SR ARQ Packet Delay Statistics on Markov Channels in the Presence of Variable Arrival Rate

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

***Abstract*— In this letter we investigate the packet delay statistics of a fully reliable Selective Repeat ARQ scheme by considering a Discrete Time Markov Channel with non-instantaneous feedback and assigned round-trip delay m. Our focus is on studying the impact of the arrival process on the delay experienced by a packet. An exact model is introduced to represent the system constituted by the transmitter buffer, the m round-trip slots, and the channel state. By means of this model, we evaluate and discuss the delay statistics and we analyze the impact of the system parameters, in particular the packet arrival rate, on the delay statistics.**

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

I. Introduction

AUTOMATIC Retransmission reQuest (ARQ) is a widely used error control technique for data communication, besides Forward Error Correction. The three basic ARQ techniques are Stop-and-Wait, Go-Back-N and Selective Repeat (SR). In SR ARQ, the sender retransmits only the negatively acknowledged packets and then resumes the transmission process from the last packet sent so far. In such a scenario, the delays experienced by different packets are related, since the packets must be released in-order, i.e., the actual delivery of a packet only occurs after the correct reception of all packets with lower identifier.

Several terms [1] contribute to the global delay experienced by a packet, called τ_G in the following. For our analysis, τ_G is subdivided in two parts. The former, called *queueing delay* and denoted with τ_Q, is the time spent in the source buffer before the first transmission, and might be related to the distribution of transmitter buffer occupancy [2]. The latter is the *delivery delay* τ_D, which is between the first transmission and the release of a packet from the re-sequencing buffer. This is the sum of the time for correct reception and the acknowledgment time for previous pending packets, called *re-sequencing delay*, which depends on the correct reception of other packets. About τ_D, note that between every transmission and the corresponding packet reception there is a time gap equal to the constant propagation delay t_c. To simplify the notation, this constant term will be omitted. This means that in the following the delivery statistics will be considered at the transmitter's side: the delivery delay at the receiver's side is simply $\tau_D + t_c$ [3].

Many contributions deal with SR ARQ statistics. However, some approximations are often introduced to make the problem more tractable. A simplifying assumption used in the literature [2] is to consider an independent (*iid*) error process on the channel. This makes the analysis easier, even though the impact of the channel error burstiness is neglected, which is undesirable as it strongly affects the results of ARQ delay [3].

Another simplification is to consider an instantaneous feedback. This situation is known in the literature [4] as *ideal* SR ARQ. In this case the information about the correct reception of a packet is immediately available after its transmission, hence the system is simpler and the analysis can neglect the possibility of having pending packets at the receiver's buffer. However, this also means that the re-sequencing delay is zero, whereas in real systems it is a large part of the delivery delay. Thus, the ideal SR ARQ case does not allow a realistic evaluation of this term.

Finally, another common approach [1] is to assume that the sender always has a packet to transmit. This so-called *Heavy Traffic assumption* is a realistic model for continuous traffic sources, but might fail to represent more general cases. In particular, this assumption prevents the queueing delay statistics from being evaluated, as the buffer occupancy is arbitrarily high, whereas it is useful for the delivery delay, as shown in the sequel.

Our contribution is to relax these simplifications, by deriving a general exact approach. Differently from other contributions appeared so far, our approach allows to obtain statistical moments of every order, instead of plain average values. In particular, in [3] we already developed an exact analysis based on the Heavy Traffic assumption and focusing on the delivery delay only. In a further contribution [5] we extended this study to a more general N-state Markov channel. Here instead we fill the gap by studying all the delay terms (thus, in particular, the queueing delay), and considering a more general arrival process. This generalization is achieved as in [6] by considering a Bernoulli model, which can be tuned by varying the arrival rate λ.

The SR ARQ delays have been topic of other contributions, in particular in [1] the authors consider a time varying channel and a finite round-trip delay, but the derived model is approximate in some components and only average values are evaluated. In [2], the distribution of buffer occupancies is derived for a general arrival process, but in the case of iid errors only. Moreover, a window-based approach is considered, which prevents the packet from being transmitted immediately after its arrival, as the transmitter must wait until

Manuscript received September 10, 2004; revised February 2, 2005; accepted February 10, 2005. The associate editor coordinating the review of this letter and approving it for publication was E. Hossain.

L. Badia and M. Rossi are with the Dept. of Engineering, University of Ferrara, via Saragat 1, 44100 Ferrara, Italy (email: {lbadia, mrossi}@ing.unife.it).

M. Zorzi is with the Dept. of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy (e-mail: zorzi@ing.unife.it).

Digital Object Identifier 10.1109/TWC.2006.04628.

the end of the window. Also [6] considers a Bernoulli arrival process, but again with iid error process. In [7], the end-to-end delay in case of Adaptive SR ARQ and general arrival process is studied, but the analysis is approximated. Finally, a very recent contribution on the matter, which also investigates the queueing delay, can be found in [8].

II. Model for SR ARQ Queueing and Transmission Processes

The system under analysis consists of a pair transmitter/receiver. The former sends data packets to the latter through a slotted noisy channel, where the time for a packet transmission corresponds to one slot. The receiver answers with ACK/NACK packets according to the correct/erroneous reception of the data packets, respectively. After a full round-trip time, feedback packets arrive at the transmitter's side. As long as ACKs are received, the sender transmits packets in increasing numerical order. When a NACK is received instead, a retransmission is scheduled (which therefore occurs after a full round-trip time from the previous transmission attempt). In other words, retransmissions have priority over packets queued at the transmitter. The packets arrived in the queue are immediately available for the transmission. This means that if the queue is empty and no retransmission is scheduled, a packet is transmitted in the same slot it arrives.

The data packets are released *in-order* to higher layers, i.e., release is possible only once all packets with lower identifier have also been acknowledged. In SR ARQ the receiver keeps in a buffer the packets correctly received but not yet released, so that the sender retransmits unacknowledged packets only.

The following work assumptions are introduced: i) The Link Layer protocol is fully reliable, i.e., every packet is transmitted (or retransmitted) until correct reception. ii) Both receiver and transmitter buffers have unlimited size. iii) ACK/NACK packets are error-free. For what concerns these assumptions, note that i) and ii) are standard hypotheses to make the problem analytically tractable. Also, for what concerns the transmitter buffer size, note that an upper-limit can be introduced in what follows in a straightforward manner. Assumption iii) instead can be easily removed if necessary by following the approach presented in [9], where an extended analysis accounts for erroneous feedback. We do not introduce such an extension here, as it will only bring to tedious complication in the calculus, without substantially changing the analytical approach.

We consider a Bernoulli model for the arrival process, i.e., a packet arrival may occur in every slot with constant probability λ. However, the outlined framework is very general, so that this assumptions can be replaced by more complicated arrival processes if required, basically with the same approach but with more cumbersome computations. In this view, our contribution can be easily extended to take into account correlations in the arrival process, e.g., by considering a Markov source as in [1]. The choice of the Bernoulli arrival process is however sufficient to gain deep knowledge. For example, such an arrival process is able to describe different load conditions, by varying λ. In particular, the Heavy Traffic assumption corresponds to λ equal to 1, even though the steady state condition when λ overcomes $1-\varepsilon$, where ε is the steady-state channel error probability, also approaches the Heavy Traffic case, since the buffer is never empty.

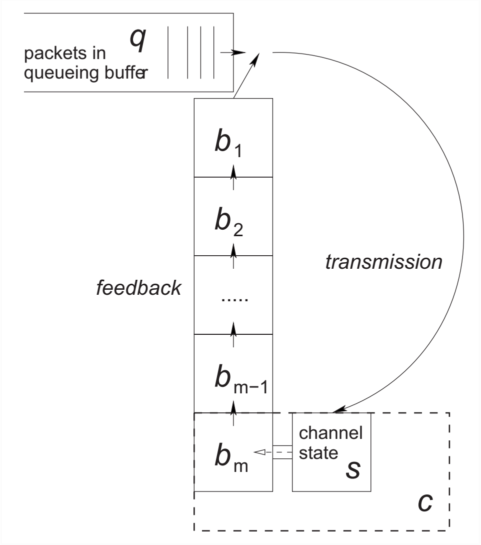

Fig. 1. Snapshot of the state of the SR ARQ transmission system.

For what concerns the channel, we represent it with a Discrete Time Markov Chain (DTMC). The transitions of this DTMC are in correspondence with the transmission slots. For the sake of simplicity, in the following we assume to have a 2-State Markov Channel, where state 0 is error-free, and 1 is always erroneous. This DTMC is fully characterized by the transition matrix $\mathbf{P} = \{p_{ij}\}$, $i, j \in \{0, 1\}$. For this model, the steady-state channel error probability is $\varepsilon = p_{01}/(p_{10} + p_{01})$ and the average error burst length is $B = 1/p_{10}$. In spite of its simplicity, the assumption of having a 2-State Markov Channel is not restrictive for what follows. In fact, a more complicated approach (which again leads only to more cumbersome formulae without significant differences in the procedure) can be derived for a more general N-state Markov Channel, as outlined in [9]. Thus, it is possible to extend our analysis to more general cases in a straightforward manner. In particular, this includes as a particular case the well-known Gilbert–Elliot model, where the good and bad states are characterized by error probabilities P_{good} and P_{bad}. This can be translated into a 4-state Markov chain where all states are either always error-free or always erroneous, as in our approach.

III. Markov Model of Queueing Buffer and Feedback Channel

The delivery process evolves as in Fig. 1. At each instant, a packet is transmitted on the channel, and it can be either a retransmission or a new packet taken from the queueing buffer. Retransmissions occur after a full round-trip time, assumed equal to m slots. This is not restrictive, since if the round-trip time does not equal an integer number of slots, we can repeat the reasoning by considering m as the smallest number of slots which exceeds the round-trip time.

Thus, an m-sized retransmission window can be used to track the status of the last m transmitted packets. This can be done by considering an m-sized vector $\mathbf{b}$, with elements $b_i \in \{0, 1\}$, $1 \leq i \leq m$. The mth bit indicates the slot currently under transmission at time t, where the bits b_j, $1 \leq j \leq m{-}1$ refer to the transmission at time $t{-}m{+}j$. For all bits, a value equal to 0 indicates that at that time no retransmission was scheduled, whereas 1 means that a transmission failed. We also need to track the number q of packets in the queue at the transmitter buffer and the channel state s, which might be either 0 or 1, i.e., good or bad. Due to the Markovian nature of the channel, it is sufficient to keep track only of the value of s at time t.

Hence, the full state of the delivery process can be described through the triple $(q(t), \mathbf{b}(t), s(t))$. However, a simplification is possible. In fact, the binary variables b_m and s are not independent, as a retransmission in the current slot is scheduled only if the channel is bad, thus it is impossible that $s = 0$ and $b_m = 1$. The converse does not hold, since b_m can be 0 even if $s = 1$, and this happens if no packet is transmitted. For this reason, we replace b_m and s with a ternary variable c, since only three situations are possible, which are: the channel state is good, which implies that there is anyway no need for retransmission (we denote this with $c = 0$), the channel state is bad *and* a packet is transmitted, which indicates a retransmission scheduling ($c = 1$), the channel is bad but no packet is transmitted, thus no retransmission is scheduled anyway (in this case, c is let equal to –1). It is necessary to distinguish $c = -1$ from $c = 0$, since both represent no retransmission but for different channel conditions.

Now, $X(t) = (q, b_1, b_2, \ldots, b_{m-1}, c)$ is a Markov chain[1] . In fact, observe that the knowledge of $X(t)$ is sufficient to determine the value of $X(t + 1)$ by considering every possibility of channel transition and packet arrival (on the aggregate, 2×2 cases). In the following, we discuss the evolution of this Markov chain by explicitly deriving its transition matrix.

First of all, note that due to the cyclic behavior of the ARQ window, it is easy to realize that the values of $b_1, \ldots, b_{m-1}$ at time $t{+}1$ evolve deterministically, depending on $\mathbf{b}$ and c, as follows: $b_j(t{+}1) = b_{j+1}(t)$ for $1 \leq j \leq m-2$, and $b_{m-1} = u[c-1]$, where $u[\cdot]$ is the unit-step (i.e., $u[n] = 1$ if $n \geq 0$, and 0 otherwise). Instead, $q(t + 1)$ and $c(t + 1)$ depend on the values of $q(t)$, $c(t)$ and also $b_1(t)$, and can have different values according to the packet arrival and channel variation process. In particular, $c(t{+}1)$ always evolves following the channel transition but for the case in which no packet is transmitted, where for bad channel it is -1 instead of $+1$. This condition occurs if $q(t){+}b_1(t){=}0$ and no packet is generated. Otherwise, if either there is a retransmission, or a packet already in the queue is transmitted or finally a newly arrived packet is transmitted, the bad channel condition always implies $c = 1$.

Henceforth, the transition matrix $\mathbf{T}(\mathbf{P}, \lambda)$ of the Markov chain $X(t)$, which is a function of the matrix $\mathbf{P}$ and the arrival rate λ, has $3 \cdot 2^{m-1}$ rows for each possible value of $q(t)$ and every row has only 4 non-zero elements. In particular, consider a generic transition starting from state $X(t) = (q, b_1, b_2, b_3, \ldots, b_{m-1}, c)$, where all internal components are evaluated at time t. If $s(t{+}1) = d$, which can take value in $\{0, 1\}$, the destination states can be $X(t{+}1) = (q + b_1, b_2, b_3, \ldots, b_{m-1}, u[c{-}1], d)$, with probabilities $\lambda p_{|c|d}$, and $X(t{+}1) = (q{+}b_1{-}u[q{+}b_1{-}1], b_2, b_3, \ldots, b_{m-1}, u[c{-}1], d \cdot (2u[q{+}b_1{-}1] - 1))$, with probabilities $(1{-}\lambda)p_{|c|d}$.

The following set of balance equations can be written [2] :

$$\pi(q, b_1, b_2, \ldots, b_{m-2}, \beta, c) = \quad (\text{for } q > 0, c \in \{0, 1\})$$
$$= \sum_{x=2\beta-1}^{\beta} \sum_{\alpha=0}^{1} \Big(\lambda p_{|x|c} \pi(q{-}\alpha, \alpha, b_1, b_2, \ldots, b_{m-2}, x) +$$
$$+(1{-}\lambda) p_{|x|c} \pi(q{-}\alpha{+}1, \alpha, b_1, \ldots, b_{m-2}, x) \Big) \tag{1}$$

$$\pi(q, b_1, \ldots, b_{m-2}, \beta, -1) = 0 \quad (\text{for } q > 0) \tag{2}$$

$$\pi(0, b_1, \ldots, b_{m-2}, \beta, 0) =$$
$$= \sum_{x=2\beta-1}^{\beta} \Big(\lambda p_{|x|0} \pi(0, 0, b_1, b_2, \cdots, b_{m-2}, x) +$$
$$+ \sum_{\alpha=-1}^{1} (1{-}\lambda) p_{|x|0} \pi\big((1{-}\alpha)u[\alpha], \alpha u[\alpha], b_1, \ldots, b_{m-2}, x\big)\Big) \tag{3}$$

$$\pi(0, b_1, b_2, \ldots, b_{m-2}, \beta, 1) =$$
$$= \sum_{x=2\beta-1}^{\beta} \Big(\lambda p_{|x|1} \pi(0, 0, b_1, b_2, \ldots, b_{m-2}, x) +$$
$$+ \sum_{\alpha=0}^{1} (1 - \lambda) p_{|x|1} \pi(1 - \alpha, \alpha, b_1, b_2, \ldots, b_{m-2}, x) \Big) \tag{4}$$

$$\pi(0, b_1, b_2, \ldots, b_{m-2}, \beta, -1) =$$
$$= \sum_{x=2\beta-1}^{\beta} \big((1 - \lambda) p_{|x|1} \pi(0, 0, b_1, b_2, \ldots, b_{m-2}, x) \big) \tag{5}$$

This set of equations cover all possible states. In particular, Eq. (1) includes two main cases, according to whether or not there is an arrival during the previous slot, which give the two terms multiplied by λ and $(1 - \lambda)$, respectively. Thus, a buffer occupancy q larger than 0 is derived in the former case from a previous buffer occupancy $q - \alpha$ and a first bit of the bitmap $\mathbf{b}$ equal to α, where α can be 0 or 1. This means that either the number of packets in the buffer was $q{-}1$ but a retransmission occurs (left-most bit equal to $\alpha = 1$), so that the number of packets in the buffer increases, or it was q and the new arrival is compensated by the transmission of a packet from the buffer. In the latter (no packet arrival) we can repeat the above reasoning but we must account for a buffer occupancy in the previous slot with one more packet. Eq. (2) follows immediately from the observation that it is impossible to have $c = -1$ when the buffer occupancy is larger than 0; in fact, $c = -1$ describes a bad channel condition where no packet is transmitted since the buffer is empty, no retransmission is scheduled and no packets arrived in the previous slot. Eq. (3) completes the case of good channel by using the same approach as in Eq. (1). However, here the inner sum comprises only one case in the first term, i.e., when a packet has arrived, as a buffer occupancy equal to 0 can be achieved only if the buffer was already empty and no

[1] All components are here evaluated at time t. To avoid long expressions, the time indication will be omitted when obvious.

[2] In the following, the script b_{m-1}, which occurs often, has been replaced by β, only to simplify the notation.

retransmission is scheduled. In the second term instead three possibilities are included, since we now have to account also for the case where the buffer was empty and no transmission was scheduled, which is the term of the sum corresponding to $\alpha = -1$, whereas $\alpha = 0, 1$ gives the terms already included in the sum, as in Eq. (1). Finally, Eqs. (4)–(5) describe any remaining possibility of channel transition to the erroneous state. Remember again that the case where the buffer is empty and no retransmission is scheduled evolves with $c = -1$, otherwise $c = 1$. Thus, the latter case is considered in Eq. (4), where Eq. (5) accounts for the special case where $c = -1$.

If we impose the sum of all π's to be 1, the above set of equations can be analytically solved for any value of $0 \leq \lambda < 1 - \epsilon$ by observing that the matrix $\mathbf{T}(\mathbf{P}, \lambda)$ is partitioned in the form:

$$\begin{pmatrix} \mathbf{S_0} & \mathbf{L_0} & \mathbf{0} & \cdots & & & \\ \mathbf{M_1} & \mathbf{S_1} & \mathbf{L_1} & \mathbf{0} & \cdots & & \\ \mathbf{0} & \mathbf{M_1} & \mathbf{S_1} & \mathbf{L_1} & \mathbf{0} & \cdots & \\ \vdots & \mathbf{0} & \mathbf{M_1} & \mathbf{S_1} & \mathbf{L_1} & \mathbf{0} & \cdots \\ & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

where the block of size $3 \cdot 2^{m-1}$ in position (q, q') includes the transitions from buffer occupancy q to buffer occupancy q'.

The transition matrix above is very similar to the ones characterizing Quasi Birth and Death (QBD) processes [10], even though it is not a true QBD process since the sub-matrix $\mathbf{L_0}$ is not equal to the sub-matrices $\mathbf{L_1}$. In fact, the topmost row relates to Eqs. (3)–(5), whereas the rows describing the transitions from every $q > 0$ can be inferred from Eqs. (1)–(2). This difference does not prevent a recursive solution of the chain by following an approach akin to the one presented in [10] to solve generalized birth-and-death processes where the arrival and departure rates depend on the system state. The modifications necessary to solve our problem concern the fact that the system state is not fully described by the channel evolution only, since in our whole Markov chain also the buffer state impacts on the feedback vector (in particular on c). However, this changes only the first part of the recursion, i.e., when the $\pi(1,\cdot,\cdot)$'s are expressed in terms of the $\pi(0,\cdot,\cdot)$'s. From this point on, the derivation of the $\pi(q+1,\cdot,\cdot)$'s in terms of the $\pi(q,\cdot,\cdot)$'s is always the same. By following again [10], we can prove that this approach admits a solution when $\lambda < 1 - \varepsilon$. In fact, the recursive approach is convergent if the generalized departure rate, which is either 0 or 1 according to $1 - b_1$, is on average higher than the arrival rate, which is always equal to λ. The computational complexity of such an approach corresponds to the solution of a linear system with $6 \cdot 2^{m-1}$ equations, which accounts for the blocks $q = 0$ and $q = 1$, then the recursion for higher q's is obtained at the price of subsequent multiplication by a $3 \cdot 2^{m-1}$ matrix. Thus, the complexity of the solution strongly depends on m.

IV. Queueing and Delivery Delay Evaluation

The Markov chain described in the previous section allows us to determine the delay statistics in an exact way. The evaluation of the full statistics of both queueing and delivery delay for the general case of packet arrival rate λ are original contributions presented here.

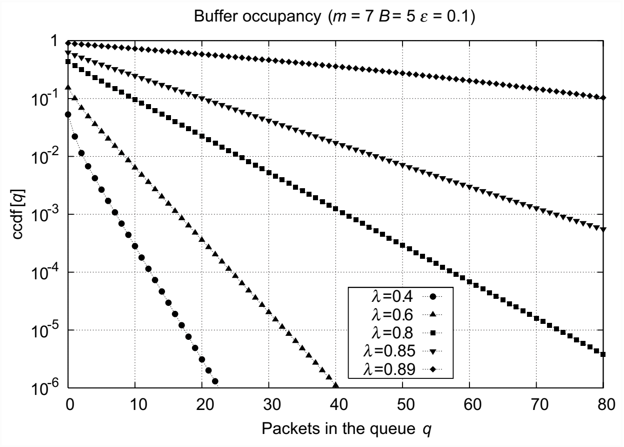

Fig. 2. Complementary cumulative distribution (ccdf) of the number q of packets in the queueing buffer for $\varepsilon = 0.1$, $B = 5$, $m = 7$.

Define $\mathbf{a}$ as $(b_1, b_2, \ldots, b_{m-1})$, i.e., a truncated $\mathbf{b}$, without b_m. Let $\pi(q, \mathbf{a}, c)$ be the stationary probability of a generic state $X(t) = (q, \mathbf{a}, c)$. The probability of having queueing buffer occupancy equal to q is: $P[q] = \sum_{\mathbf{a} \in \mathcal{A}} \sum_{c=-1}^{1} \pi(q, \mathbf{a}, c)$, where $\mathcal{A} = \{0,1\}^{m-1} = \{\mathbf{w} = (w_1, w_2, \ldots, w_{m-1}) \mid w_i \in \{0,1\} \ \forall i = 1, 2, \ldots, m-1\}$. In Fig. 2 we report the complementary cumulative distribution function of the queueing buffer occupancy, i.e., the probability that more than q packets are queued in the transmitter's buffer, evaluated in an exact way with the above model for different values of λ when the average error probability, the average burst length and the round-trip delay are $\varepsilon = 0.1$, $B = 5$ and $m = 7$, respectively. From Fig. 2 it can be observed that λ has a heavy impact on the buffer occupancy, in particular when $\lambda \approx 1 - \varepsilon$ the occupancy tends to become arbitrarily high.

To evaluate the packet delay, consider the arrival of a given packet in the queueing buffer. The conditional probability $\Lambda(q, b_1, b_2, \ldots, b_{m-2}, \beta, c)$ that the system state is $(q, \mathbf{a}, c)$ given that in the previous slot a packet had arrived can be evaluated as follows:

$$\Lambda(q, b_1, b_2, \ldots, b_{m-2}, \beta, c) = \tag{6}$$
$$= \begin{cases} \displaystyle\sum_{x=2\beta-1}^{\beta} \sum_{\alpha=0}^{u[q-1]} p_{|x|c} \pi(q-\alpha, \alpha, b_1, \ldots, b_{m-2}, x) & \text{if } c \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Eq. (6) is easily derived from Eqs. (1)-(5) by considering only the transitions with a packet arrival.

If the column vector $\mathbf{v}_m$ is defined as a vector of m elements all equal to 1, the newly arrived packet has $q - u[q-1] + \mathbf{b}\mathbf{v}_m$ packets ahead in the transmission order, which are still not correctly received[3]. Now, it is possible to consider the Markov chain defined by the transition matrix $\mathbf{T}(\mathbf{P}, 0)$, in which the arrival process is "turned off." In fact, as shown in [3], future arrivals do not affect the queueing delay, nor the

[3] Observe that $\mathbf{b}\mathbf{v}_m$ is the number of elements of $\mathbf{b}$ equal to 1. If $q = 0$ after an arrival, this means that the packet has been transmitted immediately.

delivery delay, of the packet of interest. The Markov chain with $\lambda = 0$ evolves again by following the procedure outlined in Section III.

We present two equivalent ways to solve this Markov chain. The first method exploits the fact that, intuitively speaking, any packet eventually exits the queue and arrives at correct delivery with probability 1. Formally, $\mathcal{Q} = \{(0, \mathbf{a}, c) : \mathbf{a} \in \mathcal{A}, -1 \le c \le 1\}$ is an absorbing set for the Markov chain and so is $\mathcal{G} = \{(0, \mathbf{0}, 0), (0, \mathbf{0}, -1)\}$, where $\mathbf{0}$ is an $(m-1)$-sized zero vector. The proof of this statement follows immediately, for if $\lambda = 0$, then $\lim_{t\to\infty} q$ and $\lim_{t\to\infty} \mathbf{bv}_m$ are both zero. When the Markov chain enters the set $\mathcal{Q}$ the packet of interest is released from the queueing buffer, where the set $\mathcal{G}$ corresponds to the conditions where the packet and also all previously transmitted packets are acknowledged. Thus, if $f_{(q,\mathbf{a},c)\,\mathcal{Q}}(t)$ and $f_{(q,\mathbf{a},c)\,\mathcal{G}}(t)$ are the probabilities that the first passage times [11] from the state $(q, \mathbf{a}, c)$ to the absorbing sets $\mathcal{Q}$ and $\mathcal{G}$, respectively, equal t slots, the statistics of the queueing delay τ_Q can be evaluated as:

$$\text{Prob}\{\tau_Q = t\} = \sum_{q=0}^{+\infty} \sum_{\mathbf{a}\in\mathcal{A}} \sum_{c=-1}^{+1} \Lambda(q, \mathbf{a}, c) f_{(q,\mathbf{a},c)\,\mathcal{Q}}(t)\ . \quad (7)$$

An alternative view of the problem can be given by considering a column vector $\mathbf{e}_Q$ of all ones in the entries with $q = 0$ and all zeros in the entries with $q > 0$, i.e., the vector of indicator functions of the set $\mathcal{Q}$. In this case:

$$\mathcal{C}_\mathcal{Q}[t] = \mathbf{\Lambda} \cdot [\mathbf{T}(\mathbf{P}, 0)]^t \cdot \mathbf{e}_\mathcal{Q}, \quad t \ge 0\,, \quad (8)$$

where $\mathbf{\Lambda}$ denotes the vector collecting all $\Lambda(q, \mathbf{a}, c)$'s. The distribution $\mathcal{C}_\mathcal{Q}[t]$ is the probability that the queueing delay is lower than or equal to t. Thus, the probability $\text{Prob}\{\tau_Q = t\}$ is:

$$\text{Prob}\{\tau_Q = t\} = \begin{cases} \mathcal{C}_\mathcal{Q}[0] & \text{if } t = 0 \\ \mathcal{C}_\mathcal{Q}[t] - \mathcal{C}_\mathcal{Q}[t-1] & \text{if } t > 0 \end{cases} \quad (9)$$

In both cases apparently q goes to infinity, which requires either an infinite sum in Eq. (7) or an infinite matrix in Eq. (8). However, the observation that $f_{(q,\mathbf{a},c)\,\mathcal{Q}}(t) = 0$ if $q > t$, i.e., a buffer with q packets can not be emptied in less than q timeslots, means that the evaluations above only involve a finite number of terms, i.e., the terms where $q > t$ are all zero. The statistics of the overall delay τ_G can be evaluated by following the same approach. Only, to obtain $\text{Prob}\{\tau_G = t\}$ it is necessary to replace $f_{(q,\mathbf{a},c)\,\mathcal{Q}}(t)$ with $f_{(q,\mathbf{a},c)\,\mathcal{G}}(t)$, or equivalently $\mathbf{e}_Q$ with a vector $\mathbf{e}_G$ which has ones only in positions $(0, \mathbf{0}, 0)$ and $(0, \mathbf{0}, -1)$, and zeros elsewhere. Finally, the delivery delay τ_D is $\tau_G - \tau_Q$. Thus, $\text{Prob}\{\tau_D = t\}$ can be obtained as deconvolution of $\text{Prob}\{\tau_G = t\}$ and $\text{Prob}\{\tau_Q = t\}$.

In Fig. 3 we report the complementary cumulative distributions of the queueing delay. We consider the same parameters as in Fig. 2. Fig. 3 shows how the trend of the buffer occupancy is reflected on the queueing delay (even though the queueing delay distribution has a heavier tail). Simulation results are plotted to show the accuracy of the analysis. Note that the analytical results are even more accurate, since for example for low probability values or for high values of the delay the simulation results require a very long time to converge.

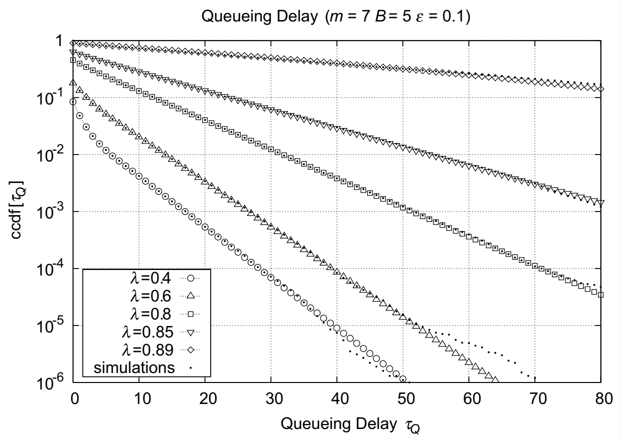

Fig. 3. Complementary cumulative distribution (ccdf) of the queueing delay for $\varepsilon = 0.1$, $B = 5$, $m = 7$.

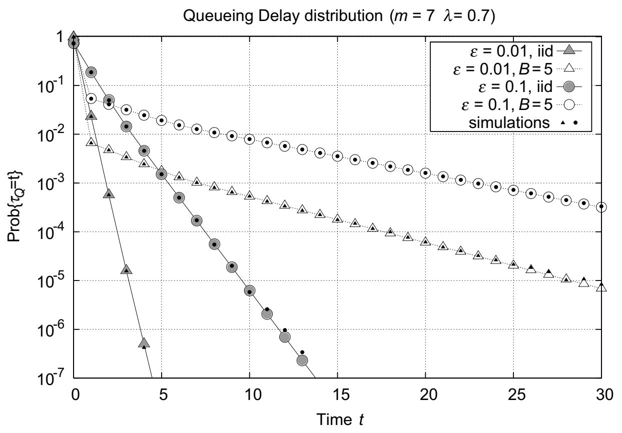

Fig. 4. Statistics of the queueing delay for $m = 7$ and $\lambda = 0.7$ for different values of the channel burstiness B and error probability ε.

The effect of the distribution of the error bursts of the Markov channel on the queueing delay is presented in Fig. 4, where simulation results are again shown for comparison. It is emphasized that the performance reported in Fig. 3 is similar to other cases with different values of ε, where the curves are simply translated without changing their behavior. However, the impact of the channel burstiness deserves more emphasis. The comparison made in Fig. 4 of the performance of bursty ($B = 5$) and iid channel shows in fact that, even though the trend is similar, an iid channel is not a good model for wireless channels, which are often characterized by bursty errors. In particular, the delay distribution of the bursty channel has a heavier tail than in the iid case. This means that bursts of errors can bring the delay to higher values [1]; in fact, once the bad channel state is entered the system stays in that state for a longer period, thereby postponing the resolution of the corrupted packets.

In Fig. 5 we plot complementary cumulative distribution of the delivery delay. Here, the case with $\lambda = 0.7$, $\varepsilon = 0.1$, $B = 5$ is plotted as a reference, so that this curve can be compared with other results obtained by changing only one of the parameters. However, we tested the comparison also for different combinations, so that the conclusions are quite general. The main insight gained from Fig. 5 is that the delivery delay is almost insensitive to the packet arrival

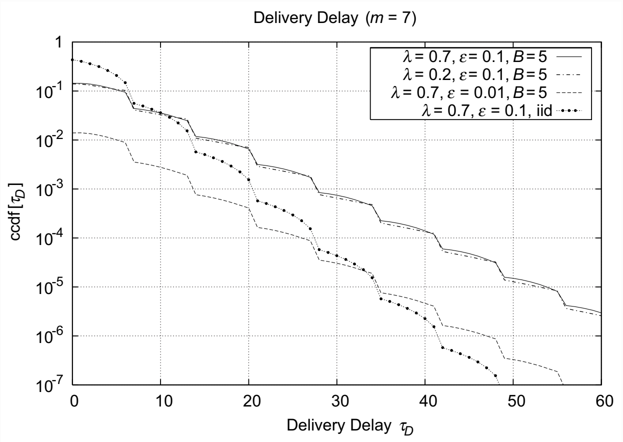

Fig. 5. Complementary cumulative distribution (ccdf) of the delivery delay for $m = 7$ and various values of the other parameters.

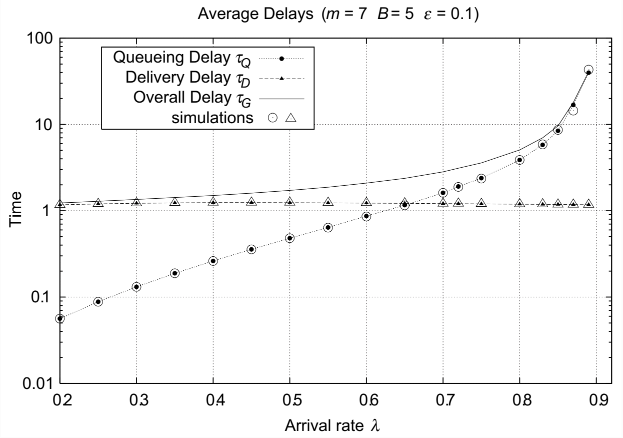

Fig. 6. Average values of the queueing delay, delivery delay and overall delay for $m = 7$, $B = 5$, $\varepsilon = 0.1$ as a function of λ.

process (λ). In fact, even with a different λ the curves remain very close. Note that for $\lambda \geq 0.3$ the curves overlap almost perfectly, so they are not even plotted as they would be indistinguishable from the case $\lambda = 0.7$. As a consequence, the study of the delivery delay under the Heavy Traffic condition is reasonable, unless λ is very small. This justifies the studies presented in [1], [3], where the delivery delay has been analyzed under Heavy Traffic. The effect of the average channel error probability is simply quantitative and not qualitative, as a change in ε has simply the effect of translating the curves. Instead, the comparison between different values of the channel burstiness offers again interesting conclusions. In particular, for an iid channel the results are also qualitatively different. We observe the same trend of the bursty channel, which brings the tail of the distribution to higher values. This phenomenon will increase as B is increased. However, the main gap is between the iid case and the bursty channel. This again confirms that neglecting the channel correlation leads to poor approximations in evaluating the transmission performance.

Finally, the dependence of all these results on the arrival rate is summarized in Fig. 6, where the average delays (queueing, delivery and overall[4]) are plotted versus λ. Here, the same system parameters as in Fig. 2 have been considered, and also simulation results are plotted for both delays (queueing and delivery), which demonstrate that the analysis is accurate[5]. In Fig. 6 it is shown again that the average τ_Q and hence the average τ_G heavily increase as λ increases, whereas the average τ_D remains constant. Such a figure can also be useful to recognize the contribution to the total overall delay of the two terms. In fact, the queueing and the delivery delays have comparable weights when λ is between 0.5 and 0.7. For lower values, the delivery delay is more relevant, whereas the queueing delay dominates for $\lambda > 0.7$ and approaching the Heavy Traffic condition. However, these conclusions depend on the specific scenario.

V. Conclusions

In this letter we studied an exact Markov model to investigate the delay statistics by considering the effect of the arrival process. We derived the statistics of all delay contributions in close form for a Bernoulli arrival process with arbitrary packet arrival rate λ. This allows us to quantify the overall delay and the single delay components not only as average values but with detailed statistics. In particular, we analytically showed that the impact of the arrival process on the delivery delay is negligible for the majority of the cases, i.e., unless the error rate is unrealistically high and the arrival rate is low. Conversely, our analysis is of interest for the queueing delay evaluation, which is the most significant part of the overall delay when λ is high. Note that when the round-trip time m is large, this exact approach becomes prohibitive; therefore, approximate models, accurate enough to be used for practical purposes, are possible topics of future research.

References

[1] 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, 2000.

[2] 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, 1990.

[3] M. Rossi, L. Badia, and M. Zorzi, “On the delay statistics of SR ARQ over Markov channels with finite round-trip delay,” *IEEE Trans. Wirel. Commun.*, vol. 4, no. 4, pp. 1858–1868, 2005.

[4] M. Zorzi, R. R. Rao, and L. B. Milstein, “Error statistics in data transmission over fading channels,” *IEEE Trans. Commun.*, vol. 46, pp. 1468–1477, 1998.

[5] M. Rossi, L. Badia, and M. Zorzi, “On the delay statistics of an aggregate of SR-ARQ packets over Markov channels with finite round-trip delay,” in *IEEE WCNC 2003*, vol. 3, pp. 1773–1778, Mar. 2003.

[6] M. E. Anagnostou and E. N. Protonotarios, “Performance analysis of the Selective-Repeat ARQ protocol,” *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 127–135, 1986.

[7] J. Chang and T. Yang, “End-to-end delay of an adaptive Selective Repeat ARQ protocol,” *IEEE Trans. Commun.*, vol. 42, pp. 2926–2928, 1994.

[8] B. L. Long, E. Hossain, and A. S. Alfa, “Queuing analysis for radio link level scheduling in a multi-rate TDMA wireless network,” in *IEEE Globecom 2004*, vol. 6, Dallas, pp. 4061–4065, Dec. 2004.

[9] M. Rossi, L. Badia, and M. Zorzi, “SR-ARQ delay statistics on N-state Markov channels with finite round trip delay,” *Proc. IEEE Globecom 2004*, vol. 5, pp. 3032-3036, 2004.

[10] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*. New York: Dover Publications, INC., 1981.

[11] R. A. Howard, *Dynamic Probabilistic Systems*. New York: John Wiley & Sons, INC., 1971.

[4] Recall that the constant propagation term t_c is to be added to the delivery delay and hence also to the overall delay.

[5] For high arrival rates, simulation results are heavily sensitive to numerical approximations, since the buffer occupation might be very large and consequently the numerical evaluation is quite unstable.