On the Delay Statistics of SR ARQ Over Markov Channels With Finite Round-Trip Delay

Michele Rossi, *Member, IEEE*, Leonardo Badia, *Member, IEEE*, and Michele Zorzi, *Senior Member, IEEE*

***Abstract*—In this paper, the packet delay statistics of a fully reliable selective repeat automatic repeat request (SRARQ) scheme is investigated. The study is carried out assuming that the packet error process can be characterized by means of a discrete time Markov channel. The packets transmitted by the sender are checked for errors at the receiver's side, and acknowledgment messages (ACK or NACK), assumed error free, are sent back accordingly. It is assumed that the feedback message is known at the transmitter *m* channel slots after the packet transmission started. First, an analytical exact approach is described, in which an appropriate Markov model is developed in order to find the exact delay statistics. This allows to write close formulas related to the delivery delay experienced by ARQ packets. Moreover, in order to reduce the computational complexity of this analysis, an approximate model is presented. The results obtained from the approximate approach are in good agreement with the ones derived from the exact analysis.**

***Index Terms*—Automatic repeat request, data communication, delay estimation, error analysis, Markov processes, modeling.**

I. Introduction

MODERN communication systems are now facing an increasing development of multimedia applications. In this scenario, effective error control techniques are required, especially for error-prone channels, like wireless links. In fact, many multimedia applications are very sensitive to channel impairments, which may occur with higher probability than the application can tolerate. Thus, the need for good performance in terms of data reliability, latency, and efficient bandwidth usage implies a careful design of error control techniques as well as a deep understanding of the underlying mechanisms.

In this work, we focus on automatic repeat request (ARQ) techniques, whose study in recent years is enjoying renewed popularity. Link layer retransmission techniques are already used, in combination with physical layer forward error correction (FEC), by the universal mobile telecommunications system (UMTS) and general packet radio service (GPRS) [1]. In these systems, ARQ is coupled with FEC-like strategies to ensure data integrity while keeping both acceptable delays and tolerable consumed energy levels. ARQ strategies are also used for Bluetooth piconets [2], and analogous solutions are considered in mobile satellite communication systems [3], where the higher level retransmissions, e.g., transmission control protocol (TCP), and the physical layer processing often do not suffice to achieve acceptable performance.

Another fact showing the importance of having an accurate description of the ARQ recovery process is that the performance of higher level packets, in terms of delay/jitter and error probability, is directly affected by its behavior. In other words, a correct ARQ setting is key in achieving the needed higher level quality of service. Hence, an exact study of the delivery delay process at the ARQ level is pivotal to accurately assess the performance perceived by the user.

In ARQ, the transmitter sends packets (protocol data units, PDUs) consisting of payload and error detection codes. At the receiver side, based on the outcome of the error detection procedure, acknowledgment messages are sent back to the transmitter (ACK or NACK according to the result of error detection). The sender performs packet retransmissions based on such feedback. In general, ARQ protocols are variants of the following basic schemes: stop-and-wait (SW); go-back-*N* (GBN); and selective repeat (SR). In SW, only one packet in a round-trip time is transmitted, i.e., a new packet is transmitted only when the ACK of the current one is received. This scheme is not very efficient, especially when the round-trip delay is large. In GBN, packets are transmitted continuously, without waiting for acknowledgment messages. When a NACK is received, the transmitter simply retransmits the erroneous packet and all the subsequent ones. The SR scheme is the most efficient: Here, packets are transmitted continuously, and only negatively acknowledged packets are retransmitted, i.e., retransmissions are selectively triggered by NACK messages. When the round-trip delay goes to zero all the schemes become identical. In the literature [4], [5], this situation is referred to as an "ideal" SR ARQ.

The overall PDU delay with ARQ protocol can be subdivided in three contributions. These quantities will be referred to as queueing delay, transmission delay, and resequencing delay, as usually done in the literature [6], [7]. The first is the time spent in the source buffer queue, i.e., the time between the instant of the first PDU transmission over the channel and its release by higher levels. This term depends on both the channel behavior and the PDU arrival process. The second contribution is the time between the first transmission and the correct reception of the PDU, which only depends on the channel behavior. The

Manuscript received August 18, 2003; revised April 17, 2004; accepted February 14, 2004. The editor coordinating the review of this paper and approving it for publication is Y. Fang.

M. Rossi and L. Badia are with the Department of Engineering, University of Ferrara, 44100 Ferrara, Italy (e-mail: mrossi@ing.unife.it; lbadia@ing.unife.it).

M. Zorzi is with the Department of Engineering, University of Ferrara, 44100 Ferrara, Italy and also with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: zorzi@ing.unife.it).

Digital Object Identifier 10.1109/TWC.2005.850316

last term is the time spent in the receiver resequencing buffer. In fact, even though the sender transmits packets in order, they can arrive out of sequence due to random errors and consequent retransmissions. Hence, a correctly received PDU must wait in the receiver resequencing buffer until all the PDUs with lower identifier have been correctly received. This last term is the most complicated because it depends on errors experienced by other PDUs. In the following, the term "resolution" (of a packet) will mean correct reception, whereas "delivery" (or equivalently, "release") refers to the joint resolution of the considered packet as well as of all packets with a smaller identifier (id). In this paper, we are interested in the statistics of the delivery delay, defined as the time between the first transmission of the packet and its successful release from the resequencing buffer; in other words, the sum of the second and third terms.

Several studies have been performed on the delay performance of the SR protocol over a wireless channel [5]–[12]. In [8], Konheim proposes an analytical model for deriving the distribution of packet delay and buffer occupancy at the transmitter of an SR ARQ control system and considering a static channel. In [9], an alternative approach for the same problem considering a Bernoulli arrival process is proposed by Anagnostou and Protonotarios. Rosberg and Shacham [10] and Rosberg and Sidi [11] analyzed in detail resequencing delay and resequencing buffer occupancy at the transmitter and at the transmitter and the receiver jointly, respectively, but again in the independent error case. The impact of time-varying channel was considered for example in [13], [14], though for SW and GBN only. An analysis of correlated errors for SR ARQ by means of flow graph theory was presented for the first time by Lu and Chang [12], where both Markovian models for the channel and gap functions are considered. The scope of the investigation is mainly about the throughput, and it is proven that the performance of ARQ is not degraded (in certain cases, it is even improved) in the presence of channel correlation. Fantacci [5] carried out another analysis considering a nonstationary channel and deriving mean values for the queue length at the transmitter and for the packet delay. Exact quantities are obtained for the ideal SR, whereas approximated formulas are given to match the finite round-trip delay case. Another approximate analysis is presented by Chang and Yang [6], with the additional hypothesis of an adaptation in the SR ARQ protocol. Here, an estimate of the total end-to-end delay is presented. Finally, Kim and Krunz [7] accounted for a time-varying channel and a finite round-trip delay by developing a detailed analysis for all the ARQ delay contributions. However, several approximations are introduced, for example, the hypothesis of ideal SR is used for the queueing delay evaluation, so that only approximate mean values for the three components can be quantified.

The goal of this paper is to extend all the above contributions, by considering as general a scenario as possible. Hence, we study the delay performance of the SR ARQ scheme, considering both time-varying channel and finite round-trip time, taking into account the effect of bursty channel errors. Moreover, our goal is to provide exact expressions, rather than approximations, where possible. In particular, we present an exact analysis for the delivery delay statistics, which has never been obtained before. This contribution greatly extends the existing literature since we do not give just approximate mean values, that in certain cases could be misleading, but we derive the complete statistical description, so that the performance can be exactly evaluated. In the following, some assumptions will be made to simplify the formal description. However, note that the generality of the results is not affected. In fact, the analytical method that we present does not critically rely on any of these assumptions, which are chosen only for the sake of clarity in the presentation. This means that, if needed, every hypothesis can be relaxed to extend the model to more complicated cases (at the only price of having a more cumbersome analysis, but without substantial changes in the formal approach).

As shown next, an exact analysis has, however, the drawback of a notable computational complexity. Nevertheless, we solve this drawback with another contribution, which is an approximate evaluation of the delivery delay statistics. This is obtained using the same approach adopted for the exact analysis, but keeping the complexity linear in the round-trip delay value, rather than exponential. We also prove that, in spite of this adjustment, the approximate approach achieves results that are in excellent agreement with the exact ones. Thus, also in the approximate analysis, the novelty of our contribution is maintained. The proposed approach is computationally convenient, even for very long round-trip delays. Hence, it enables the computation of the delivery delay statistics in resource-limited wireless systems.

The remaining part of the paper is organized as follows: In Section II, the ARQ policy and the channel model are described; in Section III, the exact analysis of the delivery delay is reported; the complexity of the resulting Markov chain is exponential in m. In Section IV, we report an approximate approach for the computation of such statistics by keeping the model complexity linear in m. Section V reports some results, and finally, in Section VI, some conclusions are given.

II. Model for ARQ Processes

We consider a pair of nodes that communicate through a noisy wireless link using a fully reliable link layer protocol (unlimited retransmission attempts) to counteract channel impairments. We assume that both transmitter and receiver have unlimited buffer size and adopt the SR ARQ protocol at the link layer. This implies that data packets (ARQ PDUs) and ACKs/NACKs flow in the forward and backward direction, respectively.

Let us assume that the wireless channel is characterized by means of a two-state discrete time Markov chain (DTMC), with states 0 and 1. We assume that transmissions in state 1 are always erroneous, whereas state 0 is error-free. This is a reasonable assumption in many cases [4]. From a theoretical point of view, even such a simple model is able to give the necessary insight for the analysis. However, if desired, this can be extended to account for a more complex system, like a higher order Markov chain [12], without changing the nature of the analytical approach.

The channel transition probability matrix $\mathbf{P}$ and the corresponding i-step transition probability matrix $\mathbf{P}(i)$ are as follows:

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}, \quad \mathbf{P}(i) = \mathbf{P}^i = \begin{pmatrix} p_{00}(i) & p_{01}(i) \\ p_{10}(i) & p_{11}(i) \end{pmatrix}. \quad (1)$$

From this equation, the steady-state channel error probability and the average error burst length can be derived as $\varepsilon = p_{01}/(p_{10} + p_{01})$ and $b = 1/p_{10}$, respectively.

Moreover, our analysis can be easily extended to account for erroneous ACKs/NACKs. In this case, it is reasonable to assume that forward and backward error processes are independent. Then, if the ACK error process is independent identically distributed (i.i.d.), the extension is trivial and consists of accounting for the single ACK error probability $\varepsilon_{\mathrm{ACK}}$ so that every packet correctly delivered on the forward link has a probability $\varepsilon_{\mathrm{ACK}}$ of not being acknowledged. Instead, in the case where the backward channel has memory, its error process can be tracked, e.g., by introducing a second transition matrix and considering a packet transmission as successful only when both the forward and the backward channels are error-free. This only leads to complications in the calculus without changing the formal approach. An example of the usage of the feedback channel matrix is given in [15]. As a side remark, note that in real systems, feedback packets are usually assumed to be error-free since they are much shorter than data packets. In addition, they are often protected with coding techniques, thereby making the impact of ACK/NACK errors negligible as shown in [16].

We consider the SR ARQ scheme as a generalization of the protocol described in [17]. In this system, the sender continuously transmits new PDUs from its buffer in increasing numerical order as long as ACKs are received. For the purpose of analysis, we assume that time is slotted with the time slot duration corresponding to the constant transmission of a PDU. We also assume that the value of the round-trip time is an appropriate integer number of slots m, as commonly done in the literature [18], where m is referred to as the ARQ window size. Thus, the sender receives the ACK/NACK message for each packet after the transmission of $m - 1$ subsequent PDUs, which can be new PDUs or retransmissions of erroneous packets. Due to the SR procedure, m slots after each transmission, i.e., when the feedback packet is received, the PDU is immediately retransmitted in case of NACK, else a new PDU is sent.

The main goal of this paper is to give an explicit expression for the computation of the delivery delay statistics (sum of transmission and resequencing delay). In particular, our contribution is to jointly track the Markov channel evolution and the correct reception of the packets. This study only marginally depends on the arrival process. In particular, the delivery statistics is influenced by the number and the position of the packets which can block the in-order delivery. However, once this information is given, the analysis presented in the following can be applied without substantial changes.

For what concerns the traffic model, one could observe that it mainly affects the queueing delay, that is out of the scope of our analysis, whereas the delivery delay only slightly depends on it.[1] So, it is reasonable to consider a simple model for the arrival process, although our analysis could again be extended with a more complicated scheme if necessary. Hence, we suppose that once a PDU is correctly transmitted, a new one is always present in the source buffer. This model is referred to in the literature [7] as "Heavy Traffic" condition, and describes exactly a continuous packet source. Thus, it holds, for example, for a TCP file transfer protocol (FTP)-like session or video/audio continuous data streaming. Reliable ARQ almost completely avoids TCP timeouts (when the channel error rate is not too large) and the TCP level, after filling the bandwidth-delay product, behaves as a continuous packet source (the TCP window size is not decreasing because error recovery is never triggered). Should the Heavy Traffic assumption not be verified, the delivery delay computed with it can be seen as a good approximation. However, note that an evaluation relaxing this hypothesis, omitted here due to space constraints, would still be possible. In fact, Yoshimoto *et al.* [19] already gave useful considerations about the effect of more complicated arrival processes in SR ARQ systems, in particular, the average queue length is analytically evaluated. By means of this contribution and considering an approach similar to the one followed in [11], it is possible to generalize the statistics derived in the present paper to arbitrarily complicated cases.

III. Computation of the Delivery Delay Statistics

In this section, we compute the delivery delay statistics of SR ARQ for a single PDU. The basic idea is to mark a PDU of interest, called in the following tagged PDU, and to track its successful delivery. Note that this implies that all previous PDUs must be jointly tracked as well, as they must also be correctly received before the tagged PDU can be released (in order) to the higher layers.

Let $t = m$ be the slot in which the tagged PDU is transmitted for the first time, which implies that a successful transmission occurred a round-trip time earlier, i.e., in slot $t = 0$ (otherwise, a retransmission would take place in slot m). The m-sized window from slot 1 to slot m will be called fundamental window, due to the important role that it plays in the analysis. The tagged PDU is finally released when all PDUs in the fundamental window are correctly received. It can be shown that these are the only ones that need to be considered, i.e., PDUs that are not transmitted in the fundamental window play no role in the delivery of the tagged PDU. (A formal proof of this statement, as well as a more precise treatment of the following discussion, can be found in [20].)

Note that due to the finite round-trip time, there is a time difference between the transmission process at the sender and the reception process at the destination, which depends

[1] This behavior has been observed by simulation and is not reported here due to space constraints.

on the propagation and processing delays involved. While we take this explicitly into account in studying the effect of successive retransmissions (which must be separated by the round-trip time of m slots), in order to simplify the notation, we ignore it in the definition of the delay. That is, if all PDU transmissions are successful, the delay in our analysis is zero (which means that a PDU is correctly received and passed up to the higher layers with the minimum possible delay). The neglected contribution t_c, which has a constant value and is approximately equal to $m/2$, can be added to find the absolute delay values. Thus, in the following, we will study the statistics $P_d[k]$, defined as the probability that the delivery delay equals k slots plus the constant term t_c.

In view of the above discussion, the problem to be solved is therefore to find the time that it takes for all PDUs transmitted in slots 1 through m to be eventually received correctly, given that a successful transmission occurred in slot 0. When this happens, the tagged PDU is finally released to the higher layers.

Consider the transmission of the fundamental window, following the (necessarily successful) transmission in slot 0. Some of the slots in $\{1, 2, \ldots, m\}$ are successful, and we denote them as resolved (a resolved slot corresponds to a PDU that does not need retransmission). Note that if slot i is resolved, all slots $i + \kappa m$, with κ a positive integer, will correspond to the transmission of PDUs whose id is higher than the tagged PDU, and therefore can be ignored for our purposes. Thus, if slot i is resolved, all slots $i + \kappa m$, with κ a positive integer, will also be denoted as resolved (regardless of the channel state).

On the other hand, a slot of the fundamental window in which a transmission failure occurs is denoted as unresolved, which means that a retransmission of the failed PDU must take place m slots later. If the retransmission is successful, the slot becomes resolved (and all future slots at m-slot intervals will also be resolved), otherwise, it will remain unresolved until eventually a successful retransmission occurs.

In this setting, whenever we have that m consecutive slots are marked as resolved, the whole fundamental window will be correctly received, and therefore the tagged PDU can be released to the higher layers. That is, the tagged PDU in-order delivery delay is the time from its first transmission (slot m) to the time when the last unresolved slot becomes resolved.

As an example, let $m = 3$ and suppose that a good channel state at $t = 1$ is followed by a burst of four erroneous slots and then the channel is again good for three more slots. The algorithm gives: 1 = resolved; 2 = unresolved; 3 = unresolved; 4 = resolved (despite the channel error, as it was previously marked); 5 = unresolved; 6 = resolved; 7 = resolved; 8 = resolved. After slot 8, there is no need to go further, since every slot in a higher position is marked as resolved. This means that the delivery of the tagged PDU occurs in slot 8. Note in fact that slot 8 is the last slot of the first sequence of three slots comprising all resolved slots (starting in slot 6).

To evaluate when such a sequence of m consecutive resolved slots occurs, we proceed slot by slot, as in the example above. At any given time t, we need to know the status of slots $t - m + 1, t - m + 2, \ldots, t - 1$. We do so by keeping a bitmap $\mathbf{b}$, where $b_k = 1$ if slot $t - m + 1 + k$ is still unresolved, and $b_k = 0$ otherwise, for $k = 0, 1, \ldots, m - 2$. For convenience, in the analysis we will also use an integer representation of $\mathbf{b}$, $i = \sum_{k=0}^{m-2} b_k 2^k$.

In addition to keeping memory of the past $m - 1$ slots, we also need to specify the status of the current slot, i.e., slot t. In this case, a binary variable is no longer sufficient, since we also need to track the channel state, which is necessary to determine the future evolution of successful transmissions. Notice that the Markovian nature of the channel evolution makes it possible to ignore the channel state in slots $t - m + 1$, $t - m + 2, \ldots, t - 1$ once the channel state in t is known. The only channel state required for the analysis is thus that of the current slot. Three situations are possible: the channel is good, which implies that the slot is resolved (if it was not resolved already, the good channel state makes it resolved now); the channel is bad and the slot is resolved (in a previous transmission); the channel is bad and the slot is still unresolved. These three possibilities will be denoted by 0, 1, and 2, respectively, and the associated variable will be denoted by ω.

Consider now the random process $X(t) = (i(t), \omega(t))$, which jointly tracks slot by slot the Markov channel evolution and the status of the m latest slots. This process is a Markov chain. The full resolution of the fundamental window corresponds to the first transition of this Markov chain to states (0,0) or (0,1). In fact, the current slot is resolved if $\omega \in \{0, 1\}$ and unresolved if $\omega = 2$, whereas the other slots are all resolved if $i = 0$. This will be taken into account next in the closed-form computation of the delivery delay.

In order to determine the possible transitions $X(t) \to X(t + 1) = (i', \omega')$ and the corresponding transition probabilities, suppose that at time t the bitmap $\mathbf{b}$ is $(b_0, b_1, \ldots, b_{m-2})$, where the most significant bit b_{m-2} denotes the status of the most recent among the past slots. At time $t + 1$ this bitmap is clocked one position into the past, i.e., $\mathbf{b}' = (b'_0, b'_1, \ldots, b'_{m-3}, b'_{m-2}) = (b_1, b_2, \ldots, b_{m-2}, f(\omega))$, where $f(\omega) = 1$ if $\omega = 2$ (current slot at time t was still unresolved), and $f(\omega) = 0$ if $\omega = 0$, 1. More compactly, in this case $f(\omega) = \lfloor \omega/2 \rfloor$.

Regarding the value of $\omega' = \omega(t + 1)$, note the following. If $b_0 = 0$ at time t, the corresponding slot has already been resolved, and therefore $\omega' = 0$ or 1 according to the channel state at time $t + 1$. On the other hand, if $b_0 = 1$, the slot is still unresolved at time t, hence, we have $\omega' = 0$ if the channel at time $t + 1$ is good (slot is resolved at this time), and $\omega' = 2$ otherwise (slot remains unresolved). There are only two possible destinations for $X(t + 1)$, given $X(t)$, since the shift of the bitmap is deterministic and the only random variable is the channel state, which can assume two values. More precisely, the transition probabilities are given as follows:

- if i is even (i.e., $b_0 = 0$), then

$$P\big[X(t+1) = (i', \omega') | X(t) = (i, \omega)\big] = \begin{cases} p_{xy} & \text{if } i' = \left\lfloor \frac{i}{2} \right\rfloor + \left\lfloor \frac{\omega}{2} \right\rfloor 2^{m-2}, \\ & x = \left\lceil \frac{\omega}{2} \right\rceil,\ \omega' = y,\ y = 0, 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

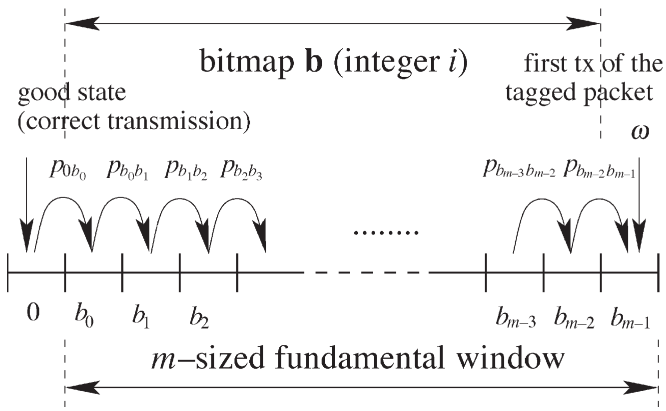


Fig. 1. Diagram for the computation of the system starting probabilities.

- if i is odd (i.e., $b_0 = 1$), then

$$P\big[X(t+1) = (i', \omega') | X(t) = (i, \omega)\big] = \begin{cases} p_{xy} & \text{if } i' = \lfloor \frac{i}{2} \rfloor + \lfloor \frac{\omega}{2} \rfloor 2^{m-2}, \\ & x = \lceil \frac{\omega}{2} \rceil,\ \omega' = 2y,\ y = 0, 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where the use of $\omega' = 2y$ in the latter case means that a good channel $y = 0$ leads to $\omega' = 0$, whereas a bad channel $y = 1$ leads to $\omega' = 2$, i.e., the situation of bad channel and unresolved slot. According to the rule above, the transition probability matrix can be built, with two nonzero entries per row.

In order to find the delay statistics, we proceed as follows. Let $\boldsymbol{\pi} = [\pi_0\ \pi_1 \ldots \pi_{K-1}]$ be a $1 \times K$ vector whose $K = 3 \cdot 2^{m-1}$ entries represent the probabilities that the system starts in a given state. $\boldsymbol{\pi}$ is computed as follows:

$$\begin{aligned} &\text{if } \omega \text{ is even } (0,\ 2): \quad \pi_{(i,\omega)} = p_{0b_0} \left[\prod_{j=1}^{m-2} p_{b_{j-1}b_j} \right] p_{b_{m-2}\frac{\omega}{2}} \\ &\text{if } \omega \text{ is odd } (1): \quad \pi_{(i,\omega)} = 0. \end{aligned} \quad (4)$$

In order to explain (4), consider Fig. 1, where the first transmission of the tagged packet is depicted along with its fundamental window. First, in slot m, it is impossible to have $\omega = 1$, i.e., the event corresponding to bad channel state and resolved slot. In fact, the tagged packet in that slot is at its first transmission attempt and the slot label (resolved/unresolved) will therefore correspond to the channel state at time m (good/bad). Hence, we only have to consider the remaining cases $\omega \in \{0, 2\}$. Given that the probability that the system starts in a state (i, ω) is a product of channel transition probabilities (matrix $\mathbf{P}$). In particular, the channel starts with a good state (slot 0), then evolves according to the channel states in slots 1 through $m - 1$ (represented by b_j's) and ends in either good or bad state if $\omega = 0$ or $\omega = 2$, respectively. Hence, the last channel transition is $b_{m-2} \to \omega/2$.

Let $\boldsymbol{e}_0$ be a column vector of all zeros except for the entries corresponding to states (0,0) and (0,1), which are equal to 1. According to what was explained above, these are the only two states where the fundamental window is resolved. If $\boldsymbol{T}$ is the transition matrix of the Markov chain $X(t)$, we can find the probability that the delivery delay is less than or equal to $t_c + k$ as

$$\mathcal{P}_c[k] = \boldsymbol{\pi} \boldsymbol{T}^k \boldsymbol{e}_0, \quad k \geq 0 \quad (5)$$

and $P_d[k]$ is determined as

$$P_d[0] = \mathcal{P}_c[0], \quad P_d[k] = \mathcal{P}_c[k] - \mathcal{P}_c[k-1] \quad \forall k > 0. \quad (6)$$

Another interesting distribution is the cumulative complementary distribution of the delivery delay statistics, ccdf$[k]$, defined as

$$\text{ccdf}[k] = \text{Prob}\{delay > t_c + k\} = 1 - \mathcal{P}_c[k]. \quad (7)$$

This value, which can be directly derived in our model, represents the probability that the delivery delay exceeds k slots. Henceforth, it has great importance in estimating whether delay constraints are met for real-time applications.

IV. Approximation of the Delivery Delay Statistics

Even though the analysis above is exact, it has the main drawback of having a complexity that is exponential in m. Hence, for large values of the round-trip delay, the computation of the exact statistics becomes both memory and time expensive. In this section, we propose a simple approximation that allows us to reduce the computational complexity, enabling the computation of the statistics for large m. In the following, we will introduce heuristic assumptions with the goal of simplifying the analysis without losing adherence to the problem.

In particular, instead of tracking the state and the position of each unresolved PDU, we propose to analyze the resolving process by just tracking the number of remaining slots to be resolved. We consider rounds of m slots and build a Markov chain in which the state is represented by $X(r) = (n_e(r), C(r))$, $r \geq 1$, where $n_e(r)$ and $C(r)$ are the number of unresolved slots in round r and the channel state in the last slot of that round, respectively, i.e., we only track the number of still unresolved PDUs in every round. This Markov chain evolves round by round, i.e., m slots at a time. We define round 1 as the one comprising slots 1 through m. Moreover, in order to keep track of errors and corrections, i.e., to compute $X(r)$ from $X(r-1)$, we need to position the $n_e(r-1)$ unresolved PDUs in round $r-1$ (to derive the number of slots that are still unresolved in round r). To this end, several strategies are possible. As an example, one could randomly distribute the $n_e(r-1)$ unresolved PDUs inherited from the previous round $(r-1)$ according to a given probability distribution. On the other hand, one could think of placing these packets following a predefined deterministic pattern. In this sense, we checked several possible approximations such as the uniform random placement and the deterministic placement. Moreover, regarding the last one, we tried both the uniform and the bursty positioning methods. At the end, we selected the bursty deterministic strategy because of its simplicity and its good agreement with the exact analysis. In this method, the unresolved slots are deterministically placed at the end of every round. By the definition of the process $X(r)$ and the introduction of these approximations, we have been able to neglect the exact position of the unresolved slots and to achieve a complexity linear in m.

Let slot k be the slot where the fundamental window is resolved. Here, we refer to the round containing this slot as the final round, whereas the previous round is referred to as prefinal. To derive $P_d[k]$, we proceed as follows. We first consider $X(r)$ and the resolving strategy discussed above to derive the distribution for $n_e(r)$ up to and including the prefinal round. Then, we apply a further resolution strategy for the last round only. This last resolution strategy is also deterministic and is needed to account for the relative position (within the prefinal round) of the last unresolved PDU, which equals the relative position where the tagged packet is finally released in the final round. In the following, we first present the procedure considered to derive the distribution for $n_e(r)$, i.e., the analysis considered up to and including the prefinal round.

Let $\varphi_{ij}(s,n)$, $i, j \in \{0,1\}$ be the probability that s slots in $\{1,2,\ldots,n\}$ are successful and the channel state is j at time n, given that the channel state was i at time 0. This is a well-known function that can be derived in recursive [21] or close [22] form.

Now, let $\Psi_{0j}(e,r)$ be the probability of having e $(1 \le e \le m)$ unresolved slots in round r and that the channel in the last slot of round r is j given that the channel in slot 0 was correct. This function can be computed recursively as follows:

$$\Psi_{0j}(e,r) = \begin{cases} \varphi_{0j}(m-e,m) & r=1 \\ \sum_{x=e}^{m} \sum_{i\in\{0,1\}} \Psi_{0i}(x,r-1)\mathcal{R}_{ij}(x-e,x) & r>1 \end{cases} \tag{8}$$

where

$$\mathcal{R}_{ij}(q,x) = \begin{cases} \varphi_{ij}(q,x) & x=m \\ \sum_{c\in\{0,1\}} p_{ic}(m-x)\varphi_{cj}(q,x) & x<m. \end{cases} \tag{9}$$

The function $\mathcal{R}_{ij}(q,x)$ is used to compute the probability that q slots out of x are resolved in round r, and that the channel state in the last slot of round r is j given that the channel state in the last slot of the previous round $(r-1)$ was i, and that all unresolved slots are deterministically grouped at the end of each round. Note that the recursive expression (8) is initialized (round $r=1$) by exploiting the knowledge of the channel at time 0 and computing the mean probability of having e erroneous transmissions in any order in that round. Moreover, the probability to have e unresolved slots, $1 \le e \le m$, at the end of the generic round r, $r>1$ $(\Psi_{0j}(e,r))$ is obtained by considering the probability of having x $(\Psi_{0i}(x,r-1),\ e \le x \le m)$ unresolved slots after $r-1$ rounds and of having exactly $x-e$ of these slots resolved in round r $(\mathcal{R}_{ij}(x-e,x))$. Finally, these probabilities are summed over $e \le x \le m$ and over $i \in \{0,1\}$ to account for all the feasible values of unresolved slots and channel states at the end of round $r-1$. Observe that the probability of being in state $X(r) = (n_e(r), C(r))$, $1 \le n_e \le m$, $C \in \{0,1\}$, $r \ge 1$ is given by $\Psi_{0C}(n_e, r)$.

Next, we report the approaches followed to track the resolution taking place in the final round, i.e., where all the PDUs in the fundamental window are eventually resolved. To this end, we write slot k as $k = \xi m + \eta$, where $\xi \ge 0$ and $1 \le \eta \le m$ are the number of full rounds and the number of slots in the current round covered by k, respectively. With this decomposition, slot k belongs to round $\xi + 1$ (final round), whereas ξ is the previous (prefinal) round. Note that, in what follows, k is the absolute time index, whereas η is the relative slot position within the final round.

In the following, we discuss an approximate approach for the evaluation of the probability $P_d[k]$. As a first step, using the function $\Psi_{0j}(e,\xi)$, we evaluate the probability that the last PDU is resolved in round $\xi+1$. Note that this is the probability of having $e>0$ outstanding errors in round ξ and zero in round $\xi+1$. This evaluation assumes that the e outstanding errors correspond to consecutive slots within a round. Conditioned on the last PDU being resolved in round $\xi+1$, we can approximate the probability distribution of the specific slot in round $\xi+1$ in which this resolution takes place by just assuming that all positions of the error burst within the round are equally likely.

More specifically, we adopt two slightly different approaches. In the first approach, called burst at the end (BE), we first compute the probability that $1 \le e \le m$ erroneous packets $(\Psi_{0j}(e,\xi))$ are inherited from the prefinal round ξ, and we evaluate the probability of resolving these PDUs by grouping them into a single burst at the end of round $\xi+1$, i.e., in positions $m-e+1$ through m of that round. Note that following this procedure, the exact position of the erroneous PDUs is not tracked and what we obtain is the approximate probability for the release of the tagged PDU in round $\xi+1$, $P_{\xi+1}$. However, what we need is not just the probability of releasing the tagged PDU in the final round $(P_{\xi+1})$, but the probability that the tagged packet is released in the ηth $(1 \le \eta \le m)$ slot of that round, i.e., slot $k = \xi m + \eta$ in absolute terms. We derive this probability by *a posteriori* distributing $P_{\xi+1}$ among the slots of round $\xi+1$. To do that, we need to specify the distribution of the slot in the last round where the tagged packet is resolved. Here, we consider that η is uniformly distributed in the last round. The full BE method is presented in detail in Section IV-A.

In the second approach, named in the sequel shifted burst (SB), we evaluate the probability of resolving the burst of consecutive erroneous slots inherited from the previous round (ξ) in a uniformly chosen position of the final round. Unlike in the BE approach, where the e-sized burst could not be resolved in slot $\eta < e$, in this case, we allow any value for η (between

1 and m). If η is smaller than the size of the burst of errors, we admit that the burst is cyclically shifted. For example, if the last position of a burst of length 3 is the second slot, we split the shifted burst in two subburst sequences such that the erroneous slots will be the first two and the last one. Note that the burst can be in any position with equal probability.

The difference between the two strategies above is in the order between the evaluation of the resolution probability and the uniform distribution assumption among the slots in round $\xi+1$. BE first evaluates the probability of resolving the errors concentrated in a burst at the end of the final round $\xi+1$; after that, this probability is uniformly distributed among the possible slots in that round. SB, instead, first conditions on η in round $\xi+1$, and then sums the contributions of all the m-cyclically shifted bursts where the last erroneous packet occupies position $k=\xi m+\eta$ to give the resolution probability in that slot.

Let us define the function $\rho(\cdot)$ as the probability that the channel in the last slot of round ξ is j and that exactly e $(1\le e\le m)$ slots are yet to be resolved after round ξ and that these e slots are all resolved in round $\xi+1$ (final round) given that the last unresolved slot is in position p, $e\le p\le m$, and that a successful transmission occurred at time 0. Formally

$$\rho(j,e,\xi|p)=\Psi_{0j}(e,\xi)p_{j0}(p-e+1)p_{00}^{e-1}. \qquad (10)$$

This function is the probability of resolving a burst accounting for its length (e), the position occupied by its last element (p), and the channel state (j) at the end of round ξ. In the following, we report the detailed description of the two approaches.

A. Burst at the End

The delivery delay statistics is computed in the following way:

$$P_d[k]=\begin{cases}\dfrac{1}{m}\displaystyle\sum_{j\in\{0,1\}}\sum_{e=1}^{\eta}\rho(j,e,\xi|m) & \eta\neq m\\ \dfrac{1}{m}\displaystyle\sum_{j\in\{0,1\}}\sum_{e=1}^{m}\rho(j,e,\xi|m)e & \eta=m\end{cases} \qquad (11)$$

where $k=\xi m+\eta$, with $\xi\ge 0$ and $1\le\eta\le m$.

The probability of the first passage through states $(0,C)$ in round $\xi+1$, i.e., for $k\in\{\xi m+1,\ldots,\xi m+m\}$ is computed by considering all the unresolved slots inherited from round ξ to be placed at the end of round $\xi+1$. To compute this probability, we use the $\rho(\cdot)$ function by summing all and only the contributions for which $e\le\eta$. For a given η, the cases where $e>\eta$ are indeed unfeasible. Moreover, this probability is subdivided among slots $\xi m+\eta,\forall\,\eta\in\{1,\ldots,m\}$ by assuming that the final unresolved slot is distributed by means of the function $\mathcal{P}_E(\eta|e)$ that is the approximate probability that the burst ends in position η given that it consists of e slots. $\mathcal{P}_E(\eta|e)$ is defined a priori as follows:

$$\mathcal{P}_E(\eta|e)=\begin{cases}0 & \eta<e\\ \dfrac{1}{m} & e\le\eta<m\\ \dfrac{e}{m} & \eta=m.\end{cases} \qquad (12)$$

To sum up, in this approach, we first resolve all the unresolved slots inherited from the previous round ξ by deterministically grouping them at the end of the round. Thus, the last unresolved slot is always in position m of each round. What we obtain in this way is the probability of release of the tagged packet in round $\xi+1$. After that, we subdivide such probability among all slots $\xi m+\eta$, with $1\le\eta\le m$ by considering the distribution $\mathcal{P}_E(\eta|e)$.

B. Shifted Burst

In this approach, we also consider that the unresolved slots inherited from round ξ are grouped in a single burst, but unlike in BE, we evaluate the resolution of such a burst assuming that its last element can be cyclically shifted between positions 1 and m of round $\xi+1$. We first condition on η, considering $P[\eta=i]=1/m$, $1\le i\le m$. Then, once η has been fixed, we compute the resolution probability for the e still unresolved slots in round ξ by summing the probabilities of the events leading to a resolution of the burst in slot η. Note that unlike in BE, the e-sized burst is resolved by considering its last element to be placed in position η. With this assumption, the delivery delay statistics for $k=\xi m+\eta$ can be written as

$$P_d[k]=\begin{cases}\dfrac{1}{m}\displaystyle\sum_{j\in\{0,1\}}\sum_{e=1}^{\eta}\rho(j,e,\xi|\eta) & \eta\neq m\\ \dfrac{1}{m}\displaystyle\sum_{j\in\{0,1\}}\sum_{e=1}^{m}\Bigg[\rho(j,e,\xi|m) & \\ \qquad\qquad +\displaystyle\sum_{y<e}\rho'(j,e,\xi|y)\Bigg] & \eta=m\end{cases}$$

where

$$\rho'(j,e,\xi|\,p)=\Psi_{0j}(e,\xi)p_{j0}p_{00}^{p-1}p_{00}(m-e+1)p_{00}^{e-p-1}. \qquad (13)$$

With (13), we compute, for each $\eta\in\{1,\ldots,m\}$ in round $\xi+1$, the probability of resolving the burst given that its last slot is in position η, where $\eta\ge e$. Thus, when $\eta\ge e$, we guarantee that the first slot of the burst is in position f with $f\ge 1$. For the correction of the burst of unresolved slots, see the function $\rho(j,e,\xi|p)$ with $p=\eta$.[2]

When $\eta=m$, the contributions of m-cyclically shifted versions of the burst are also considered. In more detail, we split the e-sized (where e are the residual errors from the previous

[2]The same function has been used in the BE approach, but with $p=m$.

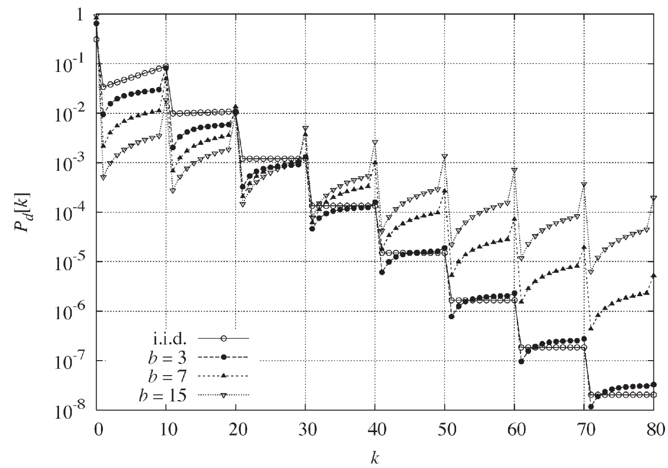
round ξ) burst in two parts where the first part is composed by the y slots in positions $\{1, \ldots, y\}$, whereas the second part is composed by the $e - y$ slots in positions $\{m - e + y + 1, \ldots, m\}$.[3] This burst of unresolved slots is then resolved by means of the function $\rho'(j, e, \xi|y)$. ρ' is used to evaluate the probability that the channel in the last slot of round ξ is j and that exactly e $(1 \leq e \leq m)$ slots are to be resolved at the end of round ξ and that these e erroneous packets are all resolved in round $\xi + 1$ given that they are subdivided in two bursts, where the first one occupies positions $\{1, \ldots, y\}$ and the second occupies positions $\{m - e + y + 1, \ldots, m\}$.

V. Results

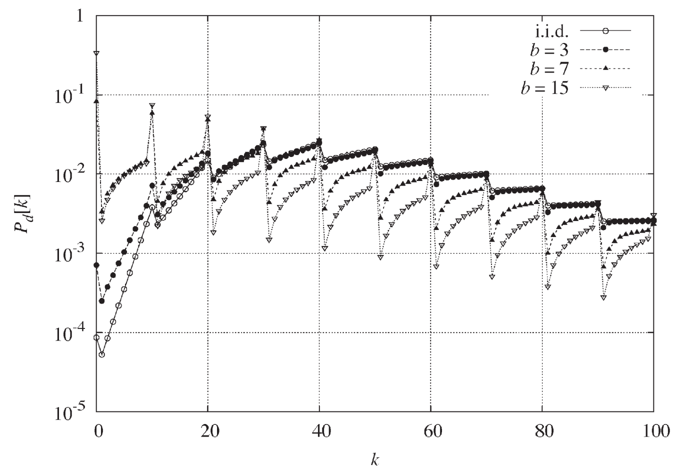
The delivery delay statistics $P_d[k]$ has been computed according to the above analysis, for various values of the channel error probability ε and of the channel burstiness b. To test the accuracy, we used a simulator in which we implemented the simple transmission of packets with an SR ARQ scheme applied to the same scenario; thus, we empirically measured the delivery delay statistics, in addition to deriving them from the exact analysis.

In Fig. 2, we evaluate $P_d[k]$ and compare the case of i.i.d. channel with different values of the correlation b. In any case, the shape of the delivery delay statistics presents a stepwise behavior with a logarithmic-constant gap after every positions κm, κ integer. Moreover, in both the i.i.d. and the correlated error cases, the resolution probability presents an increasing behavior with the maximum placed at the end of each round. In the i.i.d. case, this effect is due to the larger number of combinatorial events leading to the resolution of an m-sized window in the last slot. This behavior vanishes after a few rounds, where $P_d[k]$ becomes almost constant within a given m-sized window. In addition to this, over bursty channels, another effect is present. In fact, when the channel correlation is large, an erroneous tagged PDU transmission at time m likely comes with the erroneous transmission of the $m - 1$ packets in positions 1 through $m - 1$. In this case, all packets are likely released in the subsequent error-free period and this leads to a larger probability of resolving the window at the end of a round.

An interesting value is $P_d[0]$. When the delivery delay is 0, the tagged packet is released at the end of the slot in which it is received, i.e., both the tagged packet transmission and the $m - 1$ previous transmissions are correct. In this case, the transmission plus propagation delay is equal to about half the round-trip delay, and the resequencing delay is zero. It can be observed that $P_d[0]$ in the bursty case is higher, due to the higher probability of having a whole window of correct slots when errors occur in bursts. This phenomenon has a larger impact if the error probability is high, see Fig. 2(b). As a matter of fact, the probability of delivering the tagged PDU within a delay equal to m can be heavily underestimated by



(a)



(b)

Fig. 2. $P_d[k]$, comparison between the i.i.d. channel and a correlated channel with $b = \{3, 7, 15\}$ by considering $m = 10$. (a) $\varepsilon = 0.1$. (b) $\varepsilon = 0.6$.

considering the independent error model when the channel is in fact bursty. The value of $P_d[0]$ has a great impact on the throughput. Thus, we can say that Fig. 2 confirms the results presented in [12], where the throughput performance of ARQ on a correlated channel is shown to be better than in the case of static channel. On the other hand, the slope of each curve decreases with increasing b, that is, the larger the bursts, the higher the probability of having large delays. In fact, on average, the channel recovers from an error, i.e., is restored into the good state, after a number of slots equal to b. Thus, for high k, we see an increase of the probability of large delivery delays due to the channel burstiness.

In Fig. 3(a), $P_d[k]$ is plotted in the i.i.d. case for various values of ε. When ε is small, the maximum value of $P_d[k]$ is in $k = 0$. As ε increases, the maximum of $P_d[k]$ shifts to the right. The same behavior can be observed when the channel is correlated, see Fig. 3(b). Note the difference in shape for each round between the two cases.

The different behavior when the channel is correlated can also be observed by looking at Fig. 4(a), where the mean

[3]Note that, for these bursts, the mth slot of round ξ is always erroneous. That is the reason why they are assigned to the case where the resolution occurs in position $\eta = m$.

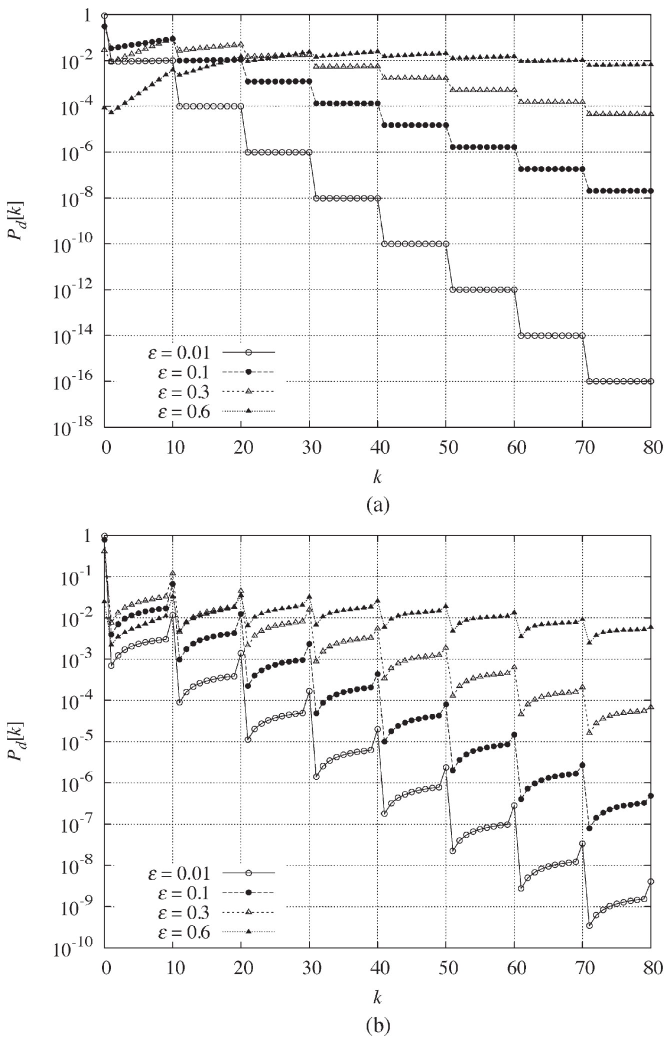


Fig. 3. $P_d[k]$ as a function of ε by considering $m = 10$. (a) i.i.d. channel. (b) Correlated channel with $b = 5$.

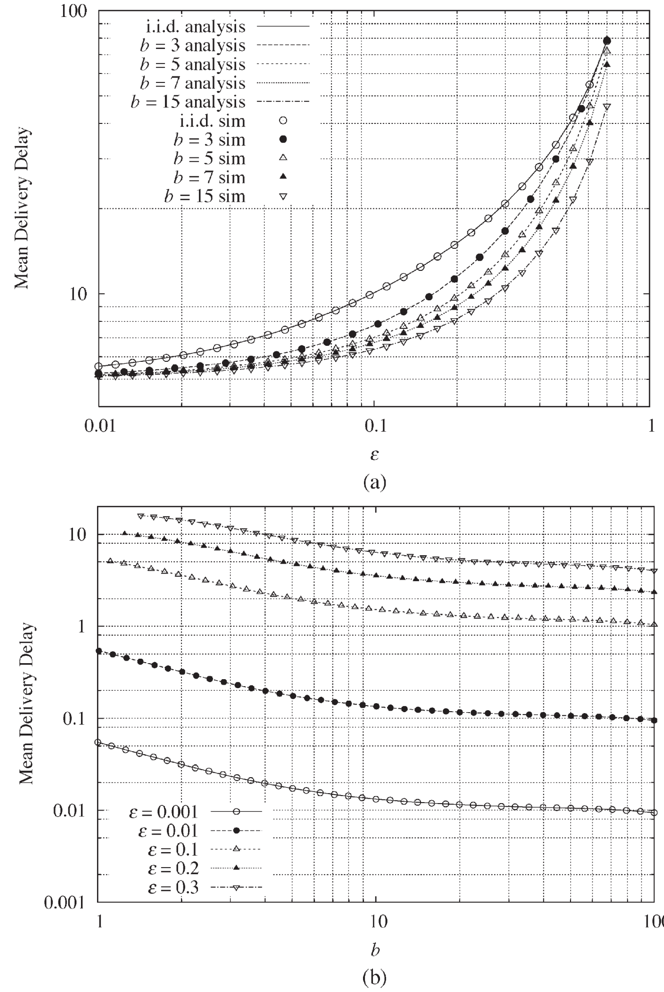


Fig. 4. Mean delivery delay. (a) Mean delivery delay as a function of ε. (b) Mean delivery delay as a function of b.

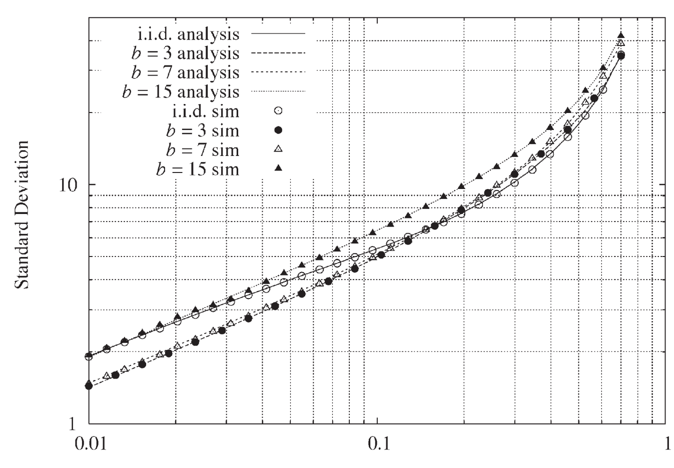


Fig. 5. Delivery delay standard deviation as a function of ε.

delivery delay is reported as a function of ε by varying b. In this graph, simulation points are also plotted to test the correctness of the analysis. In Fig. 4(b), the mean delivery delay is reported against the error burstiness b by varying ε. The first value of b on the leftmost part of the graph corresponds to the i.i.d. case, where $b = 1/(1 - \varepsilon)$. For each ε, the mean delivery delay is monotonically decreasing as a function of b. In other words, the i.i.d. case is the one characterized by the highest mean delivery delay under all channel conditions.

Fig. 5 reports the delivery delay standard deviation, simulation points are reported for comparison. Unlike for the mean delivery time, this metric in the i.i.d. case cannot be interpreted as a bound. In fact, its role with respect to the correlated case depends on both b and ε. Moreover, its behavior is clearly different from that of the other curves. However, this again shows that considering the channel as i.i.d. can be a misleading assumption when the channel is correlated.

In Fig. 6, we report the complementary distribution $\mathrm{ccdf}[k]$ by varying b and plotting the same graph in linear, Fig. 6(a), and logarithmic scales, Fig. 6(b). Again, from Fig. 6(a), it is clear that the i.i.d. case is not a suitable model when errors are correlated. In particular, we emphasize the higher probability of

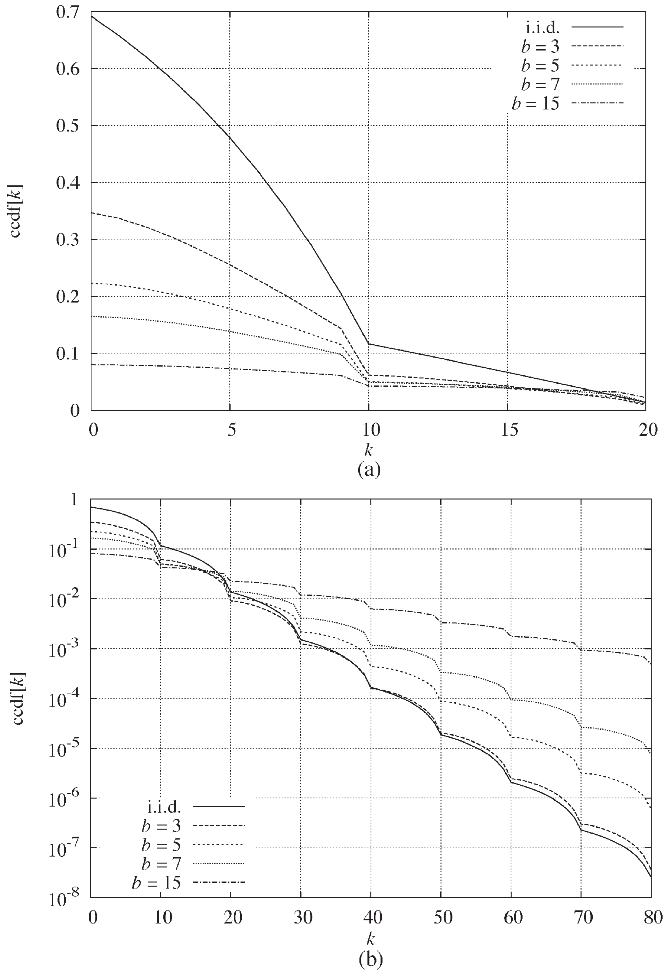


Fig. 6. Cumulative complementary delivery delay distribution, ccdf$[k]$ ($\varepsilon = 0.1$, $m = 10$). (a) Linear scale. (b) Logarithmic scale.

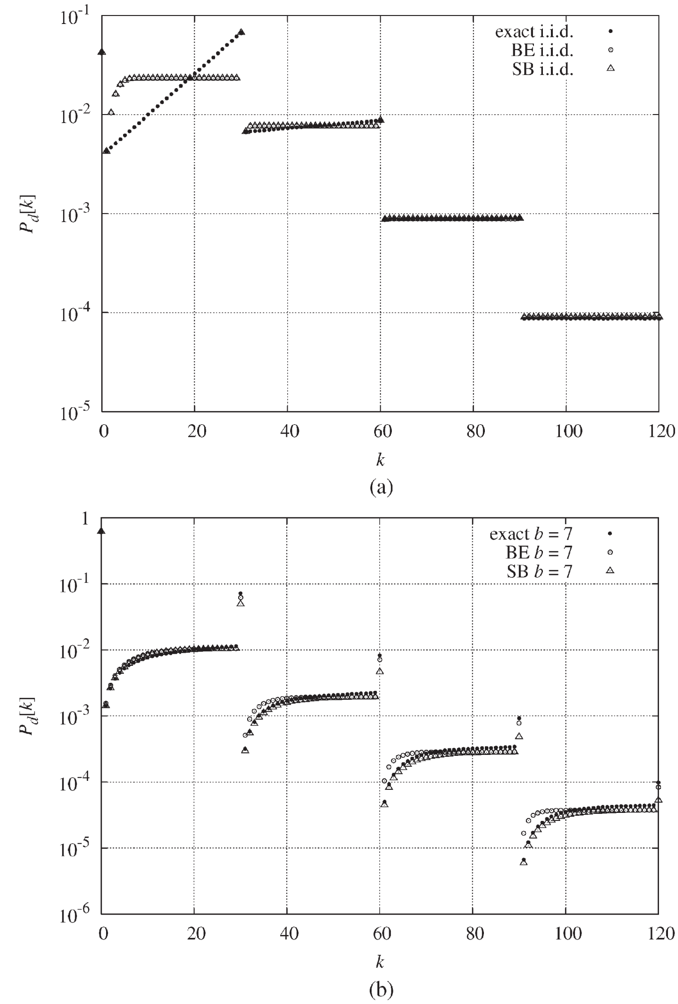


Fig. 7. Comparison between approximation and exact statistics, $m = 30$, $\varepsilon = 0.1$. (a) i.i.d. channel. (b) Correlated channel with $b = 7$.

a rapid delivery (i.e., for low k) in the correlated case, and also when the correlation is low, e.g., $b = 3$. Even after a full round ($k = m$), there is a gap in the curves: For instance, ccdf$[m]$ for $m = 10$ in the i.i.d. case is almost twice that in the correlated case with $b = 15$. Other differences for higher values of k can be observed in Fig. 6(b).

Next, we focus on the comparison between the exact analysis and the approximate approaches presented in Section IV. This comparison is reported in Fig. 7(a) and (b) for an i.i.d. channel and a correlated channel ($b = 7$), respectively. In these figures, the round-trip time has been considered to be large ($m = 30$). In general, the approximate approaches are in excellent agreement with the exact statistics. The only region where the approximations fail is for uncorrelated channel and at low delays.[4] In an i.i.d. channel, in fact, errors do not occur in bursts, and so the approximation made in Section IV that unresolved slots are disposed in a bursty way in this case does not hold. However, the effect of this approximation vanishes very quickly as the delay increases and the approximation becomes very close to the exact curve. In the i.i.d. case, both BE and SB approaches give the same results. When the channel is correlated, instead, the statistics obtained from BE and SB are in good agreement with the exact curve for any value of the delay, see Fig. 7(b). The BE approach overestimates the delivery delay statistics at the beginning of each round. The SB approach, instead, appears to underestimate the exact statistics for any value of the delay. Moreover, the estimate obtained from BE degrades as the delay k increases until, for a very large k, all points in the round are aligned over a straight line. SB, instead, gives a good approximation also for large values of k. The points derived using SB are the closest to the exact curve, except for $k = im$, $i \geq 1$, where the best estimate is given by BE. For what concerns the cumulative delay distribution (Fig. 8), BE gives the best estimate for any value of b. Moreover, in the independent case, the burst assumption only results in a small discrepancy regarding the first round. For any other value of k, exact statistics and approximation match almost perfectly. In

[4]Note, however, that in this particular case (i.i.d. errors), it is possible to develop a simpler exact analysis, so that an approximate approach becomes unnecessary.

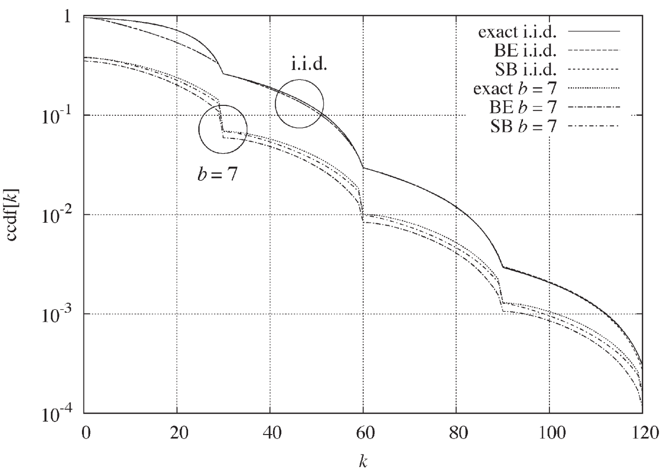


Fig. 8. Complementary cumulative delivery delay distribution, ccdf[k]: comparison between approximate and exact cases.

the correlated case ($b = 7$), the BE approach gives the best complementary cumulative delay distribution estimate. From the obtained results, we can conclude that the estimate of the delivery delay statistics is reasonably accurate for any b, so approximate methods could effectively be used in real systems enabling a fast and less memory expensive computation of the cumulative delay distribution also for large values of m.

VI. Conclusion

In this paper, we studied the delivery delay performance of an SR ARQ scheme over a two-state DTMC, considering a finite round-trip delay. We obtained the exact statistics of the delivery delay process regarding a single ARQ packet. The main characteristics of the delay statistics have been compared for several values of the channel error probability and error correlation. Simulation results confirm the goodness of the analysis. The main drawback of the exact analysis is that its complexity grows exponentially with the round-trip delay. To cope with this problem, an approximate approach has been presented. The statistics obtained using this approximate analysis is in good agreement with exact curves while keeping the complexity linear in the round-trip time value. In the final part of the paper, the distributions and their main characteristics are compared for several values of the channel error probability and error correlation. In particular, the results show that to neglect the channel correlation can be, though simpler, misleading. In general, this assumption leads to underestimating the system performance. Moreover, both our models, exact and approximate, allow to take into account this effect in a very accurate way by giving the possibility of integrating the ARQ effect in further analysis so as to characterize the higher layer protocol performance.

References

[1] ETSI SMG2, "Hybrid ARQ," Tech. doc. SMG2 UMTS-L23 165/98. Helsinki, Finland, Nov. 1998.

[2] A. Das, A. Ghose, V. Gupta, A. Razdan, H. Saran, and R. Shorey, "Adaptive link-level error recovery mechanisms in Bluetooth," in *Proc. IEEE Int. Conf. Personal Wireless Communications*, Hyderabad, India, Dec. 2000, pp. 85–89.

[3] Z. Jing and S. Roy, "An adaptive two-copy delayed SR ARQ for satellite channels with shadowing," in *Proc. IEEE Vehicular Technology Conf. Fall 2002*, vol. 2, pp. 849–853.

[4] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Commun.*, vol. 46, no. 11, pp. 1468–1477, Nov. 1998.

[5] R. Fantacci, "Queueing analysis of the selective repeat automatic repeat reQuest protocol for wireless packet networks," *IEEE Trans. Veh. Technol.*, vol. 45, no. 2, pp. 258–264, May 1996.

[6] J. Chang and T. Yang, "End-to-end delay of an adaptive selective repeat ARQ protocol," *IEEE Trans. Commun.*, vol. 42, no. 11, pp. 2926–2928, Nov. 1994.

[7] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968–1981, Sep. 2000.

[8] A. G. Konheim, "A queueing analysis of two ARQ protocols," *IEEE Trans. Commun.*, vol. COM-28, no. 7, pp. 1004–1014, Jul. 1980.

[9] M. E. Anagnostou and E. N. Protonotarios, "Performance analysis of the selective repeat ARQ protocol," *IEEE Trans. Commun.*, vol. COM-34, no. 2, pp. 127–135, Feb. 1986.

[10] Z. Rosberg and N. Shacham, "Resequencing delay and buffer occupancy under the selective repeat ARQ," *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 166–173, Jan. 1989.

[11] Z. Rosberg and M. Sidi, "Selective-repeat ARQ: The joint distribution of the transmitter and the receiver resequencing buffer occupancies," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1430–1438, Sep. 1990.

[12] D.-L. Lu and J.-F. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 721–730, May 1993.

[13] D. Towsley, "A statistical analysis of ARQ protocols operating in a nonindependent error environment," *IEEE Trans. Commun.*, vol. COM-29, no. 7, pp. 971–981, Jul. 1981.

[14] C. Leung, Y. Kikumoto, and S. A. Sorensen, "The throughput efficiency of the go-back-*N* ARQ under Markov and related error structures," *IEEE Trans. Commun.*, vol. 36, no. 2, pp. 231–234, Feb. 1988.

[15] M. Zorzi and R. R. Rao, "Throughput performance of ARQ selective-repeat protocol with time diversity in Markov channels with unreliable feedback," *Wireless Netw.*, vol. 2, pp. 63–75, Mar. 1996.

[16] R. Cam and C. Leung, "Throughput analysis of some ARQ protocols in the presence of feedback errors," *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 35–44, Jan. 1997.

[17] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-repeat-reQuest error control schemes," *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5–17, Dec. 1984.

[18] D. Bertsekas and R. Gallager, *Data Network*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[19] M. Yoshimoto, T. Takine, Y. Takahashi, and T. Hasegawa, "Waiting time and queue length distributions for go-back-*N* and selective-repeat ARQ protocols," *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1687–1693, Nov. 1993.

[20] M. Rossi, L. Badia, and M. Zorzi, "Exact statistics of ARQ packet delivery delay over Markov channel with finite round-trip delay," in *Proc. IEEE Globecom 2003*, San Francisco, CA, vol. 6, pp. 3356–3360.

[21] R. A. Howard, *Dynamic Probabilistic Systems*. New York: Wiley, 1971.

[22] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state Markov channel," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1537–1548, Oct. 1999.

Michele Rossi (S'02–M'04) was born in Ferrara, Italy, on October 30, 1974. He received the Laurea degree in electrical engineering (with honors) and the Ph.D. degree in information engineering from the University of Ferrara, Ferrara, Italy, in 2000 and 2004, respectively.

Since 2000, he has been a Research Fellow at the Department of Engineering, University of Ferrara. During 2003, he was on leave at the Center for Wireless Communications (CWC) at the University of California San Diego (UCSD), where he did research on wireless sensor networks. His research interests are on: TCP/IP protocols on wireless networks, TCP/IP header compression, performance analysis of selective repeat link layer retransmission techniques, efficient multicast data delivery in 3G cellular networks, and MAC/routing in wireless sensor networks.

Leonardo Badia (S'02–M'04) was born in Ferrara, Italy, in 1977. He received the Laurea degree (with honors) in electrical engineering and the Ph.D. degree in information engineering from the University of Ferrara, Ferrara, Italy, in 2000 and 2004, respectively.

Since 2001, he has been with the Department of Engineering of the University of Ferrara. During 2002 and 2003, he was on leave at the Radio System Technology Labs (now Wireless@KTH), Royal Institute of Technology of Stockholm, Sweden. His research interests include energy-efficient ad hoc networks, transmission protocol modeling, admission control, and economic modeling of radio resource management for wireless networks.

Michele Zorzi (S'89–M'95–SM'98) was born in Venice, Italy, in 1966. He received the Laurea degree and the Ph.D. degree in electrical engineering from the University of Padova, Padova, Italy, in 1990 and 1994, respectively.

During the Academic Year 1992–1993, he was on leave at the University of California, San Diego (UCSD), attending graduate courses and doing research on multiple access in mobile radio networks. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, he joined the School of Engineering of the University of Ferrara, Italy, in 1998, and the Department of Information Engineering of the University of Padova, Italy, in 2003, where he is currently a Professor. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, and energy constrained communications protocols.

Dr. Zorzi is the Editor-In-Chief of the *IEEE Wireless Communications Magazine*, and currently serves on the Editorial Boards of the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the Wiley Journal of Wireless Communications and Mobile Computing, and the ACM/URSI/Kluwer Journal of Wireless Networks. He was also Guest Editor for special issues in the *IEEE Personal Communications Magazine* (Energy Management in Personal Communications Systems) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Multimedia Network Radios).