Markov Analysis of Video Transmission Based on Differential Encoded HARQ

Valentina Vadori, Anna V. Guglielmi, and Leonardo Badia
Dept. of Information Engineering, University of Padova, via Gradenigo 6B, 35131 Padova, Italy
email: {vadoriva, guglielm, badia}@dei.unipd.it

Abstract—**In this paper, we analyze hybrid automatic repeat request applied to the transmission of video content over the wireless channel. Retransmission-based techniques are usually applied to queueing systems assuming a homogeneous flow of identical packets, which are all transmitted and possibly retransmitted in the same way. However, multimedia packets are encoded with incremental methods leveraging spatial and temporal redundancy and as such, they have different roles and should be treated differently by the retransmission mechanism. Therefore, our work considers a selective retransmission scheme with unequal error protection applied to a multimedia flow subdivided into distinguishable packets. We assume a binary channel with memory and non-zero round-trip time. We utilize discrete-time Markov chains to model the channel and the transmission/retransmission system. This enables a closed-form derivation of performance metrics via Markov analysis. Numerical results are discussed and possible implications on multimedia communications are evaluated.**

Index Terms—**Automatic repeat request, Markov processes, error analysis, queueing analysis, multimedia communications.**

I. Introduction

According to several recent investigations, video traffic is steadily growing. The annual Cisco report forecasts a huge increase due to the massive use of tablets and similar devices [1]. By 2019, nearly a million minutes of video content will cross the network, accounting for 80% of global traffic.

In this scenario, transmission paradigms to avoid network congestion and convey data correctly and efficiently become essential. In our work, we are concerned with error control techniques, which are exploited to attain the reliable data delivery. Usually, forward error correction (FEC) is employed, but some contributions argued that automatic repeat request (ARQ) is better, or useful as well [2].

If the data flow is packet-switched, both FEC and ARQ introduce redundancy in the individual packets, but for different purposes. FEC uses error correcting codes to recover the original data without any further interaction with the transmitter. ARQ uses instead codes for error detection; in case the received packets are corrupted, the receiver asks for their retransmission. This means that a bidirectional interaction happens between the sender and the receiver, where the latter replies to each packet, at least from a conceptual standpoint, with positive or negative acknowledgments (denoted as ACKs or NACKs, respectively), depending on the absence or presence of residual errors in the packet. We focus on a Selective Repeat (SR) ARQ approach [3]. This means that, whenever the transmitter receives a NACK, it selectively retransmits the erroneous packet only, i.e., the one that caused the NACK.

A combined solution for error control is *Hybrid* ARQ (HARQ), which fuses FEC and ARQ. Here, redundancy is introduced in the packets with a twofold purpose. First, the receiver checks data integrity and tries to recover corrupted data, according to FEC scheme. In case the number of errors is higher than the correction capability of the code (but not of its detection capability), the receiver sends a retransmission request to the transmitter as per the ARQ [4]. In our investigation we concentrate on this approach, which has been shown to contain the delay introduced by a basic ARQ scheme.

Video contents usually impose strict constraints on transmission time and may need a non-homogeneous error control, due to the presence of packets with different relevance in the video flow. Hybrid schemes are particularly suitable for dealing with such requirements. In this paper, we focus on a hybrid SR ARQ scheme. We consider a video flow made up by packets of different kinds and we assume that the error control mechanism distinguishes among them.

The different nature of video packets belonging to the same flow originates from the procedure used to encode them [5]. We classify them having in mind the Moving Picture Experts Group (MPEG) standard, or any similar technique. MPEG leverages temporal redundancy between subsequent frames by introducing three different types of frames: I (Intra), P (Predicted), and B (Bidirectionally predicted) [6]. For each of them, a distinct compression strategy is used. An I-type frame is processed using *Intra-frame* techniques, i.e., by considering the frame as a stand-alone unit and by simply removing the spatial redundancy (similar to what done for a JPEG picture). P-type and B-type frames are compressed through *Inter-frame* techniques, i.e., by considering predictions with respect to temporally preceding and/or subsequent frames. Both spatial and temporal redundancy are removed. As a result, a video sequence is divided into smaller subsets, each of which is called a *Groups of Pictures* (GoP), where the three types of frames alternate according to a specific pattern.

In this work, a multimedia flow is modeled as a sequence of packets that are sent from a source to a destination. We distinguish between *A*-packets, that, similar to I-frames of MPEG standard, are autonomous and are encoded independently from the others, and *B*-packets, that, like P- and B-frames, are encoded incrementally based on an *A*-packet. One independent *A*-packet and several *B*-packets incrementally encoded from

978-1-5090-2185-7/16/$31.00 ©2016

it are bundled together in a *packet-group*. This methodology, as well as the terminology, are directly inspired by [7], [8], even though the analysis of those papers is different. Similar approaches have been used also by [2], [9]. Indeed, this two-level approach (where also packet dependence is limited within a packet-group) allows for a tractable analysis, while still accounting for the system memory introduced by packet interdependencies. However, the ARQ scheme introduce other memory effects due to retransmissions. Moreover, since a successful detection of an *A*-packet is needed to correctly decode a *B*-packet, the hybrid SR ARQ scheme ought to keep this difference into account.

In this context, our contribution can be summarized as follows. We propose a mathematical model for the analysis of a hybrid SR ARQ scheme applied to the transmission of video content over the wireless channel. We model the channel through a Markov chain, whose main parameters are the average error burst length and the steady-state error probability. We describe a finite-state machine that tracks the transmission state of packets for which the transmitter is still waiting for the related ACK/NACK (i.e., the *pending packets*). Differently from previous contributions, our finite-state machine is analytically tractable even for systems with relatively large memory (i.e., with long round-trip time and large size of the packet-group), within the limits of a Markov analysis. Built on that, we propose a macroscopic Markov chain to model the whole video transmission system and we quantify some indicative performance metrics, namely throughput, average number of transmissions, and packet discarding probability, via Markov analysis. We identify the effect of the channel parameters on video quality and we extract some useful considerations about the implementation of the HARQ scheme in real scenarios. Our evaluation is orthogonal to the presence of dynamic adaptations to channel and source conditions at the physical layer; the model can be extended to also include rate optimization and adaptive modulation and coding techniques, which exploit the knowledge of the system state to further enhance video transmission in a cross-layer context [10].

The rest of this paper is organized as follows. In Section II we outline the related works. In Section III, we illustrate the HARQ transmission model and we specify the considered assumptions. Section IV presents the macroscopic HARQ system description and its evaluation as a result of Markov analysis. Section V discusses some numerical results; finally, Section VI concludes the paper.

II. Related work

In the literature, several papers relate to the analysis of error control techniques for wireless connections through mathematical models of the whole transmission system; based on these models, they evaluate some performance metrics, such as throughput and delay.

The papers most directly related to our contribution are [2], [7], and [11]; in all these papers we considered HARQ for video transmission with selective retransmission and unequal error protection, and we proved that retransmission-based techniques can be jointly used with pure FEC to improve the transmission. The proposed unequal error protection technique, that consists of retransmitting only the independently encoded packets and discard the incremental ones, was first introduced in [7], and it is the same considered here. However, the present paper is different from all these previous contributions in that [7] does not consider any analytical modeling of the Markov chain. Subsequently, [2] gives a full Markov model but for complexity reasons the round-trip time is limited to two packets. Finally, [11] extends this analysis to longer packet-groups but internal limitations of the model still limit the maximum number of retransmissions and/or packet interdependencies. This work is the first to relax all these constraints and give a general analysis where the only limit is the computational complexity of the resulting Markov chain.

Other papers are relevant to our investigations, since they also focus on an ARQ mechanism applied to video transmission, which poses two challenges: (i) how to introduce retransmission while at the same time still meeting the delay constraints of a real time application such as video; (ii) how to differentiate among packets, since multimedia packets have different roles within the flow.

For example, in [12], the authors try to reduce the delay and buffer size needed by the ARQ approach by using a Multi-ARQ scheme and a rescheduling technique for transmitting video streams over wireless channels. They subdivide a typical video stream into three sensitivity classes and apply the unequal error protection to them, using ARQ-schemes with higher reliability for higher sensitive portions of video data. In particular, SR ARQ is adopted for high sensitive video data, while no ARQ scheme is applied to low sensitive video data. Thus, a differentiation mechanism is applied to reduce delay, but this refers to the flow as a whole.

Also in [13], a prioritized ARQ scheme to enhance the error robustness of streaming video is defined. The proposed priority based retransmission scheme is instead adopted with packet importance information. The level of packet importance is measured by estimating the error propagation effect caused by the corresponding packet loss. Lost packets with higher impact values can be retransmitted while regular packets of less importance can be discarded to make room for retransmissions. A two-state Markov model analogous to ours is employed to generate the channel packet loss patterns. However, the importance level is specified individually for each packet, while we consider instead multiple levels depending on the structure of the packet-group. A similar priority model for the packets also appear in [14], where the highest priorities are given to the packets whose loss has stronger effect in the perceived video quality. An alternative model is proposed by [15] where ARQ is better matched with delay constraints typical of video content by setting a retransmission limit that may force to drop some packet if it is expected that they will not meet a pre-defined real-time deadline. This time, the prioritization of packets is purely based on delay aspects, not on their video characteristics.

Our proposal can also be regarded in the context of unequal error protection for video. Along these lines, the authors of [16] proposed a more detailed unequal error protection scheme than ours. It is also worth noting that they considered the impact of error propagation on the video quality, which is a very important issue [9]. Our model, based on a two-level abstraction, does not allow to explore this issue in more depth. An analogous proposal, but also involving a more comprehensive cross-layer optimization is suggested by [17]. However, these proposals are limited to FEC, without any discussion on whether this can be iteratively modified through ARQ. Thus, we believe that our proposal can be seen in addition to theirs as a complementing extension of their unequal error protection schemes.

Other related papers suggest possible expansions in line with our work at the network [18] and transport layer [19], basically identifying a possible development of hybrid ARQ in the application of network coding, and also envisioning a streaming control mechanism where ARQ is actually selectively applied but controlling, and possibly choking, the stream if delay constraints risk not to be met. Our proposed scheme can be superimposed to improve the performance of these scenarios as well.

III. Assumptions and model of HARQ transmission

The analyzed system consists of a transmitter and a receiver. The transmitter sends a multimedia flow to the receiver through the wireless channel, which is modeled as a noisy and correlated channel with known statistics. The flow is subdivided into packets, each of them representing an encoded information unit. As it is generally the case in ARQ analysis, we assume that the *Heavy Traffic* condition holds, i.e., the transmitter's queue is always backlogged and, thus, there are always packets to transmit.

The system adopts a hybrid SR ARQ technique with unequal error protection. Therefore, the receiver checks packets integrity upon reception. Whenever errors are detected, it tries to reconstruct the original data by exploiting the redundancy of channel coding in a FEC-like fashion. The transmitter is then informed about correct or incorrect packet reception through selective acknowledgments sent on a feedback channel. Positive and negative acknowledgments are denoted as ACKs and NACKs, respectively. The feedback channel is assumed not to be instantaneous and *error-free*, which is reasonable due to the short length of ACK/NACK messages. In the literature, there are examples of investigation about the consequences of errors affecting feedback messages [20]. The main conclusion is that the impact of errors roughly consists in a re-scaling of the error probability on the forward channel.

We consider two different types of packets, denoted as *A* and *B*. *A*-packets are *independently encoded*, meaning that they are stand-alone data units that can be decoded without requiring that any other packet is decoded first. Conversely, a *B*-packet is *incrementally encoded* from a single *A*-packet; thus, to be decoded, its correct reception is not sufficient, as the generating *A*-packet must be correctly received as well. Without loss of generality, it is assumed that *A*- and *B*-packets have the same length[1] and that, for each *A*-packet, the number of *B*-packets incrementally encoded from it is ℓ, $\ell \in \mathbb{N}, \ell \geq 1$. An $(\ell{+}1)$-element set consisting of an *A*-packet and its corresponding *B*-packets is referred to as a *packet-group*. *B*-packets are transmitted after the corresponding *A*-packet. As a consequence, *A*-packets and *B*-packets alternate within the data flow according to a precise pattern, which consists of an *A*-packet followed by ℓ *B*-packets. Apart from that, there is no dependence between *B*-packets. These assumptions are not restrictive, as the analysis could still be performed considering more interdependences among packets, only with more involuted math. Moreover, note that the *packet-group* can be considered as the analogous of a GoP in *MPEG*.

We assume that time is slotted and a slot is equal to the time needed to transmit a packet of either type. When a packet is transmitted, the corresponding acknowledgment arrives after a number m of slots equal to the *round-trip time*, which coincides with the ARQ window. This implies that, at any time instant, the number of pending packets does not exceed m. Without loss of generality, we consider $m = M(\ell{+}1){+}1$, with $M \in \mathbb{N}, M \geq 1$. If the round-trip time is not an integer multiple of the time slot, M may still be chosen such that m is the smallest integer higher than the round-trip time.

The unequal error protection requires that the *A*-packets for which the transmitter receives a NACK are retransmitted; erroneous *B*-packets are instead discarded. There are essentially two motivations for differentiating between *A*- and *B*-packets. First, *A*-packets are more important; indeed, to correctly decode *B*-packets it is necessary to successfully decode their corresponding *A*-packet. Second, the retransmission of all types of packets may lead to queue instability and long delays. Provided that retransmissions can be triggered for *A*-packets only, just a fraction of $\frac{1}{\ell+1}$ packets may undergo retransmissions and a system in which no additional delay increase occurs can be designed.

To avoid delay increases, we assume that retransmissions of *A*-packets are prioritized, replacing transmissions of *B*-packets in some following packet-group [7]. Moreover, *A*-packets can be retransmitted at most r times. We assume that the maximum number of transmissions r of a given *A*-packet satisfies $r \leq \ell + 1$. A packet-group always starts with a new *A*-packet, dubbed a "fresh" *A*-packet. If a fresh *A*-packet α_1 transmitted at slot k is followed by ℓ *B*-packets and is in error, then the packet-group transmitted at time slot $k + M(\ell + 1)$ through $k + (M + 1)(\ell + 1) - 1$ will contain two *A*-packets. A fresh *A*-packet, α_2, in slot $k + M(\ell + 1)$, i.e., in head position, and the second *A*-packet, α_1 at its first retransmission, in slot $k + M(\ell + 1) + 1$, in place of the first *B*-packet incrementally encoded from α_2. If α_1 results in error again, its second retransmission replaces the second *B*-packet in the packet-group transmitted at slot $k{+}2M(\ell{+}1)$ through $k + (2M + 1)(\ell + 1) - 1$, and so on. It follows that an *A*-packet at its nth retransmission is transmitted in place of the nth *B*-packet of a packet-group, exactly m slots after the beginning of its last transmission. The number of transmissions already performed by an *A*-packet can be recognized as its position within a packet-group. In fact, a fresh *A*-packet occupies the head position. If retransmitted, it shifts one position to the right, moving to the second position.

[1]The model actually does not consider the packet lengths. As long as the channel quality is approximately constant throughout each packet transmission, considering different packet lengths just complicates the model without adding any significant insight. See [2] for a detailed discussion on this point.

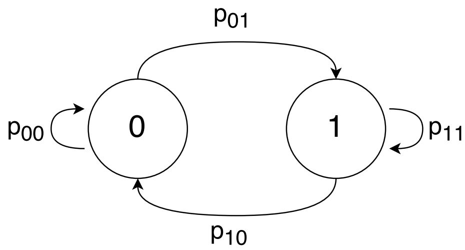

Fig. 1. Markov representation of the channel (channel chain).

If retransmitted again, it shifts to the third, and so on. This means that when a retransmitted *A*-packet reaches the end position of a packet-group it can no longer be retransmitted and is discarded if in error. *B*-packets fill the empty slots not used for retransmissions. In particular, for each fresh *A*-packet, at least $\ell - r + 1$ *B*-packets are transmitted.

To determine the outcome of packet transmissions, we model the channel state evolution through a discrete-time Markov chain (DTMC). This DTMC, referred to as *channel* chain, has two states: state 0, representing the error-free channel, and state 1, representing an erroneous channel. The channel DTMC is depicted in Fig. 1. This is a model similar to the Gilbert Elliot model, however, in our model a transmission performed when the channel is in state 0 or 1 is always successful or always erroneous, respectively [21]. The channel state is constant within each time slot, and makes transitions from a slot to the next according to the one-step transition probability matrix $\mathbf{P}$, which completely characterizes the channel chain. $\mathbf{P} = \{p_{ij}\}$, $i, j \in \{0, 1\}$, where p_{ij} is the probability of going from state i to state j in a single step,

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}. \tag{1}$$

Also note that transitions to K slots ahead are governed by the K-step transition probability matrix $\mathbf{P}^K$.

Matrix $\mathbf{P}$ can be used to infer information about the average number of consecutive erroneous slots, which we refer to as the *burstiness* B, and the packet error probability at steady-state, which we denote as ε. The burstiness quantifies the channel correlation and its value is $B = \frac{1}{p_{10}}$. The steady-state packet error probability is $\varepsilon = \frac{p_{01}}{p_{10}+p_{01}}$.

Even though the model seems simple, it allows to investigate the impacts of both the average error rate ε and the channel correlation, related to the burstiness B.

IV. Macroscopic HARQ System Description and Performance Metrics

Since the considered HARQ scheme uses selective retransmissions based on packet type and the channel has a non-zero round-trip time, our analysis requires to track previously transmitted packets. Indeed, in order to determine which packet should be transmitted at a generic slot k, it is necessary to know which type of packets have been transmitted at slot $k-m$ (i.e., the slot in which the packet was transmitted, whose acknowledgment is received at the beginning of slot k) through slot $k-1$. The reception of an ACK or NACK at the beginning of slot k then triggers the transmission of a new packet (*A* or *B*) or the retransmission of an already transmitted *A*-packet. Correspondingly, the model of the whole transmission system needs to memorize the types of pending packets, i.e. those for which the transmitter has not received the feedback message yet, and the channel state during their transmission [3].

As a first step, we define a finite-state machine that models the possible pending packets configurations, i.e. the possible sequences of *A*-packets and *B*-packets that at a given time slot k could be pending. Transitions occur at each time slot, leading to a new pending packet configuration that depends on ACK or NACK reception, with the ACK or NACK corresponding to the oldest pending packet at time slot k (i.e., that sent in slot $k - m + 1$). We refer to the machine states as *machine stages* and we indicate the number of machine stages as L. Each machine stage stands for a possible pending packets configuration and is denoted by a vector $\boldsymbol{\sigma} = [\sigma_0\, \sigma_1\, \ldots\, \sigma_{m-1}]$ of length m, with $\sigma_j \in \{0, 1, \ldots, r\}, 0 \leq j \leq m-1$. Considering that at time slot k the finite-state machine is in stage $\boldsymbol{\sigma}(k)$, we have that each $\sigma_j(k)$, for $0 \leq j \leq m-1$, identifies the type of packet transmitted at slot $k - m + 1 + j$ and the number of transmissions that this packet has already undergone. More precisely, $\sigma_j(k) = 0$ if the packet transmitted in slot $k-m+1+j$ is a *B*-packet; $\sigma_j(k) = n, n \in \{1, 2, \ldots, r\}$ if the packet transmitted in slot $k - m + 1 + j$ is an *A*-packet at its nth transmission. Note that the last element, $\sigma_{m-1}(k)$, refers to the packet transmitted at slot k, while the first element, $\sigma_0(k)$, refers to the packet transmitted at slot $k-m+1$, for which the transmitter receives the corresponding feedback at the end of slot k.

The number of machine stages, which influences the overall system complexity, can be obtained by considering that for any choice of the parameters ℓ, r, M, and thus $m = M(\ell+1)+1$, the finite-state machine would pass through $\ell + 1$ different machine stages only, provided that the channel was error-free. To see it, consider a pending packets configuration that contains fresh *A*-packets only and where the packet in head position is of type *A*. We indicate such configuration as $\mathcal{C}$. Since no retransmitted *A*-packet is present, the number of *B*-packets associated to each *A*-packet is ℓ. Therefore, the vector representation of $\mathcal{C}$ contains M consecutive sequences $[1\, 0\, \ldots\, 0]$ with ℓ zeros, the last one followed by value 1. Now, suppose that the channel is error-free. This implies that no NACK is received and that the finite-state machine indefinitely transits over the same cycle of $\ell + 1$ stages. The first one coincides with $\mathcal{C}$. The following ones can be obtained orderly by shifting one position to the left the values in the vector representation of the preceding stage, removing the first element and adding a 0 in last position. Since the number of shifts to the left that can be applied to the vector representation of $\mathcal{C}$ before returning to $\mathcal{C}$ itself is $\ell+1$, this is also the number of machine stages in the considered cycle, which we refer to as *ideal cycle*, since the channel is assumed to be error-free.

From the ideal cycle, we can derive the total number of stages of the finite-state machine for all pending packet configurations. The idea is to sum the number of stages of the idle cycle and the number of stages that the machine may pass through when the channel resulted to be erroneous in at least one pending packet transmission (and thus, the assumption of

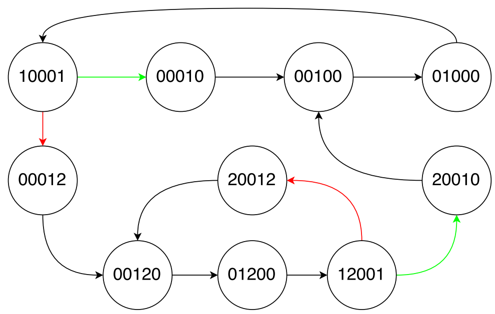

Fig. 2. Graphical display of the finite-state machine of pending packets configurations; $\ell = 3$, $r = 2$, $M = 1$, $m = M(\ell+1)+1 = 5$, $L = 10$.

error-free channel is removed). The stages in this last set have an associated vector representation where the 1s are the same (same number and position) as those on one of the stages of the idle cycle, while some 0 element is replaced by a value $x \in \{2, \ldots, r\}$, due to retransmissions of A-packets.

Consider for example the simplest case $r = \ell+1$ e $M = 1$. The number of machine stages with a vector representation where 1s are placed as in $\mathcal{C}$ are 2^ℓ, including $\mathcal{C}$. The number of machine stages with a vector representation where 1s are placed as in the stages following $\mathcal{C}$ in the ideal cycle is $2^{\ell+1}$ for each stage, including the stage of the ideal cycle. Thus, the total number of machine stages in this simple case is given by $L = 2^\ell + \ell(2^{\ell+1})$.

The same rationale, i.e. the definition of the idle cycle and the determination of stages *affine* to the stages of the idle cycle, can be applied to compute the total number of stages of a generic finite-state machine with arbitrary parameters. The following expression holds $L = [(r-1)2^r + (\ell - r + 2)2^{r-1}](2^{r-1})^{M-1} = (r+\ell)2^{M(r-1)}$.

Fig. 2 shows an example of finite-state machine considering $\ell = 3$, $r = 2$, $M = 1$, $m = M(\ell+1)+1 = 5$, $L = 10$. In Fig. 2, green arrows identify the transitions that occur when the transmitter receives an ACK for an A-packet; red arrows describe the transitions that occur when the transmitter receives a NACK for an A-packet that has not yet reached the maximum number r of transmissions. Black arrows represent transitions that occur when an ACK/NACK is received for a B-packet or when an ACK/NACK is received for an A-packet that has already been transmitted the maximum number of times, i.e., 2 times in the considered example. These last ones are forced transitions.

Given the channel DTMC and the finite-state machine of pending packets configurations, we develop a macroscopic DTMC that models that entire transmission system and which we refer to as HARQ chain, analogously to [7] [11]. The HARQ chain allows for the computation of the system steady-state probabilities which, in turn, are employed to compute three networking performance metrics, namely throughput, average number of transmissions to correctly deliver an A-packet, and A-packet dropping probability.

At each time slot k, the HARQ chain's state is described by a vector $\mathbf{S}(k) = [\boldsymbol{\sigma}(k)\, b_0(k) \, \ldots \, b_{m-2}(k)\, c(k)]$ subdivided into three different parts, each of which carries a specific information. $\boldsymbol{\sigma}(k)$ is the finite-state machine stage, as discussed previously. $\mathbf{b}(k) = [b_0(k) \, \ldots \, b_{m-2}(k)]$ describes the channel states from slot $k - m + 1$ through $k - 1$. In particular, $b_j(k) = 0$ if the channel at slot $k - m + 1 + j$ was error-free, and thus the packet transmitted at that slot was correctly received; $b_j(k) = 1$ if the channel at slot $k - m + 1 + j$ was erroneous, and thus the packet transmitted at that slot was erroneously received. $c \in \{0, 1\}$ indicates the channel state at slot k. The number of HARQ chain's states, N, can be obtained as $N = L \cdot 2^{m-1} \cdot 2$, where L is the number of machine stages of the finite-state machine. The value of N also reflects in the computational complexity of our model. As we will show later, we require the computation of steady-state probabilities, obtained by solving a linear system associated with a $N \times N$ matrix. This matrix is sparse and, as a consequence, we can compute the solution efficiently by using procedures whose (both space and time) complexity is $O(N)$. Moreover, N depends exponentially on m; however, it is possible to approximate this exponential dependence with a linear one, thereby decreasing the computational complexity, following the approach of [22].

At time slot k, $\boldsymbol{\sigma}(k+1)$ is univocally determined by $\boldsymbol{\sigma}(k)$ and the value of $b_0(k)$, which indicates whether the oldest pending packet at time slot k has been correctly received or, instead, needs to be retransmitted. Indeed, when $\boldsymbol{\sigma}(k)$ represents a set of pending packets with an A-packet in its head position and $b_0(k) = 1$, the A-packet retransmission is scheduled at time slot $k+1$, provided that this A-packet has not reached the maximum number r of allowed transmissions. The elements $b_j(k+1)$, $0 \le j \le m-3$, depend on the values of elements $b_j(k)$, $1 \le j \le m-2$, and on the value of $c(k)$. Finally, the value of $c(k+1)$ depends on the value of $c(k)$, according to the channel transition probability matrix $\mathbf{P}$.

The transition matrix $\mathbf{T}$ of the HARQ chain can be defined by observing that, given state $\mathbf{S}(k) = [\boldsymbol{\sigma}\, b_0 \, \ldots \, b_{m-2}\, c]$ (where we omit the dependance on k in the right-hand side of the equality for notation simplicity), the only admitted transitions towards a future state $\mathbf{S}(k+1)$ include

- $\mathbf{S}(k+1) = [\boldsymbol{\sigma}_c\, b_1 \, \ldots \, b_{m-2}\, c\, 0]$ with probability p_{c0}
- $\mathbf{S}(k+1) = [\boldsymbol{\sigma}_c\, b_1 \, \ldots \, b_{m-2}\, c\, 1]$ with probability p_{c1}
- $\mathbf{S}(k+1) = [\boldsymbol{\sigma}_e\, b_1 \, \ldots \, b_{m-2}\, c\, 0]$ with probability p_{c0}
- $\mathbf{S}(k+1) = [\boldsymbol{\sigma}_e\, b_1 \, \ldots \, b_{m-2}\, c\, 1]$ with probability p_{c1}

where $\boldsymbol{\sigma}_c$ and $\boldsymbol{\sigma}_e$ represent the pending packets configurations at time slot $k+1$ when the received feedback at the end of slot k is an ACK or NACK, respectively. Equivalently, they represent the pending packets configurations at time slot $k+1$ when $b_0(k) = 0$ and $b_0(k) = 1$, respectively. Table I shows $\boldsymbol{\sigma}_c$ and $\boldsymbol{\sigma}_e$ for each possible $\boldsymbol{\sigma}$, considering $\ell = 4$, $r = 3$, $M = 1$, $m = M(\ell+1)+1 = 6$, $L = 28$.

The HARQ chain is irreducible (there is a connection between each state) and each state is recurring (the probability that the system comes back to a state is 1 for each state). As a consequence, we can evaluate the steady-state probabilities $\boldsymbol{\pi} = [\pi_0\, \pi_1 \, \ldots \, \pi_{N-1}]^T$, i.e. the probabilities that the system is in a certain state at time slot k, for large k. We solve the system $\mathbf{T}\boldsymbol{\pi} = \boldsymbol{\pi}$, under the constraint that elements in $\boldsymbol{\pi}$ sum to 1. We use $\boldsymbol{\pi}$ to determine the performance metrics, as explained in the following.

TABLE I
TABLE OF THE σ-TRANSITIONS FOR $\ell = 4$, $r = 3$, $M = 1$, $m = M(\ell + 1) + 1 = 6$, $L = 28$.

$\boldsymbol{\sigma}$	**Configuration**	$\boldsymbol{\sigma}_c$	$\boldsymbol{\sigma}_e$
0	100001	1	5
1	000010	2	2
2	000100	3	3
3	001000	4	4
4	010000	0	0
5	000012	6	6
6	000120	7	7
7	001200	8	8
8	012000	9	9
9	120001	10	11
10	200010	2	12
11	200012	6	15
12	000103	13	13
13	001030	14	14
14	010300	18	18
15	000123	16	16
16	001230	17	17
17	012300	19	19
18	103001	22	23
19	123001	20	25
20	230010	21	26
21	300100	3	3
22	030010	21	21
23	030012	24	24
24	300120	7	7
25	230012	24	27
26	300103	13	13
27	300123	16	16

With respect to throughput, i.e., the amount of data correctly delivered in unit time, positive contributions are given by correct transmissions of *A*-packets and *B*-packets for which the corresponding *A*-packet has been correctly delivered. To quantify such contributions, we need to inspect the HARQ chain's states. If $\sigma_0 \in \{1, 2, \ldots, r\}$ and $b_0 = 0$, then the HARQ chain's state corresponds to a successful reception of an *A*-packet. Therefore, the throughput includes the sum of the steady-state probabilities associated to the HARQ chain's states in which $\sigma_0 \in \{1, 2, \ldots, r\}$ and $b_0 = 0$. If $\sigma_0 \in \{1, 2, \ldots, r\}$ and $b_0 = 1$, then the HARQ chain's state corresponds to a failed reception of an *A*-packet. In this case, if $\sigma_0 = r$, then the contribution to the throughput is zero; if $\sigma_0 \in \{1, 2, \ldots, r-1\}$, then further considerations need to be made, as it will be explained in the next paragraph. If $\sigma_0 = 0$ and $b_0 = 1$, then the HARQ chain's state corresponds to a failed reception of a *B*-packet and the throughput remains unchanged. If $\sigma_0 = 0$ and $b_0 = 0$, then the HARQ chain's state corresponds to a correct reception of a *B*-packet, which contributes to throughput on condition that the corresponding *A*-packet has successfully reached the destination.

In the following, we consider a system with parameters $\ell = 4$, $r = 3$, $M = 1$, $m = M(\ell+1)+1 = 6$ and we illustrate some examples of computations for throughput contributions. Consider $\boldsymbol{\sigma} = 000010$. The first packet of this configuration is of type *B*. The HARQ's chain states with such $\boldsymbol{\sigma}$ and $b_0 = 1$ do not contribute to throughput.

If $\boldsymbol{\sigma} = 000010$ and $b_0 = 0$, then we need to consider the transmission outcome of the *A* packet from which the *B* packet has been incrementally encoded. By observing $\boldsymbol{\sigma}$, we can deduce that an ACK has been sent for that A-packet. If this was not the case, the pending packets configuration would have been $\boldsymbol{\sigma} = 000012$. Therefore, the steady-state probabilities of the HARQ chain's states with $\boldsymbol{\sigma} = 000010$ and $b_0 = 0$ are added to the throughput.

Consider now $\boldsymbol{\sigma} = 000012$. The *A*-packet from which the *B*-packet in head position has been incrementally encoded is at its second transmission. If $b_0 = 0$ and $c = 0$ (i.e., the transmission of the *A*-packet is successful), then these states give a positive contribution to throughput. If $c = 1$, further reasonings are needed; the *A*-packet is transmitted for the third (and last, since we are assuming $r = 3$) time after 6 time slots starting from the current time slot, k. Therefore, if $c(k+6) = 1$, there is no throughput contribution. Otherwise, if $c(k+6) = 0$, there is a positive contribution that can be obtained, for every state with $\boldsymbol{\sigma} = 000012$, $b_0 = 0$, and $c = 1$, by multiplying the steady-state probability of such states by $\mathbf{P}^6(1, 0)$ (i.e., the entry at row 1 and column 0 in $\mathbf{P}^6$).

Finally, consider $\boldsymbol{\sigma} = 000120$. Similar observations hold. However, in this case the transmission outcome of the *A*-packet is described by b_4. Therefore, if $b_0 = 0$ and $b_4 = 0$, the contribution is positive, independently of c. Whereas, if $b_0 = 0$ and $b_4 = 1$, the contribution is positive if and only if the transmission of the *A*-packet after 5 time-slots from the current one is successful. In this case, the contribution is computed by summing the stationary probabilities associated to the states with $b_0 = 0$, $b_4 = 1$, and $c = 1$ multiplied by $\mathbf{P}^5(1, 0)$ and those associated to the states with $b_0 = 0$, $b_4 = 1$, and $c = 0$ multiplied by $\mathbf{P}^5(0, 0)$.

Another relevant metric is the average number of transmissions, which is however interesting only for *A*-packets, because *B*-packets are always transmitted just once. We define the conditional probabilities $\pi(n|A)$, $n = 1, \ldots, r$, as the probabilities that the system is observed when an *A*-packet is at its n^{th} transmission. Such probabilities are evaluated as the ratio between the sum of the steady-state probabilities of the HARQ chain's states with $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}_0 = n$ and the sum of the steady-state probabilities of the HARQ chain's states with $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}_0 \in \{1, \ldots, r\}$. For the following numerical choices of the parameters $\ell = 4$, $r = 3$, $M = 1$, $m = M(\ell + 1) + 1 = 6$, for example, the conditional probabilities can be expressed as, see [11]:

$$
\begin{aligned}
\pi(1|A) &= \frac{p_1 + p_2 + p_3}{p_1 + 2p_2 + 3p_3} \\
\pi(2|A) &= \frac{p_2 + p_3}{p_1 + 2p_2 + 3p_3} \\
\pi(3|A) &= \frac{p_3}{p_1 + 2p_2 + 3p_3}
\end{aligned}
\tag{2}
$$

where p_1, p_2, and p_3 are the probabilities that an *A*-packet is transmitted exactly one, two, three times, respectively. By solving the above system of equations (2), the average number of transmissions of A-packets can be derived as $p_1 + 2p_2 + 3p_3$. Analogous computations can be done for a system with different parameters.

Finally, when an A-packet reaches its rth transmission and is still in error, it is dropped. The probability of dropping an A-packet, denoted as $P_{\mathrm{drop,\,A}}$, is computed as the sum of the steady-state probabilities of the HARQ chain states with a pending packets configuration corresponding to an A-packet at its last transmission and with the channel state equal to 1 during this transmission. Thus,

$$P_{\mathrm{drop,\,A}} = \sum_{i=0}^{N-1} \chi(\sigma_{i,0} = r \ \ \& \ \ b_{i,0} = 1)\pi_i \qquad (3)$$

with $\chi(\cdot)$ being an indicator function valued 1 if the inner condition is true and 0 otherwise; $\sigma_{i,0}$ and $b_{i,0}$ are the head entries of $\boldsymbol{\sigma}$ and $\boldsymbol{b}$, respectively, in the ith HARQ chain's state.

V. Numerical results

We consider a system with exemplifying parameters. Our aim is to evaluate a setup sufficiently complex to give a detailed description of the HARQ operations and performance, but, at the same time, tractable enough to be solved easily, without involving high computational power. This setup can be modified to different cases of interest. In particular, it can be tuned to mimic real traffic traces; as illustrated in [2], using real traffic traces for the evaluations only involves a careful choice of the parameters' values. In light of this, we analyze a system where the number of B-packets incrementally encoded from an A-packet is $\ell = 4$, the maximum number of transmissions for A-packets is $r = 3$, the packet group length is $\ell + 1 = 5$ packets, the round-trip time is equal to $m = M(\ell + 1) + 1 = 6$ time slots, and $M = 1$. It follows that the number of pending packets configurations is $L = 28$, as illustrated in Fig. I, and that the number of HARQ chain states is $N = 28 \cdot 2^5 \cdot 2 = 1792$.

As a first set of results, we show how metrics vary depending on the steady-state error probability ε for different values of the burstiness B; then, similarly, we show how metrics vary depending on B for many values of ε.

We consider ε varying from 0.01 to 0.4. As for the burstiness B, i.e., the average number of consecutive erroneous time slots, we consider the values $5, 10, 30$, and also we consider the special case where $B = \frac{1}{1-\varepsilon}$, which is referred to as the *i.i.d.* case, since in this case the channel is just a binary memoryless symmetric channel with identical error probability ε regardless of the state of the system. The aim is to highlight the differences between performance in case of error bursts with length comparable with the length of a packet group ($B = 5, 10$), in case of error bursts significantly longer than packet groups ($B = 30$), and in the i.i.d. case, i.e., the case of independent and identically distributed errors.

In Fig. 3, each curve represents the throughput for variable ε and fixed B. We can notice that the throughput decreases as ε increases. Indeed, the increase in packet error probability causes the reduction of packets correctly delivered to the receiver and, correspondingly, a throughput decrease. It is worth observing, temporarily neglecting the i.i.d. case, that for fixed ε, the throughput increases as B increases. In fact, the channel correlation causes the temporal axis to be roughly subdivided into blocks, each of which can be considered correct or incorrect. Erroneous blocks, i.e., the ones where the channel is bad, do not bring any contribution to the throughput; correct blocks, i.e., the ones where the channel is good, determine the successful deliver of a sequence of packets, giving a relevant contribution. In the i.i.d. case, it results that the throughput is higher than that obtained with a correlated channel for small values of ε. If ε increases, the performance over correlated channels becomes better than for the i.i.d. channel. In Fig. 4, we underline these last considerations by reporting the ratio between the actual throughput and the value $1 - \varepsilon$, which is the throughput in the i.i.d. case without distinction between A- and B-packets and without limits on transmissions. We conclude that the channel correlation does not necessarily imply a degradation of the system performance. Conversely, the throughput is higher than the i.i.d. case if the channel error probability at steady-state is not too small.

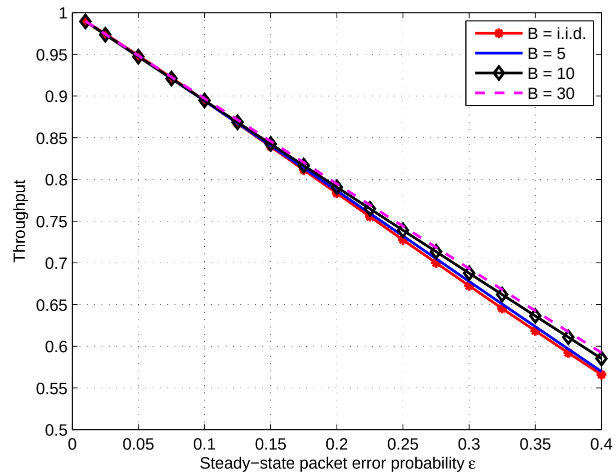

Fig. 3. Throughput *vs.* ε for various values of B.

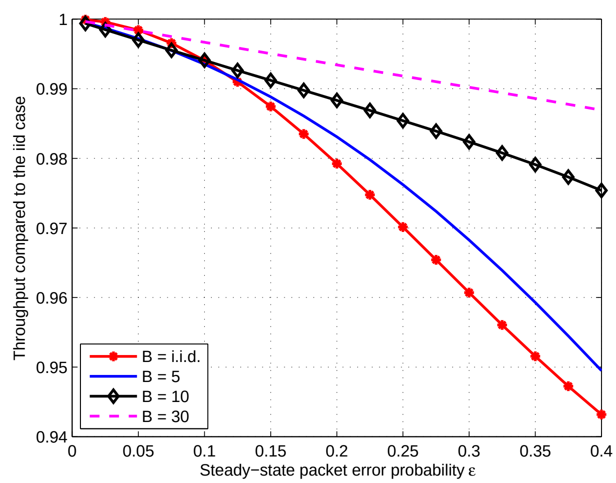

Fig. 4. Throughput compared to the i.i.d. case *vs.* ε for various values of B.

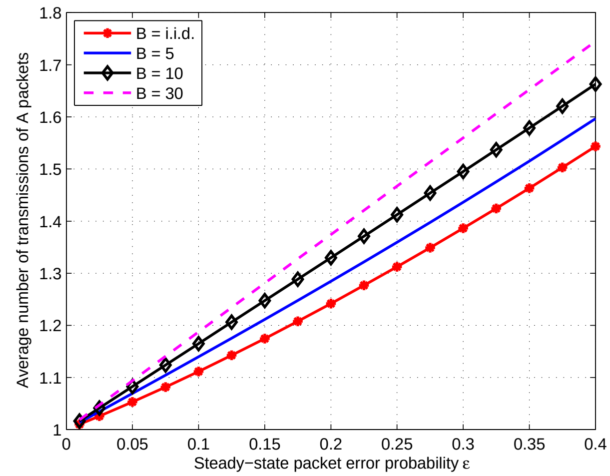

Fig. 5. Average no. of A-packet transmissions *vs.* ε for various values of B.

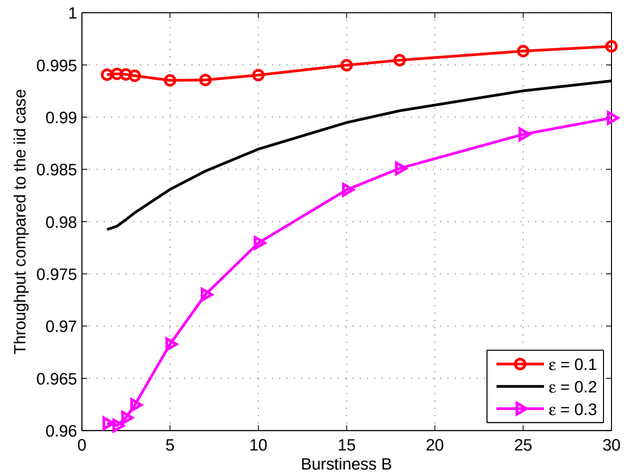

Fig. 7. Throughput compared to the i.i.d. case *vs.* B for various values of ε.

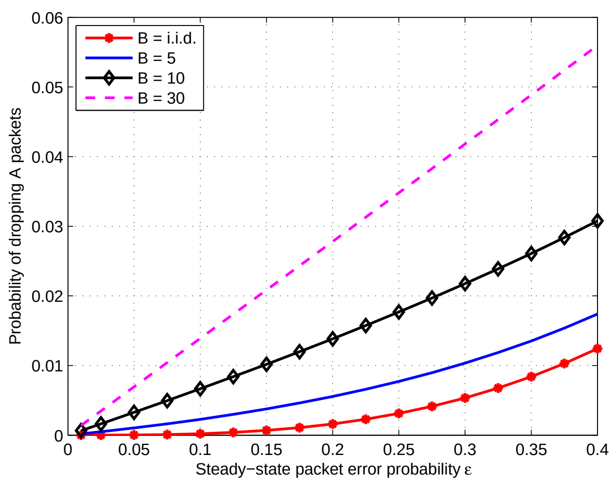

Fig. 6. Probability of dropping an A-packet *vs.* ε for various values of B.

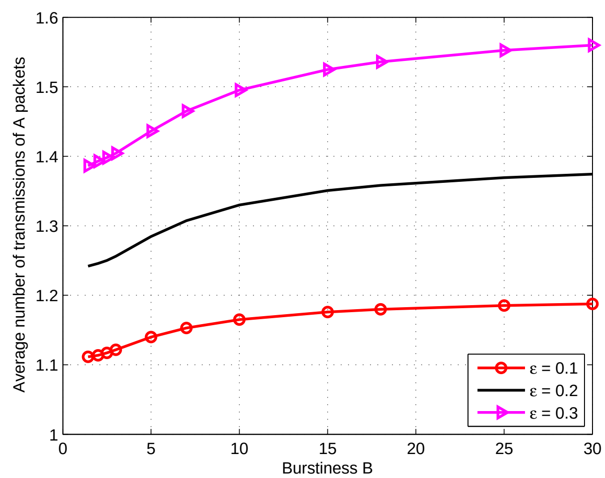

Fig. 8. Average no. of A-packet transmissions *vs.* B for various values of ε.

In Fig. 5 we show the average number of transmissions of an A-packet for varying ε and fixed B. According to the fact that a higher error probability implies a higher number of failures in A-packet transmissions, we observe that the average number of transmissions increases as ε increases, following a coarsely linear trend. Moreover, for fixed ε, the average number of transmissions is higher for higher burstiness. Indeed, if the channel is correlated and an A-packet is erroneously received, the greater the correlation, the higher the probability that the following transmissions are erroneous.

In Fig. 6 we illustrate the probability of dropping an A-packet. It can be observed that this probability increases as ε increases, for each value of B. Indeed, the increase in ε implies an increase in the average number of transmissions and in the probability that an A-packet is erroneously received at its last transmission. Furthermore, for values of B nearly equal to the packet-group length (i.i.d. errors, or $B = 5$, or $B = 10$) the dropping probability increases slowly as ε increases for small values of ε, whereas it increases faster for higher values of ε. For values of B higher than the packet-group length ($B = 30$), the dropping probability increases roughly linearly as ε increases. Finally, we can observe that the dropping probability is very small in any case. This fact agrees with the previous results. Indeed, the average number of transmissions is less than 1.8 (see Fig. 5) and the maximum number of transmissions is 3. Thus, the third transmission of an A-packet and its failure are unlikely events.

We now discuss the numerical results obtained by varying B in the range $[\frac{1}{1-\varepsilon}, 30]$ for a fixed ε value chosen within set $\{0.1, 0.2, 0.3\}$. In Fig. 7 we report the ratio of the actual throughput and that obtained in the i.i.d. case without distinction between A- and B-packets and without limits on transmissions. The variability interval of the throughput increases as ε increases. In fact, the curve for $\varepsilon = 0.1$ remains nearly constant. More precisely, the throughput initially decreases, it assumes its minimum values for B values comparable with the packet-group length, and then it increases very slowly. The curves for higher values of ε show that in these cases

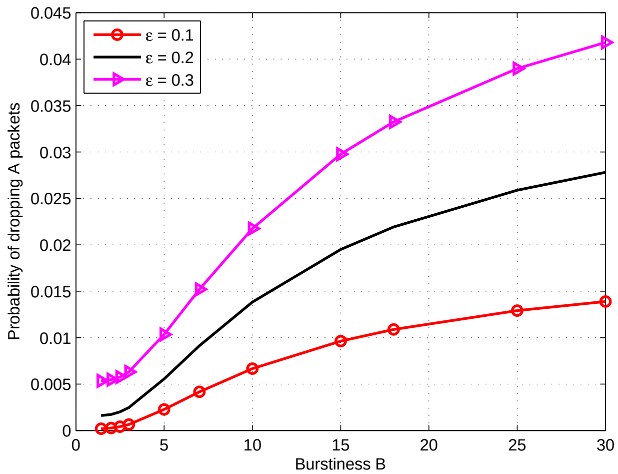

Fig. 9. Probability of dropping an A-packet *vs.* B for various values of ε.

the throughput assumes its smallest values for B roughly corresponding to the i.i.d. case and that it then increases as B increases. This increase is faster for B values lower than, say, 20, and is lighter for higher values. In any case, the throughput seems to tend to an asymptotic value.

Finally, in Fig. 8 and Fig. 9 we report the average number of transmissions and the dropping probability of an A-packet. These figures confirm the previous results. Moreover, it can be observed that the range of values assumed by either performance metric is larger for higher ε. From Fig. 8 we can also deduce that the average number of transmissions becomes almost constant for high values of B (say, higher than 10, twice the length of a packet group). The dropping probability of an A-packet in Fig. 9 is roughly constant for values of B around the i.i.d. case; it quickly increases until B reaches value 10 and it increases more slowly for higher values of B.

VI. Conclusions

We presented a performance evaluation of a hybrid SR ARQ scheme with unequal error protection applied to the transmission of multimedia contents over the wireless channel. We considered a video flow subdivided into independent and incrementally encoded packets. We assumed that the HARQ scheme privileges the transmission of independent packets, due to their more relevant role. Thus, independent packets are the only ones to be selectively retransmitted, if in error.

To avoid delay increases, we assumed that retransmissions replace transmissions of incrementally encoded packets and the number of retransmissions is limited. We modeled the entire system through a macroscopic Markov chain, including the representation of the channel and a finite-state machine identifying the possible pending packets configurations.

As performance metrics, we evaluated throughput, average number of transmissions, and dropping probability of independent packets. We reported and commented the results for a system with fixed parameters. We showed that channel correlation may be beneficial to increase throughput. We verified, by contrast, that correlation has a worsening effect on the average number of transmissions and the dropping probability, since these quantities increase with a higher channel burstiness. In a video transmission context, retransmissions lead to undesirable delays and dropped independent packets cause an irreversible loss of information, which, in turn, compromises the video quality. Thus, a tradeoff emerges between transmission speed and reproduction quality. These results can be exploited as practical guidelines; the adopted approach can be extended to better describe the performance of more specific video transmission protocols operating on possibly correlated channels.

References

[1] *Cisco White Paper*. Available online at www.cisco.com.

[2] L. Badia, N. Baldo, M. Levorato, M. Zorzi, “A Markov framework for error control techniques based on selective retransmission in video transmission over wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 488-500, 2010.

[3] S. Lin, D. J. Costello, M. J. Miller, ”Automatic-Repeat-reQuest error control schemes,” IEEE Commun. Mag., vol. 22, no. 12, pp. 5-17, 1984.

[4] H. O. Burton, D. Sullivan,“Errors and error control,” *Proc. of IEEE*, vol. 60, no. 11, pp. 1293-1301, 1972.

[5] K. Takahata, N. Uchida, N. Y. Shibata, “QoS control for real time video stream over hybrid network by wired and wireless LANs, ” *Proc. of the IEEE AINA*, pp. 45-51, 2003.

[6] T. Sikora, “MPEG digital video-coding standards,” *IEEE Sign. Proc. Magazine*, vol. 14, no. 5, pp. 82-100, 1997.

[7] L. Badia, M. Levorato, M. Zorzi, “Analysis of selective repeat retransmission techniques for differentially encoded data,” *Proc. of the IEEE ICC*, pp. 1-6, 2009.

[8] J. B. Seo, Y. S. Choi, S. Q. Lee, N. H. Park, and H. W. Lee, “Performance analysis of a type-II Hybrid-ARQ in a TDMA system with correlated arrival over a non-stationary channel,” *Proc. ISWCS*, pp. 59-63, 2005.

[9] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012-1030, 2000.

[10] I. Ahmed, L. Badia, D. Munaretto, and M. Zorzi, “Analysis of PHY/application cross-layer optimization for scalable video transmission in cellular networks,” *Proc. IEEE WoWMoM*, 2013.

[11] L. Badia, A. V. Guglielmi, “A Markov analysis of automatic repeat request for video traffic transmission,” *Proc. IEEE WoWMoM*, 2014.

[12] C. C. Li, S. C. S. Chen, “Providing unequal reliability for transmitting layered video streams over wireless networks by multi-ARQ schemes,” *ICIP*, vol.3, pp. 100-104, 1999.

[13] H. C. Wei, Y. C. Tsai, C. W. Lin, “Prioritized retransmission for error protection of video streaming over WLANs,” *Proc. of the 2004 ISCAS*, vol.2, 2004.

[14] P. Pérez and N. Garcia, “Lightweight multimedia packet prioritization model for unequal error protection,” *IEEE Trans. on Consumer Electronics*, vol. 57, no. 1, pp. 132-138, 2011.

[15] H. Lee, T. Jung, K. Seo, and C.K. Kim, “Delay constrained ARQ mechanism for MPEG media transport protocol based video streaming over Internet,” *Proc. AFIN*, 2015.

[16] H. Ha, J. Park, S. Lee, and A. C. Bovik, “Perceptually unequal packet loss protection by weighting saliency and error propagation,” *IEEE Trans. Circ. Sys. Video Tech.*, vol. 20, no. 9, pp. 1187-1199, 2010.

[17] C. H. Lin, Y. C. Wang, C. K. Shieh, and W. S. Hwang, “An unequal error protection mechanism for video streaming over IEEE 802.11e WLANs,” *Computer Networks*, vol. 56, no. 11, pp. 2590-2599, 2012.

[18] J. Lu, C. K. Wu, S. Xiao, and J. C. Du, “A network coding based hybrid ARQ algorithm for wireless video broadcast,” *Science China Information Sciences*, vol. 54, no. 6, pp. 1327-1332, 2011.

[19] M.-F. Tsai, T.-C. Huang, C.-H. Ke, C. K. Shieh, and W. S. Hwang, “Adaptive hybrid error correction model for video streaming over wireless networks,” *Multimedia Systems*, vol. 17, no. 4, pp. 327-340, 2011.

[20] L. Badia, “On the effect of feedback errors in Markov models for SR ARQ packet delay,” *Proc. IEEE Globecom*, 2009.

[21] M. Zorzi and R. R. Rao, “Lateness probability of a retransmission scheme for error control on a two-state Markov channel,” *IEEE Trans. Comm.*, vol. 47, no. 10, pp. 1537-1548, 1999.

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