This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Analysis of Selective Retransmission Techniques for Differentially Encoded Data

Leonardo Badia*, Marco Levorato†, Michele Zorzi†
* IMT Lucca Institute for Advanced Studies, piazza S. Ponziano 6, 55100 Lucca, Italy.
† Dept. of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy.
e-mail: {badia,levorato,zorzi}@dei.unipd.it.

***Abstract*—This paper presents an analytical framework for the study of Hybrid ARQ techniques aimed at the transmission of multimedia content with differential encoding. We propose a Markov model of a Selective Repeat Hybrid ARQ transmission scheme, where we assume that packets have different properties in terms of size, information content, and retransmission limit, so as to capture the differential encoding which characterizes such multimedia data. We consider a non-zero round-trip time channel modeled through a discrete-time Markov Chain. We provide an analytical tool for the evaluation of two main performance metrics, namely, throughput and goodput. The model is extremely flexible and allows evaluation of several channel conditions and comparison of different types of ARQ (plain ARQ, Type I and II hybrid ARQ). Our results can be used as an effective tool to define design guidelines of multimedia transmission systems and to understand their performance trends.**

***Index Terms*—Automatic repeat request, error analysis, video transmission, channel coding.**

I. Introduction

MULTIMEDIA data transmission is spreading over unreliable and lossy channels such as the wireless medium. However, this poses several challenges for error control. Pure Forward Error Correction (FEC) may lead to significant resource waste, whereas Automatic Repeat reQuest (ARQ) potentially results in additional delay. Both inefficient use of the available bandwidth and increased latency are undesirable effects when dealing with multimedia content.

For this reason, a solution may be to consider Hybrid Automatic Repeat reQuest (HARQ) [1], which is a widely employed combination of ARQ and FEC, meaning that information frames to be sent include error correcting codes which can repair some of the errors introduced by the channel (FEC approach). If the errors are too many to be corrected by the code, a retransmission may be triggered upon the receiver's request (ARQ approach) [2]. It is shown in [3] that video transmission schemes using hybrid error correction outperform those based on either pure ARQ or pure FEC.

When studying HARQ techniques, the data in the queue at the transmitter's side, waiting to be sent over the channel, are often modeled just as a collection of independent packets, which is not appropriate to describe certain kinds of data. Indeed, the transmission of video content can easily be achieved through a sequence of multiple independent pictures; this is the way adopted when video transmission uses the straightforward extension of the Joint Photo-graphic Experts Group (JPEG) standard [4], sometimes informally referred to as M-JPEG (Multiple JPEG), which obtains video transmission by simply transmitting subsequent JPEG frames. However, this approach is viable only for video flows where the frame rate is extremely low. To transmit higher quality video content, more efficient techniques must be used, where the correlation among different frames is exploited. A solution in this sense is represented by differential encoding techniques, such as Moving Pictures Expert Group (MPEG), which exploit inter-frame prediction [5]. This greatly reduces the bandwidth requirement for the transmission, and is a sensible approach for wireless links, where the capacity is often limited.

Thus, packets sent over the channel are not equivalent. For instance, a packet containing an independent picture may be worth of retransmission in case of failed delivery. In fact, its reception is of fundamental importance to also decode incrementally encoded packets, which in turn are less valuable.

This paper aims at capturing these aspects within an analytical framework for the performance evaluation of error control schemes with differentiated packets. We consider classic and hybrid retransmission-based techniques over a correlated channel with non-instantaneous feedback. We take a general approach with a source queue consisting of two different types of packets. Packets of the former type, called A-packets, are self-standing data elements which can be decoded independently of other packets. Those of the latter type, called B-packets, are instead differentially encoded from A-packets. This structure closely resembles the result of a network abstraction layer (NAL), i.e., the layer packetizing the video stream, in a H.264 codec [6]. Simply put, think of A-packets as the I-frames and of B-packets as P-frames in the MPEG standard [5].

Packets of both kinds are encoded, including also error protection, and transmitted over the channel with a known pattern. The transmission follows an HARQ scheme, which is applied differently for A-packets and B-packets, i.e., A-packets are retransmitted in a Selective Repeat (SR) ARQ/HARQ fashion [1], [7], whereas B-packets are simply discarded. The wireless channel is modeled with a Finite-State Markov channel, with known statistics, where every channel state describes an error level affecting the packets. Finally, we apply three different retransmission-based error control mechanisms, namely plain ARQ and Type-I and Type-II HARQ. The whole system is then represented with an appropriate Markov chain which takes into account the transmission status of the packet queue, and in particular the pending packets yet to be acknowledged, and the channel state. Through manipulation of this chain, we are able to analytically determine the statistics of the system throughput and goodput.

The rest of this paper is organized as follows. In Section II we describe the characteristics of the data flow and the two kinds of packets in detail. In Section III we outline the channel model and we discuss the implementation of ARQ and HARQ techniques. Section IV derives a Markov description of the entire transmission systems and Section V presents numerical results. Finally, we conclude in Section VI.

II. System Description

We consider the transmission of an indefinitely long message between a transmitter and a receiver over a discrete-time channel. The message is subdivided into packets of two different types, called A-packets and B-packets, respectively. A-packets are independently encoded; each of them is derived from an independent part of the message. B-packets are instead differentially encoded from A-packets. For simplicity, assume every B-packet has a single A-packet as its reference, and the size of a B-packet is exactly $1/P$ times that of an A-packet, where P is an integer. A time slot of the channel is taken equal to the transmission time of a B-packet, so that the transmission of an A-packet takes P slots.

Assume there are exactly $P(F-1)$ B-packets associated with every A-packet, with integer $F \geq 2$, so that the size of the entire group of an A-packet and its associated B-packets is F times that of an A-packet. In the following, we will call *packet-group* a set of packets which occupies FP time slots, i.e., with a size equal to that of an A-packet and its B-packets. Also, assume that B-packets are always transmitted after the A-packet they refer to. So, the transmission pattern initially consists of one A packet followed by its $P(F-1)$ associated B-packets, and this pattern is indefinitely repeated. These assumptions are not restrictive, as the analysis could still be performed, only with more cumbersome math, if they were relaxed or removed.

The receiver informs the transmitter about the correct/incorrect reception of packets by sending a positive/negative acknowledgement message, respectively, through a feedback channel. As usually done in the literature, we consider an error-free feedback channel. However, we model it as non-instantaneous: acknowledgements arrive at the transmitter after a round-trip-time of $m > 1$ slots. Again, it is not restrictive to assume that $m = MPF$, so that at every time exactly M packet-groups are ongoing.

When errors occur, they trigger a retransmission of the lost content. This may change the initial pattern of one A-packet followed by P B-packets. In fact, as is intuitive, the information content of A-packets is more important than that of B-packets, as correctly received B-packets are useless if their associated A-packet is in error. Thus, we make the following key assumption: only A-packets are retransmitted, whereas B-packets are discarded if in error. The retransmission of an A-packet occurs M packet-groups later, and P B-packets are omitted to make room for the retransmitted A-packet. Recall that each packet-group starts with an A-packet, so the $(k+1)$th A-packet is transmitted at time slots kPF through $kPF + P - 1$. If this A-packet is followed by $P(F-1)$ B-packets and is in error, the packet-group transmitted at time slots $(k+M)PF$ through $(k+M+1)PF-1$ will contain *two* A-packets and $P(F-2)$ B-packets.

The relative placement of the two A-packets within the packet-group is a free design choice. We conventionally assume that the new one, thereafter called *fresh* A-packet, is transmitted first, whereas the retransmitted one comes second. If either (or both) of the A-packets are in error, we apply the same rationale of retransmitting it (or them) M packet-groups later. More in general, we consider that A-packets always start being transmitted at time slots with an integer multiple of P as index. In other words, there are F possible *positions* for an A-packet to be transmitted within the packet-group. Therefore, each packet-group always starts with a fresh A-packet in its *head position* (the first P slots). Moreover, any retransmitted A-packet starts being transmitted exactly MFP slots after the end of its previous transmission (i.e., $MFP+P$ slots after its beginning). This means that an incorrect A-packet is retransmitted M packet-groups later and shifts one position (i.e., P slots) to the right within the packet-group. We recognize the number of transmissions already performed by an A-packet as its position within the packet-group. In fact, a fresh A-packet, which is at its first transmission, occupies the head position. If retransmitted, it moves to the second position (the first P slots being occupied by a newly transmitted A-packet); if retransmitted again, to the third, and so on.

We set the maximum number of transmissions for an A-packet to F. That is, when a retransmitted A-packet reaches the end position of the packet-group it can no longer be re-transmitted and is discarded if in error. We could, alternatively, allow a cyclical shift of A-packets; however, in practice the retransmission of multimedia content does not occur too many times, as the application delay requirements make an obsolete packet useless, thus a cyclical shift has little practical value.

B-packets take the available places not occupied by (fresh or retransmitted) A-packets. This allows for a soft degradation of the multimedia quality, since in the presence of channel errors, the most important data (A-packets) are retransmitted, whereas the amount of differentially encoded data (B-packets) is correspondingly decreased to make room for retransmissions. The fact that a fresh A-packet is transmitted at every packet-group guarantees that at least part of the continuous stream of multimedia information produced by the source can be accommodated. The overall transmission scheme is described in Fig. 1. Here, we visually describe the retransmission scheme, the cyclic shift of pending packet-groups, as well as the shift occurring to A-packets within a packet group.

III. Channel and Error-control Techniques

To determine the outcome of packet transmissions, we represent the channel with an $(L+1)$-state Discrete-Time Markov Chain (DTMC), with transition matrix $\mathbf{T} = \{t\}_{ij}$, which will be referred to in the following as *channel chain*. Thus, the channel state is constant within each time slot, and makes transitions from a slot to the next according to $\mathbf{T}$. This representation is often utilized for the wireless channel [8].

We assume that transmitted packets are only subject to erasures, [1] and every channel state $0, 1, \ldots, L$ represents an additive "error level" of the time slot, that is, state 0 corresponds to an error-free slot, whereas state L implies an entirely erroneous slot (i.e., all bits are correct with probability $1/2$) and states from 1 through $L-1$ describe proportionally different amounts of erasures, so that if the channel is in state j a fraction of the transmitted content equal to j/L is erased in

[1] Extensions to other kind of errors instead of erasures is straightforward, by modifying some of the system parameters.

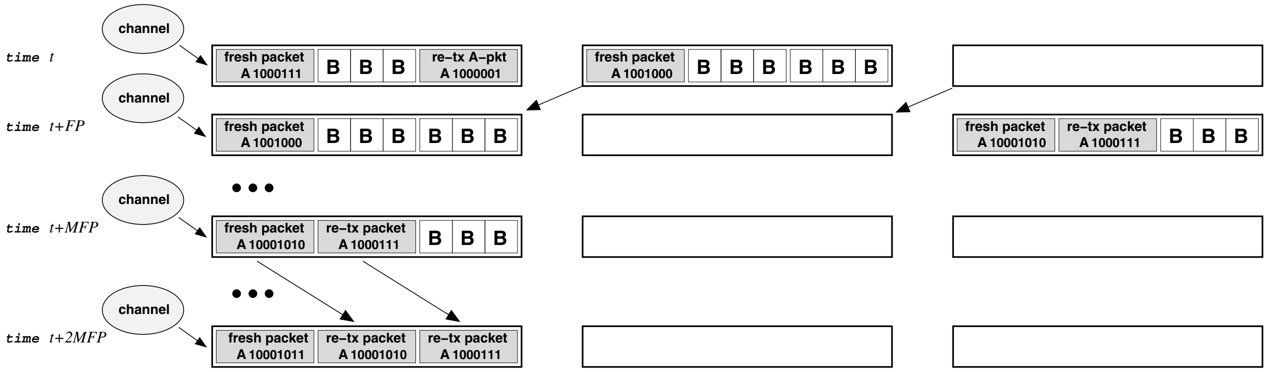

Fig. 1. The transmission system and an example of evolution over time for M=3, F=3. A-packets and B-packets are pictured by grey longer boxes and white squares, respectively. Every FP slots, packet-groups cyclically shift to the left, whereas retransmitted A-packets shift to the right within the packet-group.

that slot. Since B-packets occupy a single time slot, their error level is equal to the state of that slot. Instead, A-packets can be described by an error level j equal to the sum of the states of the P slots they span over, so that their relative amount of erasures is $j/(LP)$. This means that the error level of a B-packet can take $L+1$ values (one per channel state), whereas that of a fresh A-packet has $LP+1$ possible values, from 0 to LP.

We model different retransmission-based error control techniques. The simplest case is *plain ARQ*, denoted in the following as "ARQ-0," which considers any packet (of both types A and B) as correct if its overall error level is 0, and incorrect otherwise. Retransmission of an A-packet is triggered if the channel state is non-zero in any of its slots. A-packets are retransmitted identically, so the error level of a retransmitted A-packet can take $LP+1$ possible values, as a fresh one.

However, FEC techniques are often employed in order to improve the performance of plain ARQ, so as to realize a hybrid ARQ scheme. A first version is Type-I ARQ, which we will refer to in the following as "ARQ-1," corresponding to applying an error-correcting code to the packets. In this paper, we consider Reed-Solomon block codes [9]. We utilize a parameter $\theta < L$ to denote that the code has rate equal to $1-\theta/L$, so that up to θ/L of a packet can be corrected if affected by errors. In general, one can use different codes for A-packets and B-packets, hence different values of θ, denoted with θ_A and θ_B, respectively. As in ARQ-0, also ARQ-1 retransmits A-packets in a memoryless manner, therefore their error level can take $LP+1$ values. A-packets and B-packets can be acknowledged provided that their error levels are below $\alpha_0 = P\theta_A$ and θ_B, respectively. ARQ-0 can be seen as a special case of ARQ-1 where $\theta_A = \theta_B = 0$.

The basic mechanisms of ARQ-0 and ARQ-1 are the same, the only difference being which packets are considered erroneous. With some extensions we may similarly represent also Type-II Hybrid ARQ, which, for brevity, we denote as "ARQ-2." In such a system, B-packets, which are not retransmitted, are treated identically to ARQ-1. Instead, each transmitted (and retransmitted) A-packet is a different part of a longer codeword whose size is FP packets and has code rate $(1-\theta_A/L)/F$, so that it can be seen as a codeword of the truncated code whose rate is $1-\theta_A/L$, as was for ARQ-1.

What distinguishes ARQ-2 from ARQ-0 and ARQ-1 is that at each retransmission a physically different A-packet, i.e., the next part of the long codeword, is sent over the channel, since a "retransmission" refers to sending additional redundancy for a given information content, rather than repeating the corrupted packet. Nevertheless, in this case we speak also of retransmission, in order to have a homogeneous notation. A received A-packet can be acknowledged when it can be selectively combined with its previously received versions (which are, in reality, *different* parts of the codeword) to correctly extract the information content it carries. Now, after k transmissions the error level of the juxtaposition of the k received packets is simply the sum of their respective error levels. Thus, the error level of an A-packet at its $(k+1)$th transmission can be between 0 and $(k+1)LP$ ($(k+1)LP+1$ values). In case of linear codes, if this value is below $\alpha_k = kL+\alpha_0$ the A-packet is acknowledged. For the sake of notational uniformity, we write $\alpha_k = \alpha_0$ for ARQ-1, where no incremental redundancy is adopted. Similarly, we could take $\alpha_k = 0$ for ARQ-0.

IV. Markov Chain of the System

The system can be analyzed by taking the channel chain and the distinctive transmission pattern of A-packets and B-packets as inputs, and deriving a larger Markov chain, describing the whole ARQ process, which we will call in the following *ARQ chain* [7]. For graphical support, look again at Fig. 1.

A. System state descriptors

Since the evolutions of different packet-groups do not affect each other, we choose the time step of the ARQ chain as FP slots. That is, transitions of the ARQ chain occur only in correspondence with the start of a new packet-group transmission. The state of the ARQ chain at current time t includes the outcome $\mathbf{S}(t)$ of the M pending packet-groups, which may already be determined, but is unknown at the transmitter's side, plus the channel state $c(t)$ at current time. To simplify the notation, we will hereafter omit obvious time indices. Thus, the system state will be written as $(\mathbf{S}, c)$, with $\mathbf{S}$ being the M-tuple of vectors $(\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(M-1)})$, where vector $\mathbf{s}^{(j)}$ describes the packet-group that started transmission at time $t+(j-M)FP$. This means that the packet-group described by $\mathbf{s}^{(0)}$ was transmitted MFP slots before and its not-acknowledged A-packets are therefore about to be retransmitted in the next packet-group (unless they have already been retransmitted $F-1$ times, in which case they are discarded).

Each of the $\mathbf{s}^{(j)}$ is identical in size and contains the following elements: first, an error level descriptor $d_0^{(j)}$ for the head position, which always contains the error level of a fresh A-packet, thus it can take $LP+1$ possible values. After that, the remaining $F-1$ positions within the packet-group are associated with the $F-1$ pairs $(d_1^{(j)}, g_1^{(j)})$, $(d_2^{(j)}, g_2^{(j)})$,..., $(d_{F-1}^{(j)}, g_{F-1}^{(j)})$. The element $d_k^{(j)}$ is, similar to $d_0^{(j)}$, an error level descriptor. However, as the position can be occupied by either an A-packet or P B-packets, we also need $g_k^{(j)}$ to specify this, as well as to carry additional information.

We have the following alternatives: (i) the kth position contains P B-packets; in this case, $d_k^{(j)}$ simply describes how many of them are in error (we do not need any additional information as B-packets are not retransmitted), for which there are $P+1$ possibilities, and $g_k^{(j)}$ is set to zero; (ii) the kth position contains an A-packet at its kth transmission; thus, $d_k^{(j)}$ describes its error level, which can take either $LP+1$ or $kLP+1$ possible values, according to the ARQ scheme being either of ARQ-0 and ARQ-1, or ARQ-2, respectively.

Moreover, $g_k^{(j)}$ must be non-zero, to distinguish this case from the previous one; hence, we set $g_k^{(j)} = \psi_k^{(j)} + 1$, where $\psi_k^{(j)}$ is non-negative. Its meaning is explained by remarking that the system state must also track the number of B-packets which are associated with a (still) not-acknowledged A-packet and are correctly received. Such packets were transmitted within the packet-group where their associated A-packet was in head position; however, they can be decoded only when the A-packet is correctly received. As their associated A-packet is still pending, at present time it is undecided whether they contribute or not to the throughput. This uncertainty can only be removed when the A-packet is either acknowledged or discarded, none of which has happened yet. However, as the B-packets are not retransmitted, we need to attach their related information (actually, their number is what matters) to the status of the A-packet itself to keep it in memory. Thus, $g_k^{(j)} > 0$ means that the kth position is occupied by an A-packet (rather than by P B-packets which imply $g_k^{(j)} = 0$), and the value $\psi_k^{(j)} = g_k^{(j)} - 1$ is the number of B-packets associated with it which have been correctly received. If the A-packet is resolved, $\psi_k^{(j)}$ is to be read as a further increment to the system throughput. Note that, as B-packets are not retransmitted, the value of $\psi_k^{(j)}$ was already determined once and forever after the transmission of the packet-group where the A-packet was in head position, i.e., kM packet-groups before the current one. As $\psi_k^{(j)}$ can be from 0 to P, $g_k^{(j)}$ can take $P+2$ values.

B. System evolution

We now outline the transitions of the ARQ chain. As the transitions proceed in steps equal to the packet-group size (FP time slots of the channel chain), we can generate all possible combinations of FP channel transitions. We denote with $\mathbf{x} = (x_1, x_2, \ldots, x_{FP})$ a possible sequence of channel states and with $p(c, \mathbf{x})$ the probability that, when the channel is $c = c(t)$, we have, through the next FP time slots, a transition from c to x_1, then to x_2, and so on, until x_{FP} is reached at the FPth transition, i.e., $x_{FP} = c(t+FP)$. Due to the Markov nature of the channel, we have

$$p(c, \mathbf{x}) = t_{cx_1} \prod_{j=1}^{FP-1} t_{x_j x_{j+1}} \,. \quad (1)$$

Moreover, given vector $\mathbf{x}$ we define $\phi(\mathbf{x}, \sigma, \tau) = \sum_{j=\sigma}^{\tau} x_j$. Also, call $\mathcal{X}$ the set of all possible FP-sized sequences $\mathbf{x}$. As any x_j belongs to $\mathbb{Z}_{L+1} = \{0, 1, \ldots, L\}$, we have $\mathcal{X} = \mathbb{Z}_{L+1}^{FP}$.

We consider a transition from state $(\mathbf{S}, c) = (\mathbf{S}(t), c(t))$ at time t to state $(\mathbf{S}', c') = (\mathbf{S}(t+FP), c(t+FP))$ at time $t+FP$, where $\mathbf{S} = (\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(M-1)})$ and $\mathbf{S}' = (\mathbf{s}'^{(0)}, \mathbf{s}'^{(1)}, \ldots, \mathbf{s}'^{(M-1)})$. The probability of such a transition can be computed as follows. For all $j = 0, 1, \ldots, M-2$, $\mathbf{s}'^{(j)} = \mathbf{s}^{(j+1)}$; this describes the left-shift of the packet-groups at each ARQ chain transition. The computation of $\mathbf{s}'^{(M)} = (d_0'^{(M)}, (d_1'^{(M)}, g_1'^{(M)}), \ldots, (d_{F-1}'^{(M)}, g_{F-1}'^{(M)}))$ depends instead in part on the value of $\mathbf{s}^{(0)}$, as the shift is partially cyclical, but also it is affected by channel transitions. Assume we have a sequence of FP channel states represented by $\mathbf{x}$, which happens with probability $p(c, \mathbf{x})$. The first element of $\mathbf{s}'^{(M)}$, i.e., $d_0'^{(M)}$, represents the error level of the fresh A-packet just transmitted, which is equal to $\phi(\mathbf{x}, 1, P)$.

The next element is the pair $(d_1'^{(M)}, g_1'^{(M)})$, for which we have two alternatives: either the A-packet transmitted in the head position M packet-groups earlier was acknowledged or it was not. This depends on $d_0^{(0)}$ being below θ_A or not. In the former case, the position is occupied by P B-packets, whose error levels are given by $x_{P+1}, \ldots, x_{2P}$. Therefore, we set $g_1'^{(M)} = 0$ and $d_1'^{(M)} = \sum_{j=P+1}^{2P} \chi(x_j \le \theta_B)$, where $\chi(Y)$ is the characteristic function of the Boolean condition Y, i.e., is equal to 1 if Y is true, and 0 otherwise. In other words, $d_1'^{(M)}$ is set to the number of correctly received B-packets. Else, the position is occupied by a single A-packet, whose error level is the *sum* of $d_0^{(0)}$ and $\phi(\mathbf{x}, P+1, 2P)$. Thus, $d_1'^{(M)} = d_0^{(0)} + \phi(\mathbf{x}, P{+}1, 2P)$ and $g_1'^{(M)} = 1 + \sum_{j=1}^{F-1} (d_j^{(0)} \chi(g_j^{(0)} = 0))$. The latter expression for $g_1'^{(M)}$ means that, in case $d_0^{(0)}$ indicates that the fresh A-packet of M packet-groups earlier was not acknowledged, it is retransmitted now in position $k=1$, and we set a non-zero value of $g_1'^{(M)}$ equal to 1 plus the number of B-packets of the same packet-group that were correctly received. This last value can be computed by summing the values of the elements $d_j^{(0)}$ for all the positions from $j=1$ to $F-1$ where the element $g_j^{(0)}$ is zero (which denotes that this is a group of B-packets rather than an A-packet).

Subsequent pairs $(d_k'^{(M)}, g_k'^{(M)})$ with $k>1$ originate from $(d_{k-1}^{(0)}, g_{k-1}^{(0)})$ and have three alternatives, according to the packet-group which was transmitted M packet-groups earlier: (i) if position $k-1$ was occupied by P B-packets, the same holds now for position k; this is the case when $g_{k-1}^{(0)} = 0$. (ii) if position $k-1$ was occupied by an A-packet which was acknowledged, again position k is now taken by P B-packets; this is the case when $g_{k-1}^{(0)} > 0$ and $d_{k-1}^{(0)} \le \alpha_k$. (iii) if position $k-1$ contained an erroneous A-packet, a retransmitted version of this A-packet occupies then the kth position of packet-group M; this case occurs when $g_{k-1}^{(0)} > 0$ and $d_{k-1}^{(0)} > \alpha_k$.

In cases (i) and (ii), P B-packets are transmitted; we proceed in analogy with position 1, i.e., $d_k'^{(M)} = \sum_{j=kP+1}^{(k+1)P} \chi(x_j \le \theta_B)$ and $g_k'^{(M)} = 0$. In (iii), an A-packet is retransmitted; thus, $d_k'^{(M)} = d_{k-1}^{(0)} + \phi(\mathbf{x}, P+1, 2P)$, which is again analogous to what computed for position 1, whereas $g_k'^{(M)}$ is simply inherited from the previous state, since no further B-packets associated with that A-packet are transmitted, so that $g_k'^{(M)} = g_{k-1}^{(0)}$. (Note the transition to the right of the retransmitted A-packet within the packet-group.) Finally, the channel state switches from c to c' with probability $\sum_{\mathbf{x}\in\mathcal{X}}(p(c,\mathbf{x})\chi(x_{FP} = c'))$.

We can solve this Markov chain by computing the steady-state probabilities $\pi(\mathbf{S}, c)$ of the system state being $(\mathbf{S}, c)$, for every possible value of $(\mathbf{S}, c)$, as done in [10]. According to the discussion reported above, these are determined by solving the following steady-state equations

$$\pi(\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(M-1)}, c) = \qquad (2)$$
$$= \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{w}\in\mathcal{W}} \Big(\pi(\mathbf{s}', \mathbf{s}^{(0)}, \ldots, \mathbf{s}^{(M-2)}, x_1) \cdot p(\mathbf{x}, c)\Big) ,$$

where $\mathcal{W} = \mathcal{W}(\mathbf{x}, \mathbf{s}^{(M-1)})$ is the set of possible values for the first position from which $\mathbf{s}^{(M-1)}$ can be obtained if the sequence of channel states is $\mathbf{x}$.

The element $\mathbf{s}' = (d_0', (d_1', g_1'), \ldots, (d_{F-1}', g_{F-1}'))$ can be put in set $\mathcal{W}$ if the following conditions are verified:

$$d_0^{(M-1)} = \phi(\mathbf{x}, 1, P)$$
$$\text{if } g_1^{(M-1)} > 0: \quad d_0' = d_1^{(M-1)} - \phi(\mathbf{x}, P+1, 2P) > \alpha_0$$
$$\text{if } g_1^{(M-1)} = 0: \quad d_0' \le \alpha_0$$
$$\forall k \text{ from } 1 \text{ to } F-2:$$
$$\text{if } g_{k+1}^{(M-1)} > 0: \quad g_k' = g_{k+1}^{(M-1)} \text{ and } d_k' > \alpha_k$$
$$d_k' = d_{k+1}^{(M-1)} - \phi(\mathbf{x}, kP+1, (k+1)P)$$
$$\text{if } g_{k+1}^{(M-1)} = 0: \quad g_k' = 0 \text{ or } d_k' \le \alpha_k ,$$
$$\text{and } d_k^{(M)} = \sum_{j=kP+1}^{(k+1)P} \chi(x_j \le \theta_B)$$

Note that the last position ($k = F-1$) of the packet-group described by $\mathbf{s}'$ has no influence whatsoever, as the packets contained in this position, whether a single A-packet or P B-packets, are not retransmitted in any case. The set of steady-state equations can be solved by including the condition that $\sum_{\mathcal{S}} \pi(\mathbf{S}, c) = 1$, where $\mathcal{S}$ describes the set of all possible system states. Once this system of equations is solved, we can derive some interesting metrics. We remark that this entirely analytical approach also allows the derivation of the statistics of every order, not just the average value.

C. Performance metrics

We can evaluate the mean *throughput* Θ as the average number of correctly delivered packets at each transition of the ARQ chain, over $1+(F-1)P$, which is the maximum number of them, as in the best case the system acknowledges one A-packet and its associated B-packets, without any retransmission. To determine this, we need to weigh the contribution to Θ given by any possible state $(\mathbf{S}, c)$ in $\mathcal{S}$ over the steady-state probability $\pi(\mathbf{S}, c)$ and sum. To determine how state $(\mathbf{S}, c)$ contributes to the throughput, we focus on its first packet-group, i.e., we take $\mathbf{s} = \mathbf{s}^{(0)}$. Note that, as the packet-groups shift in a deterministic manner, we could also take another element and perform the very same computation. There are many ways in which $\mathbf{s} = (d_0, (d_1, g_1), \ldots, (d_{F-1}, g_{F-1}))$ contributes to Θ. First, when the A-packet in head position is correct, hence we check whether d_0 denotes an acknowledged A-packet or not. Conditioned to this case, B-packets also contribute to Θ; thus, we add to the throughput contribution the value of all d_k with $k > 0$ for which $g_k = 0$. Finally, the throughput is also increased by retransmitted A-packets which are acknowledged, so we sum 1 for each $g_k > 0$ where $d_k \le \alpha_k$. Additionally, we also count $g_k - 1$ as a further contribution (recall that any $g_k > 0$ describes an number of B-packet which were correctly received but are still waiting for their associated A-packet to be correctly decoded). To sum up, we have

$$\Theta = \Big(1 + (F-1)P\Big)^{-1}$$
$$\cdot \sum_{(\mathbf{S},c)\in\mathcal{S}} \bigg\{ \pi(\mathbf{S}, c) \Big[\chi(d_0 \le \alpha_0) \Big(1 + \sum_{k=1}^{F-1} \Big(d_k\, \chi(g_k = 0) \Big) \Big)$$
$$+ \sum_{k=1}^{F-1} \Big(\chi(d_k \le \alpha_k)\, \chi(g_k > 0)\, g_k \Big) \Big] \bigg\}$$

However, the computation above only evaluates the correctly received packets, which include some redundancy if a HARQ scheme (ARQ-1 or ARQ-2) is used. Similarly to Θ we can compute the *goodput*, denoted with Υ, where we count only the information bits gained per each delivery of an A-packet or a B-packet, which means we multiply each throughput contribution times $r_A = 1 - \theta_A/L$ or $r_B = 1 - \theta_B/L$ accordingly. We have

$$\Upsilon = \Big(1 + (F-1)P\Big)^{-1}$$
$$\cdot \sum_{(\mathbf{S},c)\in\mathcal{S}} \bigg\{ \pi(\mathbf{S}, c) \Big[\chi(d_0 \le \alpha_0) \Big(r_A + r_B \sum_{k=1}^{F-1} \Big(d_k\, \chi(g_k = 0) \Big) \Big)$$
$$+ \sum_{k=1}^{F-1} \Big(\chi(d_k \le \alpha_k)\, \chi(g_k > 0) \Big(r_A + r_B(g_k - 1) \Big) \Big) \Big] \bigg\}$$

V. Numerical results

We investigate throughput and goodput for the following values $P = 2$, $F = 3$, $M = 2$, which means that a B-packet is half the size of an A-packet, A-packets can be transmitted up to 3 times, and there are two packet-groups pending at any instant. The data stream generates 4 B-packets per A-packet but, according to the description made in Section II, this amount can be reduced to make room for retransmissions.

We consider a 5-state channel chain, i.e., the error level of a time slot can be between 0 and $L = 4$. The transition of the channel chain are determined by the following rules. The marginal distribution of the channel is binomial with success probability ε, i.e., the channel starts in state j with probability $\binom{L}{j}\varepsilon^j(1-\varepsilon)^{L-j}$. To introduce correlation, we define a probability w that the channel value is repeated in the next time slot, else it is generated again with the same binomial distribution. This process mimics a block fading with variable block length [11]; in fact, the channel stays constant for an average block length equal to $1/(1-w)$.

We consider the following cases: plain ARQ (ARQ-0), type-I ARQ (ARQ-1) with coding rate $3/4$ for both A-packets

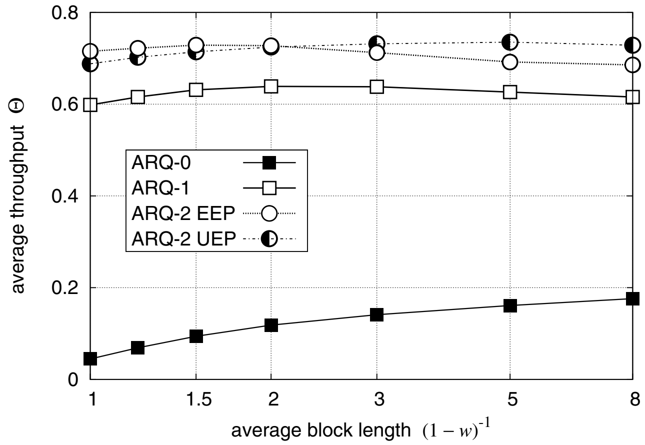

Fig. 2. Throughput of Selective Repeat for $\varepsilon = 0.3$ as a function of the average fading block length $1/(1-w)$.

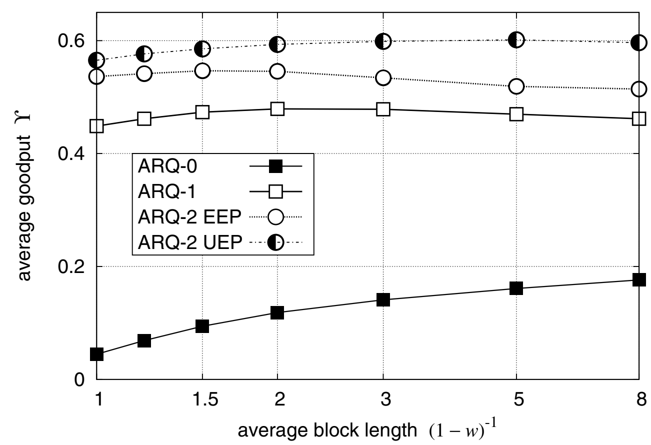

Fig. 3. Goodput of Selective Repeat for $\varepsilon = 0.3$ as a function of the average fading block length $1/(1-w)$.

and B-packets, and two different applications of type-II ARQ (ARQ-2), where we apply different coding strategies to A-packets and B-packets. Namely, we consider a case where we again apply the same FEC coding with rate $3/4$ to both kinds of packets, as in the ARQ-1 case (however, as per the previous discussion, we also combine packets, differently from ARQ-1). This case is referred to as ARQ-2 EEP, which stands for "ARQ-2 with Equal Error Protection." Moreover, we also consider a case with *Unequal Error Protection*, called ARQ-2 UEP, where we adopt a FEC with rate $3/4$ only to B-packets. A-packets follow instead what is sometimes called a Type-III ARQ [12], i.e., they have no FEC in the first transmission, whereas retransmissions only consist of redundancy. This means that the A-packets are obtained as F chunks of a codeword with rate $1/F$.

We set the value of ε to 0.3 and we change the value of w, to investigate the effect of correlation on the performance of the various schemes. Fig. 2 plots the throughput as a function of the average fading block length $1/(1-w)$. We can see that hybrid retransmission-based techniques outperform plain ARQ, which achieves very low throughput. When the channel is uncorrelated, i.e., $w = 0$, ARQ-2 EEP is better than ARQ-2 UEP. However, if the correlation is increased, ARQ-2 UEP improves whereas ARQ-2 EEP degrades. The fact that unequal error protection may better exploit differential encoding might be justified by observing that ARQ-2 UEP reduces the overhead for the A-packets which can also be corrected through retransmissions.

The throughput of ARQ-1 and ARQ-2 also accounts for redundancy, so it may be worth looking at the goodput instead, which is reported in Fig. 3, again as a function of the average fading block length. Clearly, this value is identical to the throughput for ARQ-0, whereas it is lower for the other techniques, but still stays above the performance of ARQ-0, thus justifying the additional protection through coding. Moreover, observe that ARQ-2 UEP, thanks to the absence of redundancy in the first transmission of A-packets, has a proportionally lower decrease than other techniques.

To sum up, these sample results show the ability of our proposed analytical framework to compare different ARQ schemes and analytically evaluate their performance for multimedia transmission with differentially encoded data.

VI. Conclusions and Future Work

We presented an analytical framework, based on Markov process theory, to study retransmission-based error control for multimedia data with differentially encoded packets, transmitted over correlated lossy channels. A significant application of our analysis is the delivery of video content over unreliable channels such as the wireless medium. Due to the expected increase of this and similar multimedia data transmission over wireless, this framework may be useful to evaluate and compare different error control schemes. As a further development, we are currently investigating the addition of other performance metrics, such as delay statistics, in order to enable a complete analysis of the multimedia transmission performance and its inherent tradeoffs.

References

[1] 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, 1984.
[2] H. O. Burton and D. Sullivan, "Errors and error control," *Proceedings of the IEEE*, vol. 60, no. 11, pp. 1293–1301, 1972.
[3] F. Zhai, Y. Eisemberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 40–51, 2006.
[4] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
[5] F. Fitzek and M. Reisslein, "MPEG4 and H.263 video traces for network performance evaluation," *IEEE Netw.*, vol. 15, no. 6, pp. 40–54, 2001.
[6] Y. Pourmohammadi Fallah, H. Mansour, S. Khan, P. Nasiopoulos, and H. Alnuweiri, "A link adaptation scheme for efficient transmission of H.264 scalable video over multirate WLANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 875–887, 2008.
[7] L. Badia, M. Levorato, and M. Zorzi, "Markov analysis of selective repeat type II hybrid ARQ using block codes," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1434–1441, 2008.
[8] M. Rossi, L. Badia, and M. Zorzi. "SR ARQ delay statistics on N-state Markov channels with non-instantaneous feedback," *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1526–1536, 2006.
[9] S. B. Wicker, *Error control systems for digital communication and storage*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[10] M. Rossi, L. Badia, and M. Zorzi, "Exact statistics of ARQ packet delivery delay over Markov channels with finite round-trip delay," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1858–1868, 2005.
[11] E. Malkamaki and H. Leib, "Performance of truncated type–II hybrid ARQ schemes with noisy feedback over block fading channels," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1477–1487, 2000.
[12] S. Kallel, "Complementary punctured convolutional (CPC) codes and their applications," *IEEE Trans. Commun.*, vol. 43, no. 6, pp. 2005–2009, 1995.