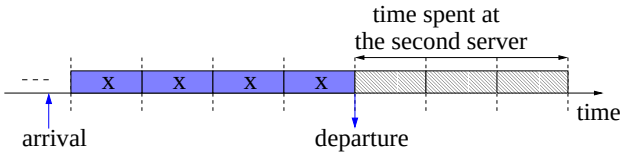


Figure 9: Frame transmission equivalent sequence in state $S_{i,j}$, with $i+j=1$

the sequence of Fig. 8, in which one frame is served continuously by the top server (as in the case of single server). In this case, the policy of the server is to complete the service of the current frame (lasting a number of time slots), before starting the service of the next frame. The frame expected service time is then:

$$\sum_{k=1}^{\infty} T_s k(1-P)P^{k-1} = \frac{T_s}{1-P}. \tag{42}$$

In the former behavior, assume to swap the frame in the top server with the idle top server, i.e., swapping state $S_{0,1}$ with $S_{1,0}$. To make sure that the expected frame latency is not affected, a number of idle slots must be forced at the top server (vacation) after completing the service time of the frame, as shown in Fig. 9. The vacation time takes into account the expected time spent by the frame at the bottom server, i.e., top server idle time. The frame expected service time in this case is then:

$$\sum_{k=1}^{\infty} T_s k(1+a_0P)(1-P)P^{k-1} = \frac{T_s}{1-P} + \frac{a_0PT_s}{1-P}. \tag{43}$$

Note that states $S_{0,1}$, $S_{1,0}$, and $S_{0,0}$ are already sufficient to determine which behavior applies. The frame expected waiting time is then computed using the mean residual service time of the top server, i.e.,

$$W = \frac{T_s}{2} + \rho\frac{PT_s}{1-P} + N_{q1s}\frac{T_s}{1-P} + \pi_{1,0}\frac{a_0PT_s}{1-P}. \tag{44}$$

These are the contributions in (44). Upon arrival, a frame must always wait an average half time slot for synchronization, i.e., $T_s/2$. In addition, if a frame is already in service, it must wait for the expected residual service time of the frame already in service, i.e., $PT_s/(1-P)$, plus the service time of the average number of frames waiting in queue, i.e., $N_{q1s}T_s/(1-P)$. Finally, there is an additional term that adjusts for the vacation time of the top server, i.e., $\pi_{1,0}\frac{a_0PT_s}{1-P}$. By using (15), (14), and applying Little's theorem, i.e., $N_{q1s} = \lambda W$, (44) can be solved for W

$$W = \frac{T_s}{2(1-\rho)} + \frac{\rho}{(1-\rho)}\frac{PT_s}{(1-P)} + \frac{PT_s(1-a_0)}{(1-a_0P)(1-P)}. \tag{45}$$

The frame expected time in the system is found by adding the service time to W, i.e.,

$$T = W + \frac{T_s}{1-P} + \frac{a_0 P T_s}{1-P}\,\frac{\pi_{1,0}}{1-\pi_{0,0}-\pi_{0,1}} = \tag{46}$$
$$= \frac{T_s}{2(1-\rho)} + \frac{\rho}{(1-\rho)}\frac{PT_s}{(1-P)} + \frac{T_s}{1-P} + \frac{P(1-a_0)}{\rho(1-P)(1-a_0P)}T_s.$$

Note that the first three terms in the last line of (46) represent the expected time in the queueing system with one single (top) server, and the last term accounts for the additional time spent in the bottom server.

Finally, the following expressions are obtained:

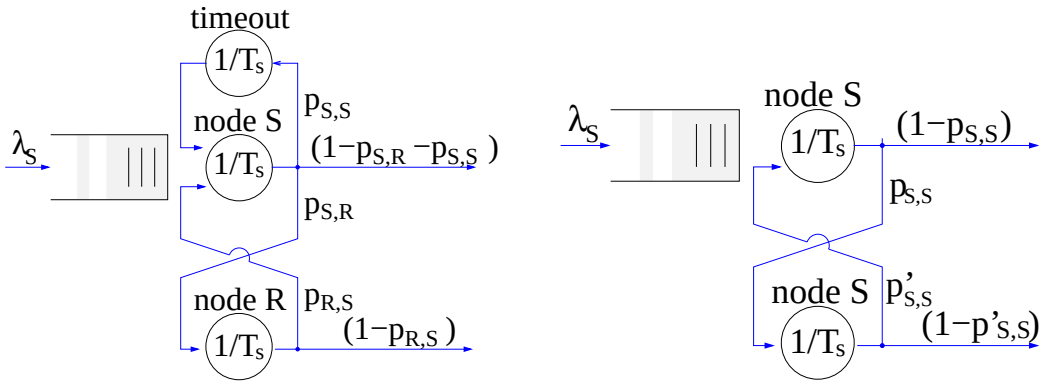
$$N_{q1s} = N - N_{2s} = \lambda T - P\rho. \tag{47}$$
$$N_{2s} = P\rho. \tag{48}$$

while the expressions for N_{2s}, N, N^{sl}, N^{sl}_{q1s}, and N_{1s} and can be found in (36), (8), (10), (11), and (6) respectively.

The interested reader can verify that (41), and (39) match perfectly with (46), and (47).

5 Delay Models for the ARQ Protocols

The analytical framework presented in Section 4 is applied now to evaluate the latency and the buffer occupancy of the six ARQ protocols described in Section 2 and 3. The derivation is carried out assuming that both frame loss and control frame error probability are negligible.



(a) C-ARQ protocols

(b) H-ARQ protocols

Figure 10: Queueing systems

Fig. 10 shows the queueing system adopted to model the four C-ARQ protocols (Fig. 10(a)) and two H-ARQ protocols (Fig. 10(b)). Recall that the service time is slotted and lasts T_s.

For each ARQ protocol, a distinct set of expressions for the probabilities $p_{S,S}$, $p_{S,R}$, $p_{R,S}$, and $p\prime_{S,S}$ is computed. In the calculation, it is assumed that the following probabilities are time invariant and known:

- $\overline{P1}_{i,j}$: frame error probability, i.e., probability that node j unsuccessfully decodes the frame sent by node i;
- $\overline{P2}_{j,D}$: probability that D unsuccessfully decodes the original data frame sent by S combined with the incremental redundancy frame sent by node j.

5.1 Cooperative ARQ Protocols

Fig. 10(a) shows the queueing model for C-ARQ protocols. Three servers are shown, i.e., node S, node R, and timeout. Timeout is required when the data frame is not correctly received by the relay. The timeout duration is selected to be 2 time slots. At the end of each service, the frame moves to another server or leaves the system as indicated by the transition probabilities shown in the figure, i.e., $p_{S,R}$, $p_{R,S}$, and $p_{S,S}$. It can be demonstrated that the solution of the queue shown in Fig. 10(a) is the same of the queue shown in Fig. 3 by using the following probabilities:

$$A = p_{S,S} + p_{S,R} \tag{49}$$

$$B = p_{R,S}\frac{p_{S,R}}{p_{S,S}+p_{S,R}} + 1 \cdot \frac{p_{S,S}}{p_{S,S}+p_{S,R}} = \frac{p_{R,S}\ p_{S,R}+p_{S,S}}{p_{S,S}+p_{S,R}}. \tag{50}$$

5.1.1 Type I $C-ARQ$ Stop and Wait

For type I $C-ARQ\ SW$ protocol, the queueing system with mutually exclusive servers (Section 4.1) is used, and

$$p_{S,S} = \overline{P1}_{S,D} \cdot \overline{P1}_{S,R} \tag{51}$$

$$p_{S,R} = \overline{P1}_{S,D} \cdot \left(1 - \overline{P1}_{S,R}\right) \tag{52}$$

$$p_{R,S} = \overline{P1}_{R,D}. \tag{53}$$

The expected waiting time, i.e., W, latency, i.e., T, and buffer occupancy at node S (excluding outstanding frames), i.e., N_{q1s} and N_{q1s}^{sl}, are given by (3), (5), (8), (10), respectively. The buffer occupancies at R is:

$$N_R^{sl} = p_{S,R} \cdot \rho, \tag{54}$$

and the buffer occupancies at D is:

$$N_D^{sl} = (p_{S,R} + p_{S,S})\rho, \tag{55}$$

where ρ is given in (4). The saturation throughput, i.e., Th, is obtained from the average number of transmissions per delivered data frame, i.e., t_r,

$$E[t_r] = \frac{\overline{X}}{T_s} \tag{56}$$

$$Th = \frac{1}{E[t_r]}. \tag{57}$$

5.1.2 Type II $C-ARQ$ Stop and Wait

For type II $C-ARQ\ SW$ protocol, the queueing system with mutually exclusive servers (Section 4.1) is used, and

$$p_{S,S} = \overline{P1}_{S,D} \cdot \overline{P1}_{S,R} \tag{58}$$

$$p_{S,R} = \overline{P1}_{S,D} \cdot \left(1 - \overline{P1}_{S,R}\right) \tag{59}$$

$$p_{R,S} = \frac{\overline{P2}_{R,D}}{\overline{P1}_{S,D}}. \tag{60}$$

All the performance metrics are obtained as already explained for type I $C-ARQ\ SW$ protocol.

5.1.3 Type I $C-ARQ$ Selective Repeat

For type I $C-ARQ\ SR$ protocol, the queueing system with independent servers (Section 4.2) is used. The transition probabilities are the same of type I $C-ARQ\ SW$, i.e., (51), (52), and (53). The probability of feedback is:

$$P = A \cdot B = p_{R,S}p_{S,R} + p_{S,S} = \overline{P1}_{S,D}(1 - \overline{P1}_{S,R})\overline{P1}_{R,D} + \overline{P1}_{S,D}\overline{P1}_{S,R}. \tag{61}$$

The terms in (61) take into account the unsuccessful transmission sequences **c** and **d** that are described in Section 2.1.

The expected waiting time, i.e., W, is given in (45). The expected buffer occupancies at S (excluding outstanding frames), i.e., N_{q1s}, N_{q1s}^{sl}, and R, i.e., N_R^{sl}, are given in (8) or (39), (11) or (34), and (54), respectively, using the value of ρ given in (13). The expected frame latency[1] is obtained by modifying (41) or (46) to take into account the delay incurred when retransmission by R is successful, i.e.,

$$T = W + \frac{T_s}{1-P} + \frac{a_0 P T_s}{1-P} \frac{\pi_{1,0}}{1-\pi_{0,0}-\pi_{0,1}} + p_{S,R} \cdot (1-p_{R,S}) \frac{T_s}{1-P}. \tag{62}$$

The saturation throughput is

$$Th = \max\left(\frac{\lambda}{\lambda/(1-P)}, \frac{\lambda}{\lambda \cdot p_{S,R}/(1-P)}\right) = (1-P), \tag{63}$$

whereby $\lambda/(1-P)$ is the maximum flow of frames correctly delivered by S, and $\lambda/(1-P) \cdot p_{S,R}$ is the maximum flow of frames correctly delivered by R.

5.1.4 Type II $C-ARQ$ Selective Repeat

For type II $C-ARQ$ SR protocol, the queueing system with independent servers (Section 4.2) is used. The transition probabilities are the same of type II $C-ARQ$ SW, i.e., (58), (59), and (60). The probability of feedback is:

$$\begin{aligned} P = A \cdot B = p_{R,S}p_{S,R} + p_{S,S} = \overline{P1}_{S,D}(1-\overline{P1}_{S,R})\frac{\overline{P2}_{R,D}}{\overline{P1}_{S,D}} + \overline{P1}_{S,D}\overline{P1}_{S,R} = \\ = \overline{P2}_{R,D}(1-\overline{P1}_{S,R}) + \overline{P1}_{S,D}\overline{P1}_{S,R}. \end{aligned} \tag{64}$$

The terms in (64) take into account the unsuccessful transmission sequences **c** and **d** that are described in Section 2.1.

All the performance metrics are obtained as already explained for type I $C-ARQ$ SR protocol.

5.2 Non-Cooperative ARQ Protocols

Recall that in the non-cooperative protocols, S sends frames to D, without the help of R. Fig. 10(b) shows the queueing model for H-ARQ protocols. At the end of each service, the frame moves to the other server or leaves the system as indicated by the transition probabilities shown in the figure, i.e., $p_{S,S}$, and $p\prime_{S,S}$. Under the assumption of negligible frame loss and control frame errors, timeouts are not required.

The solution of the queue shown in Fig. 10(b) is the same of the queue shown in Fig. 3 by using the following probabilities:

$$A = p_{S,S} \tag{65}$$

$$B = p\prime_{S,S}. \tag{66}$$

5.2.1 Type I $H-ARQ$ Stop and Wait

For type I $H-ARQ$ protocol, either the queueing systems with mutually exclusive servers (Section 4.1) may be used. The transition probabilities are:

$$p_{S,S} = \overline{P1}_{S,D} \tag{67}$$

$$p\prime_{S,S} = \overline{P1}_{S,D}.. \tag{68}$$

All the performance metrics are obtained as already explained for type I $C-ARQ$ SW protocol.

[1] This latency does not take into account the re-sequencing delay that may be incurred due to out-of-order delivery of frames at D.

An alternative way to evaluate the performance metrics is to take advantage of the mean residual time approach for the independent server model (Section 4.2). The expected frame latency is obtained from the final expression of (46) by removing the last term (i.e., instantaneous feedback) and by setting $P = \overline{P1}_{S,D}$:

$$T = \frac{T_s}{2(1-\rho)} + \frac{\rho}{1-\rho}\frac{PT_s}{1-P} + \frac{T_s}{1-P}. \tag{69}$$

From the expression of T, the other performance metrics are easily derived. The saturation throughput, i.e., Th, is

$$E[t_r] = \sum_{k=1}^{\infty} kP^{k-1}(1-P) = \sum_{k=0}^{\infty}(k+1)P^k(1-P) = \frac{1}{1-P}$$
$$Th = \frac{1}{E[t_r]} = (1-P). \tag{70}$$

5.2.2 Type II $H-ARQ$ Stop and Wait

For type II $H-ARQ$ protocol, the queueing system with mutually exclusive servers (Section 4.1) is used, and

$$p_{S,S} = \overline{P1}_{S,D} \tag{71}$$

$$p\prime_{S,S} = \frac{\overline{P2}_{S,D}}{\overline{P1}_{S,D}}.. \tag{72}$$

All the performance metrics are obtained as already explained for type I $C-ARQ$ SW protocol.

6 A Study Case

In this section a study case is presented in which the delay models are used to compare the performance of the six ARQ protocols.

6.1 Assumptions on Radio Channel with Coding

The following assumptions are made on the redundancy code and the radio channel propagation properties. These assumptions are used consistently for all the ARQ protocols.

Path loss and fading affect the transmission of both data and incremental redundancy frames. Frequency-flat, block-Rayleigh fading (quasi-static) is assumed with fading level that is constant over the duration of an entire frame transmission, i.e., time slot. The fading levels are statistically independent of the time slot, channel, and space. (These assumptions tend to favor the non-cooperative protocols, as in reality it is expected that cooperative protocols are more robust over non-cooperative protocols when fading is correlated in time, due to their spatial diversity [13].) The instantaneous SNR from node i to j is:

$$\gamma_{ij} = \frac{E_{b_i}}{N_0} \cdot K \cdot l_{ij}^{\beta} \cdot \alpha_{ij}^2, \tag{73}$$

whereby the following definitions are used:

- E_{b_i}: transmitted energy per bit at node i,
- N_0: noise spectral density of the Additive White Gaussian Noise (AWGN) channel,
- K: path loss for an arbitrary reference distance,
- l_{ij}: distance from node i to j (normalized to the reference distance),
- β: path loss exponent,

- α_{ij}: Rayleigh distributed random variable to model the Rayleigh fading magnitude from node i to j, $E[\alpha_{ij}^2] = 1\ \forall i$.

Each payload is encoded into a codeword using the puncturing technique based on a rate-compatible punctured convolutional code (RCPC) [19]. The codeword is partitioned to form two frames, i.e., the data frame of N_1 bits and the incremental redundancy frame of N_2 bits. To fit the time slot nature of the channel, $N_1 = N_2$. The N_1 bits of the data frame constitute a valid (albeit weaker) codeword. Note that before encoding, the payload is matched with a CRC code that is used at both the destination and relay to verify whether or not the received frame is decoded correctly.

The transmission error probability of a data frame sent from i to j is evaluated conservatively using the union bound technique [20, 7], i.e.,

$$P1_{i,j} \leq 1 - \left(1 - \min\left\{1, \sum_{d=d_f}^{\infty} \mathrm{a}_d \cdot P(d|\gamma_{ij})\right\}\right)^B, \tag{74}$$

whereby the following definitions are used:

- B: number of payload and CRC bits in each data frame, i.e., number of trellis branches in the codeword,
- d_f: free distance of the code [21],
- c_d: spectrum of the code [19], i.e., number of codewords of weight d,
- $P(d|\gamma_{ij})$: probability that a wrong path at distance d is selected.

Averaging (74) over the probability density function of the instantaneous SNR, i.e., $f(\gamma_{ij})$,

$$\overline{P1}_{i,j} \leq \int_0^{\infty} P1_{i,j} \cdot f(\gamma_{ij})\ \partial\gamma_{ij}. \tag{75}$$

Assuming that binary PSK with soft decoding is employed,

$$P(d|\gamma_{ij}) = Q\left(\sqrt{2 \cdot d \cdot \gamma_{ij}}\right), \tag{76}$$

where $Q(\cdot)$ is the Marcum Q function [22] and d is the weight of the codeword.

Recall that the joint decoding of both data and incremental redundancy frames takes place only when the data frame alone cannot be successfully decoded. The probability of not being able to decode the payload successfully after receiving the incremental redundancy frame from node $j \in \{S, R\}$ is upper bounded by

$$P2_{j,D} \leq 1 - \left(1 - \min\left\{1, \sum_{d_S=d_f}^{\infty} \sum_{d_j=d_{f2}-d_S}^{\infty} c_{d_S,d_j} \cdot P(d_S + d_j|\gamma_{SD}, \gamma_{jD})\right\}\right)^B \tag{77}$$

$$\overline{P2}_{j,D} \leq \int_0^{\infty} \int_0^{\infty} P2_{j,D} \cdot f(\gamma_{SD})\, f(\gamma_{jD})\, \partial\gamma_{SD}\, \partial\gamma_{jD} \tag{78}$$

whereby the following definitions are used:

- d_{f2}: free distance of the parent code [21],
- c_{d_S,d_j}: spectrum of the code, i.e., number of codewords of weight d_S in the first N_1 bits, and weight d_j in the other N_2 bits,
- $P(d_S + d_j|\gamma_{SD}, \gamma_{jD})$: probability that a wrong path at distance $d_S + d_j$ is selected, i.e.,

$$P(d_S + d_j|\gamma_{SD}, \gamma_{jD}) = Q\left(\sqrt{2d_S\gamma_{SD} + 2d_j\gamma_{jD}}\right). \tag{79}$$

6.2 Results

The first part of this section is devoted to the comparison of the ARQ protocols' performance. The second part presents more results on type II $C-ARQ$ protocols, given their superior performance.

The system parameters are set as follows: $T_s = 1$, $K = 60$ dB, $\beta = 4$, and $l_{S,D} = 1$. Payload and CRC comprise 128 bits 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) [19].

Unless otherwise indicated, R is at half distance between S and D, i.e., a good location for successful cooperation.

Simulation results have confidence interval values of 10% or better, at 95% confidence level. In the simulation, frame error probabilities are given by (74) and (77), using the instantaneous value of Rayleigh fading. For the analytical model, Monte Carlo integration is used to estimate the time invariant error frame probabilities.

6.2.1 Performance Comparison of ARQ Protocols

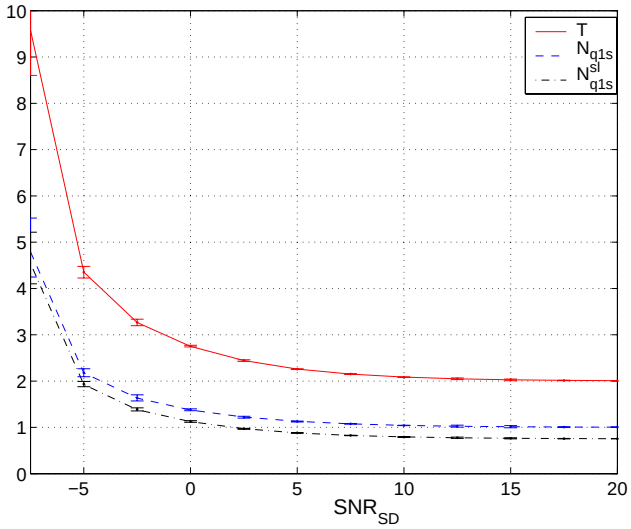


Figure 11: Type II $C-ARQ$ SR: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

Figs. 11-16 plot the expected frame latency, i.e., T, the expected buffer occupancy at S, i.e., N_{q1s} and N_{q1s}^{sl}, for various signal-to-noise ratio values on the S to D channel, i.e., SNR_{SD}. Analytical and simulations results are shown for all six ARQ protocols, assuming a $\lambda = 0.5$ arrival rate.

Fig. 17 plots the expected frame latency, i.e., T, versus the arrival rate, i.e., λ, for all six protocols, when the average SNR between S and D is SNR_{SD} = 3dB. Fig. 18 plots the saturation throughput, i.e., Th, versus the signal-to-noise ratio SNR_{SD} for all six protocols. Analytical and simulation results are compared against each other. Analytical results are plotted with dashed (type I) and solid (type II) lines. Both figures quantify the superiority of both the cooperative protocols, and the selective repeat option. The already known advantage of type II protocols over type I protocols is also clearly documented in the plots. The match between the analytical results and the simulation results supports the correctness of the models, under different load conditions.

6.2.2 Further Results on Type II $C-ARQ$ SW Protocol

Figs. 19-25 plot type II $C-ARQ$ SW protocol performance as a function of the R coordinates. S and D are fixed and located as shown in the figures. The arrival rate at source S is $\lambda = 0.5$ and the channel SNR_{SD} is 0 dB.

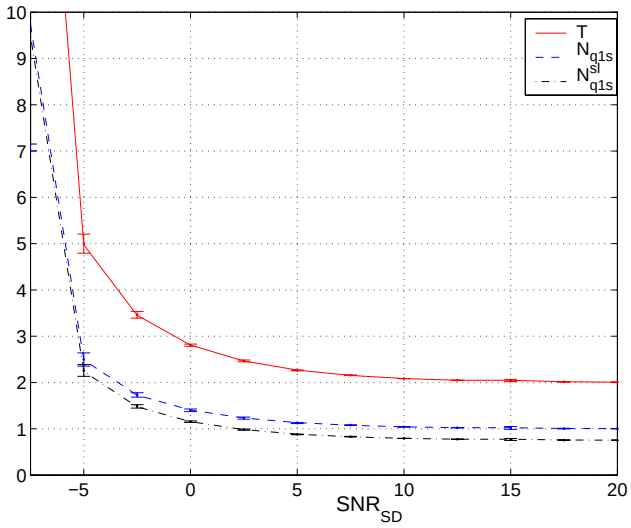


Figure 12: Type I $C-ARQ\ SR$: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

Fig. 19 plots the expected latency, T, of type II $C-ARQ\ SW$. Latency is minimized when R is in the region centered around the midpoint between S and D. As R moves away from this central region, the latency grows due to the increased number of retransmissions, until the system becomes unstable. In the presence of a better $S-D$ channel quality or a lower value of λ the stable region expands.

Figs. 20 and 21 plot the saturation throughput, Th, of type II $C-ARQ\ SW$. Saturation throughput is maximized when R is in the region centered around the midpoint between S and D. As R moves away from this region the saturation throughput decreases because the number of retransmissions grows. Note that in the region behind S the throughput first decreases, then it increases again. In the low throughput region just behind S the $S-R$ channel SNR is high, while the $R-D$ channel SNR is not, i.e., R decodes successfully the frames from S, but its retransmissions to D are not very successful. As R moves further away from S and D, also the $S-R$ channel quality decreases to the point where R is not correctly receiving the frames from S, and the ARQ protocol behaves like a non-cooperative one.

Figs. 22 and 23 plot the saturation throughput difference between type II $C-ARQ\ SW$ over type II $H-ARQ$. This plot indicates when the cooperative ARQ is advantageous (positive values) over the non-cooperative ARQ, and vice versa (negative values). The cooperative ARQ yields up to 6.5% throughput gain when compared to the non-cooperative ARQ. This gain is even more significant as the $S-D$ channel SNR decreases.

Figs. 24 and 25 plot the saturation throughput difference between type II $C-ARQ\ SW$ and type I $H-ARQ$. The results are consistent with those shown in Fig. 23.

6.2.3 Further Results for Type II $C-ARQ\ SR$ Protocol

Figs. 26-32 plot type II $C-ARQ\ SR$ protocol performance as a function of the R coordinates. S and D are fixed and located as shown in the figures. The arrival rate at source S is $\lambda = 0.5$ and the channel SNR_{SD} is 0 dB.

Fig. 26 plots the expected latency of type II $C-ARQ\ SR$. The pattern is similar to that of type II $C-ARQ\ SW$ (see Fig. 19), while the absolute values are sensibly lower and the stability region is larger.

Figs. 27 and 28 plot the saturation throughput, Th, of type II $C-ARQ\ SR$. Once again, saturation throughput is maximized when R is in the central region between S and D.

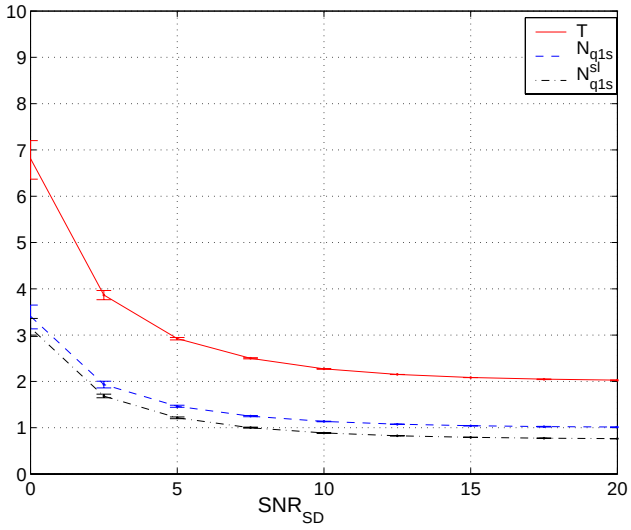


Figure 13: Type II $C-ARQ\ SW$: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

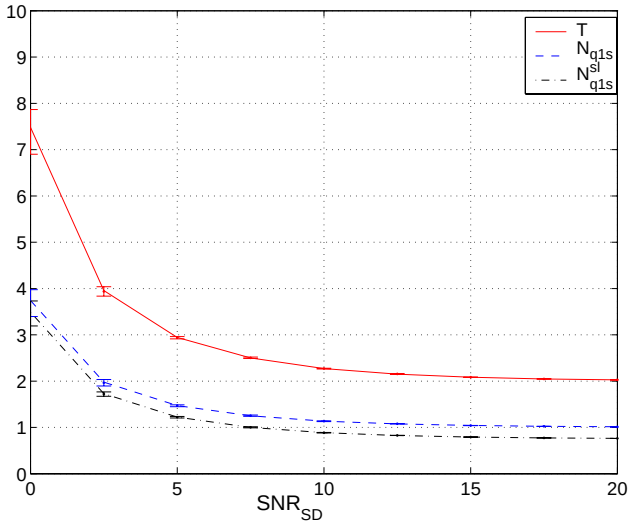


Figure 14: Type I $C-ARQ\ SW$: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

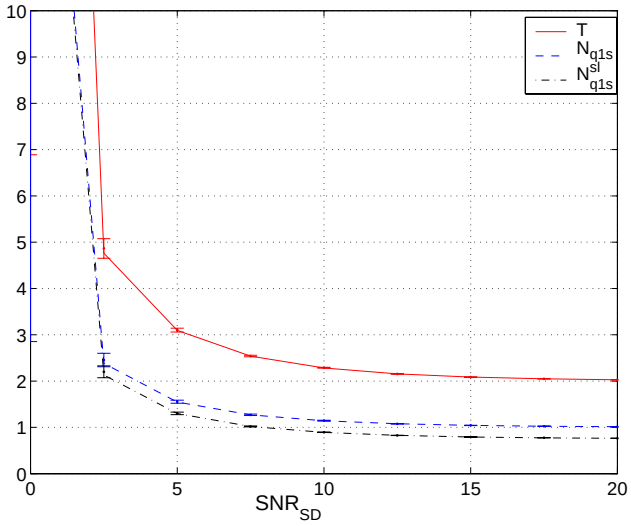


Figure 15: Type II $H-ARQ$: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

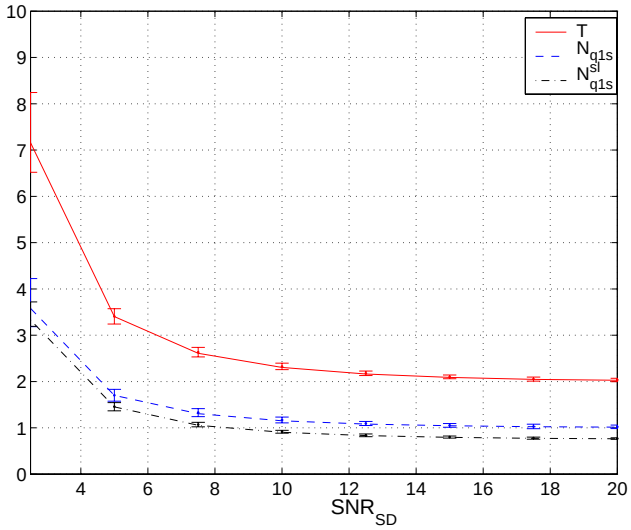


Figure 16: Type I $H-ARQ$: T, N_{q1s}, and N_{q1s}^{sl} vs. SNR_{SD} (dB), $\lambda = 0.5$

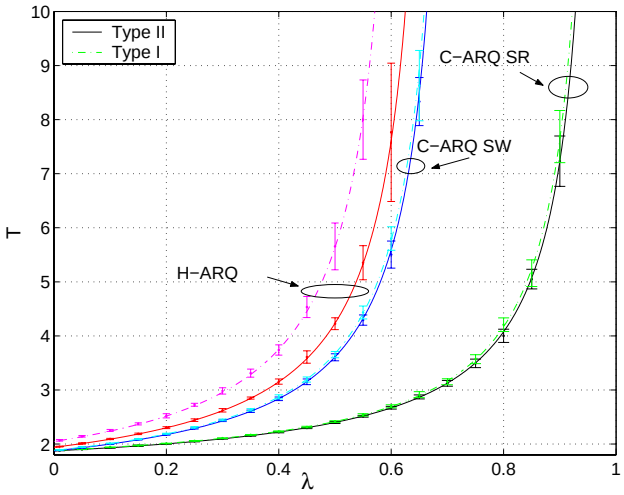


Figure 17: T vs. λ, $SNR_{SD} = 3$dB

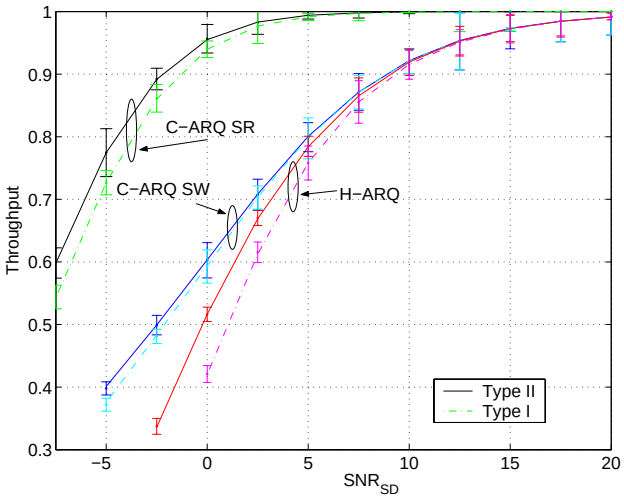


Figure 18: Th vs. received SNR_{SD} (dB)

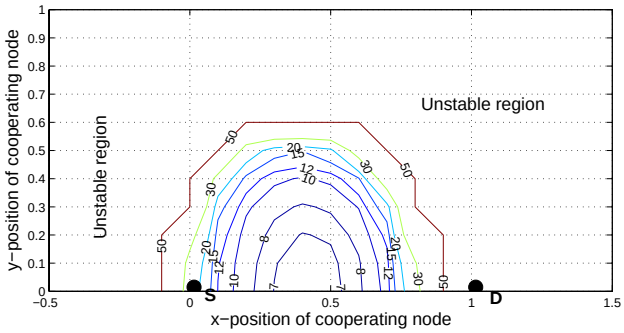


Figure 19: Type II $C-ARQ\ SW$: T vs. position of node R

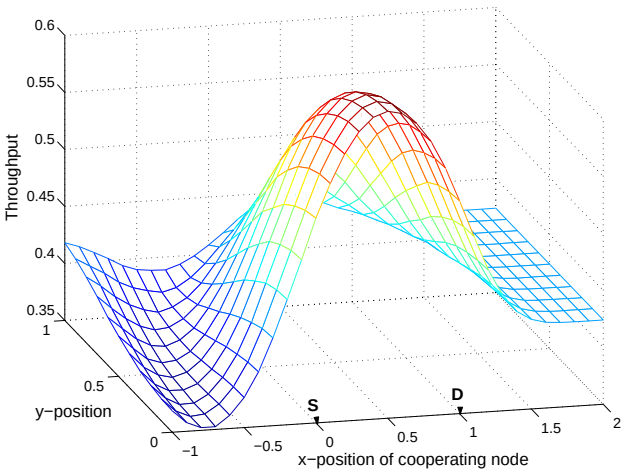


Figure 20: Type II $C-ARQ\ SW$: Th vs. position of node R

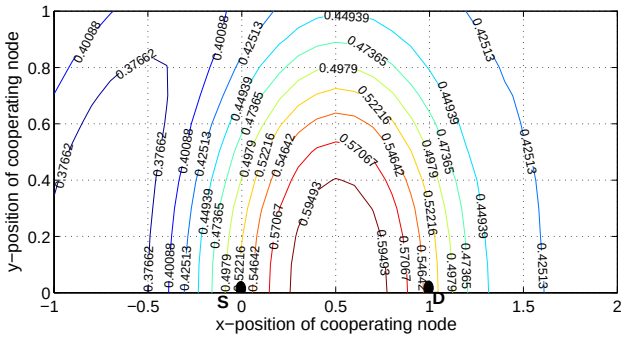


Figure 21: Type II $C-ARQ\ SW$: Th vs. position of node R

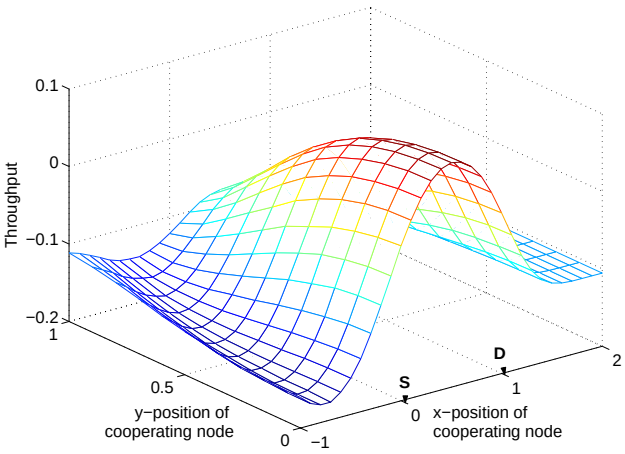


Figure 22: Difference of Th of type II $C-ARQ\ SW$ and type II $H-ARQ$ vs. position of node R

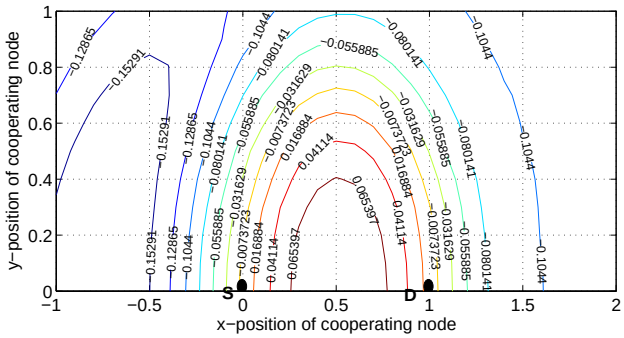


Figure 23: Difference of Th of type II $C-ARQ\ SW$ and type II $H-ARQ$ vs. position of node R

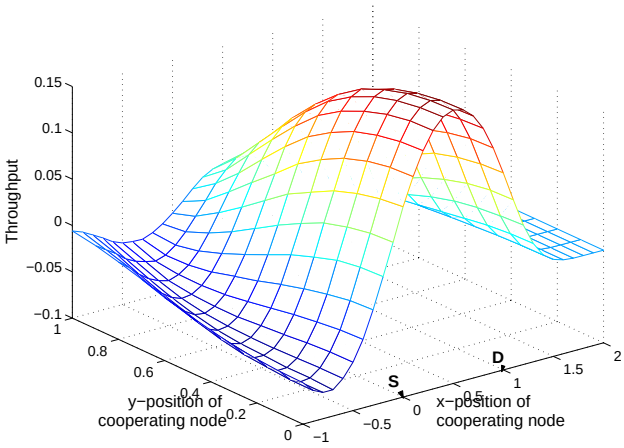


Figure 24: Difference of Th of type II $C-ARQ\ SW$ and type I $H-ARQ$ vs. position of node R

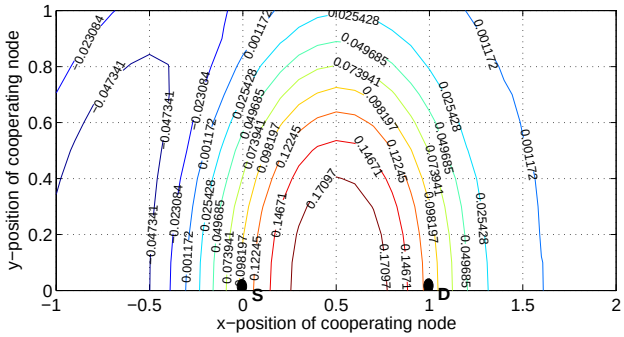


Figure 25: Difference of Th of type II $C-ARQ\ SW$ and type I $H-ARQ$ vs. position of node R

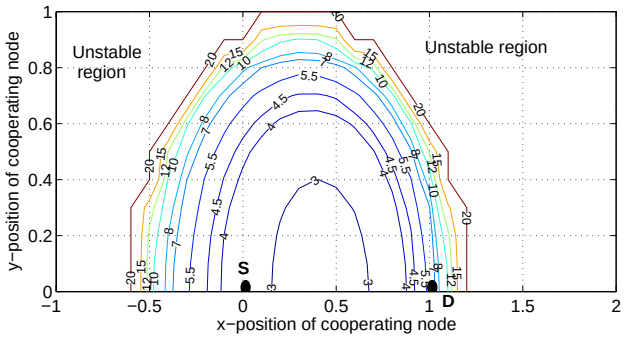


Figure 26: Type II $C-ARQ\ SR$: T vs. position of node R

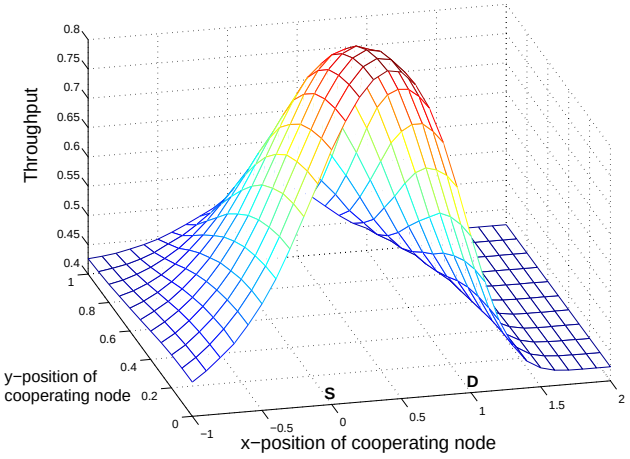


Figure 27: Type II $C-ARQ\ SR$: Th vs. position of node R

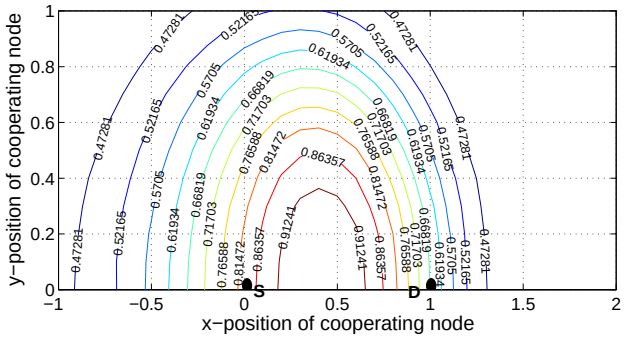


Figure 28: Type II $C-ARQ\ SR$: Th vs. position of node R

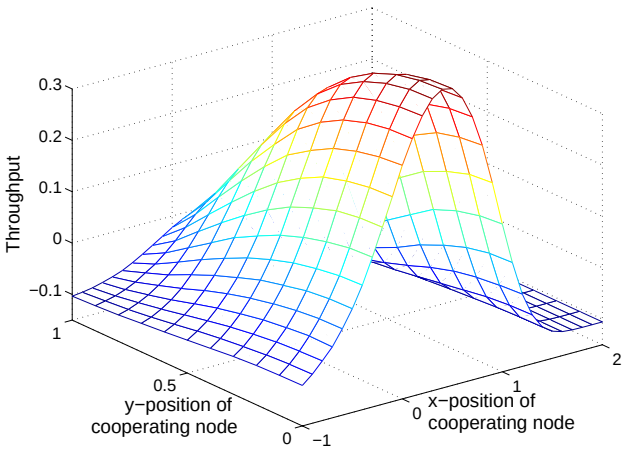


Figure 29: Difference of Th of type II $C-ARQ$ SR and type II $H-ARQ$ vs. position of node R

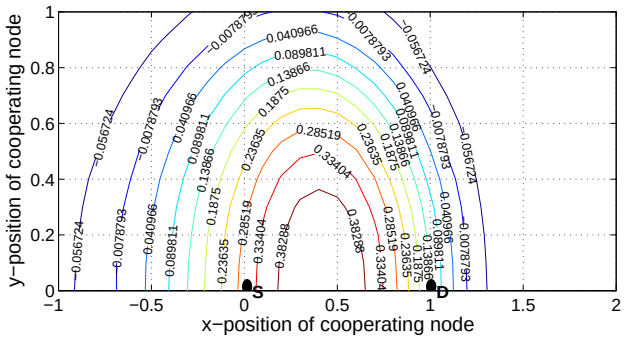


Figure 30: Difference of Th of type II $C-ARQ\ SR$ and type II $H-ARQ$ vs. position of node R

Figs. 29 and 30 plot the saturation throughput difference between type II $C-ARQ\ SR$ and type II $H-ARQ$. Type II $C-ARQ\ SR$ yields up to 38% gain when R is well positioned. The throughput gain region of $C-ARQ\ SR$ over $H-ARQ$ is larger than that of type II $C-ARQ\ SW$ and extends beyond S and D.

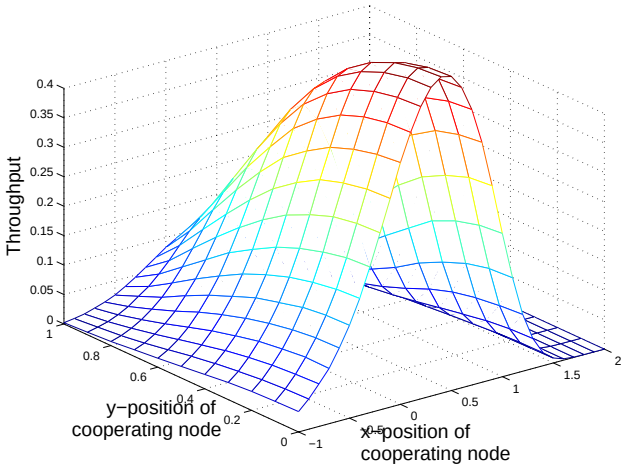


Figure 31: Difference of Th of type II $C-ARQ\ SR$ and type I $H-ARQ$ vs. position of node R

Figs. 31 an 32 plot the saturation throughput difference between type II $C-ARQ\ SR$ and type I $H-ARQ$. The throughput gain is up to 48%.

6.2.4 Re-sequencing Delay

In the $C-ARQ\ SR$ protocols, frames may be received out-of-order at D. Fig. 33 shows the impact of the re-sequencing delay at D for varying values of SNR_{SD}, when R is at half distance between S and D. The arrival rate is $\lambda = 0.5$. he curves without the re-sequencing delay are obtained using the analytical model. The curves with the re-sequencing delay are obtained through simulations. The re-sequencing delay has a small impact on the overall delay even at low SNR.

7 Conclusion

The first delay model for single-source and single-relay cooperative ARQ protocols was presented in this paper. The analytical model was used to show numerically how cooperative ARQ protocols cope with the radio channel noise and fading. The model provides encouraging results, indicating under what conditions the cooperative ARQ protocols are superior when compared to non-cooperative ARQ protocols. Both frame latency and saturation throughput may be improved by using cooperative ARQ protocols. Equivalently,

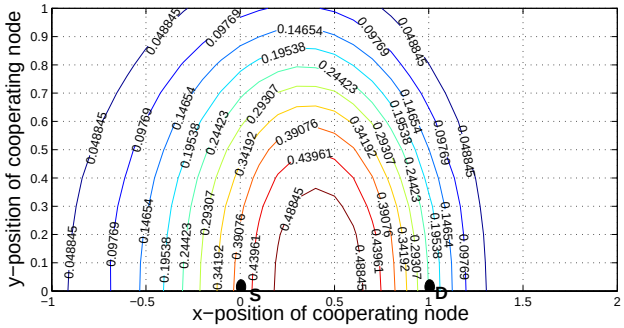


Figure 32: Difference of the saturation throughput of type II $C-ARQ\ SR$ and type I $H-ARQ$ as a function of the position of node R

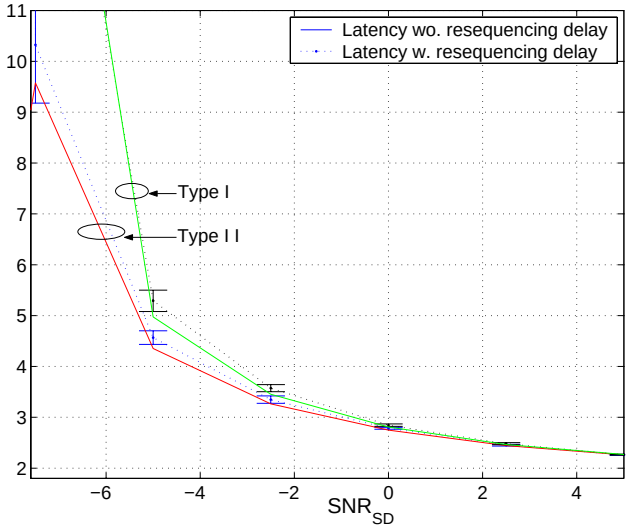


Figure 33: $C-ARQ\ SR$ protocols: T + re-sequencing delay at D vs. SNR_{SD}

cooperative ARQ protocols may reduce the SNR that is required to meet the desired throughput and frame latency. The latter option may be appealing in applications where the signal power is limited, e.g., some types of wireless sensor networks [23].

The results presented in this paper are just a modest contribution to the understanding of cooperative ARQ protocols. Further work is required in this field to consolidate and generalize these initial findings. For example, how is double-source cooperative ARQ going to work? What other frame transmission policies can be used effectively? What happens if source and relay share the same channel and collisions may occur? An initial attempt to address some of these open questions can be found in [14]. However, it is clear that much more work is required in this field.

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] P. Smyth, *Mobile and Wireless Communications: Key Technologies and Future Applications*. IEE, 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Elsevier Computer Networks*, vol. 38, no. 4, pp. 393–442, March 2002.

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

[4] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, 2004, accepted for publication.

[5] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[6] 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*, Washington, 2001, p. 294.

[7] 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*, vol. 52, no. 2, pp. 362–371, Feb. 2004.

[8] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–Part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, 2003.

[9] ——, "User cooperation diversity–Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, 2003.

[10] 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.

[11] T. E. Hunter and A. Nosratinia, "Cooperative diversity through coding," in *Proc. IEEE ISIT*, Laussane, 2002, p. 220.

[12] E. Zimmermann, P. Herhold, and G. Fettweis, "On the performance of cooperative relaying protocols in wireless networks," *European Transactions on Telecommunications (ETT)*, vol. 16, no. 1, pp. 17–35, 2005.

[13] B. Zhao and M. C. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7 – 18, Jan. 2005.

[14] P. Gupta, I. Cerutti, and A. Fumagalli, "Three transmission scheduling policies for a cooperative ARQ protocol in radio networks," in *Proc. WNCG conference*, October 2004.

[15] S. Lin, D. Costello, and M. Miller, "Automatic-repeat-request error-control schemes," *IEEE Communications Magazine*, vol. 22, no. 12, 1984.

[16] S. Lin and P. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Trans. on Comm.*, vol. 30, no. 7, pp. 1701–1719, 1982.

[17] Y.-M. Wang and S. Lin;, "A modified selective-repeat type-II hybrid ARQ system and its performance analysis," *IEEE Trans. on Comm.*, vol. 31, no. 5, pp. 593–608, 1983.

[18] L. Kleinrock, *Theory, Volume 1, Queueing Systems*. John Wiley and Sons, 1975.

[19] J. Hagenauer, "Rate-compatible punctured convolutional codes (rcpc codes) and their applications," *IEEE Trans. on Comm.*, vol. 36, no. 4, pp. 389–400, 1988.

[20] E. Malkamaki and H. Leib, "Evaluating the performance of convolutional codes over block fading channels," *IEEE Trans. on Infor. Theory*, vol. 45, no. 5, pp. 1643–1646, 1999.

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

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

[23] M. Tacca, P. Monti, and A. Fumagalli, "Cooperative and non-cooperative ARQ protocols for microwave recharged sensor nodes," in *Proc. 2nd European Workshop on Wireless Sensor Networks (EWSN)*, 2005.

Appendix

To solve (30) for $E[\tilde{q}]$, it is necessary to evaluate:

$$\lim_{n\to\infty}\left(E[q_n\triangle_{Pq_{n-1}}]-E[\triangle_{q_n}\triangle_{Pq_{n-1}}]\right). \tag{80}$$

In general, assuming that the service time of the bottom server is δ time slots, (80) becomes:

$$\lim_{n\to\infty}\left(E[q_n\triangle_{Pq_{n-\delta}}]-E[\triangle_{q_n}\triangle_{Pq_{n-\delta}}]\right), \tag{81}$$

while the other terms of (30) remain unchanged. By using (23) and (26), it is easy to see that for $\delta=0$ (zero service time at the bottom server, i.e., instantaneous feedback), (80) becomes $(E[\tilde{q}]-E[\tilde{q}]P)$. For any other value of $\delta\neq 0$, additional calculations need to be carried. Only case $\delta=1$ is addressed here.

Let $S_{i^{(n)},j^{(n)}}$ be the state of the embedded Markov chain at time n. Let $\pi_{i,j}^{(n)}=P_r\{i^{(n)}=i,j^{(n)}=j\}$ be the state probability of the embedded Markov chain at time n. The first term in (80) can be rewritten as:

$$\begin{aligned}E[q_n\triangle_{Pq_{n-1}}]&=\sum_{k=1}^{\infty}kP\cdot P_r\{i^{(n)}=k,i^{(n-1)}\geq 1\}=\\&=P\sum_{k=1}^{\infty}k\left(\pi_{k,0}^{(n)}+\pi_{k,1}^{(n)}\right)-P\sum_{k=1}^{\infty}ka_k\pi_{0,0}^{(n-1)}-P\sum_{k=1}^{\infty}ka_{k-1}\pi_{0,1}^{(n-1)}.\end{aligned} \tag{82}$$

Assuming that the system is ergodic, $\lim_{n\to\infty}\pi_{i,j}^{(n)}=\pi_{i,j}$, and (82) becomes:

$$\begin{aligned}\lim_{n\to\infty}E[q_n\triangle_{Pq_{n-1}}]&=P\cdot E[\tilde{q}]-P\cdot\sum_{k=1}^{\infty}k\frac{(\lambda T_s)^k}{k!}e^{-\lambda T_s}\pi_{0,0}-P\cdot\sum_{k=1}^{\infty}k\frac{(\lambda T_s)^{k-1}}{(k-1)!}e^{-\lambda T_s}\pi_{0,1}=\\&=PE[\tilde{q}]-P\pi_{0,0}\lambda T_s-P\pi_{0,1}(1+\lambda T_s)=PE[\tilde{q}]-P\pi_{0,0}\rho(1-P)-P\pi_{0,1}(1+\rho(1-P)),\end{aligned} \tag{83}$$

which is the result used in (31).

The second term is (80) can be rewritten as:

$$\begin{aligned}E[\triangle_{q_n}\triangle_{Pq_{n-1}}]&=1\cdot P\cdot P_r\{q_n>0,q_{n-1}>0\}=\\&=P\cdot\left(1-\pi_{0,0}^{(n-1)}-\pi_{0,1}^{(n-1)}-P_r\{\pi_{0,0}^{(n)}|\pi_{1,0}^{(n-1)}\}\pi_{1,0}^{(n-1)}-P_r\{\pi_{0,1}^{(n)}|\pi_{1,0}^{(n-1)}\}\pi_{1,0}^{(n-1)}\right).\end{aligned} \tag{84}$$

In (84), $P_r\{q_n>0,q_{n-1}>0\}$ is the probability that $i^{(n)}>0$ and $i^{(n-1)}>0$.

Assuming that the system is ergodic and using (15) the equation in (84) becomes:

$$\lim_{n\to\infty}E[\triangle_{q_n}-\triangle_{Pq_{n-1}}]=P\left(\rho-(1-P)a_0\pi_{1,0}-Pa_0\pi_{1,0}\right)=P\left(\rho-a_0\pi_{1,0}\right), \tag{85}$$

which is the result used in (32).