Cross-Layer Error Control for Multimedia Streaming in Wireless/Wireline Packet Networks

Antonios Argyriou, *Member, IEEE*

***Abstract*—In this paper, we propose a cross-layer error control framework for robust and low delay multimedia streaming in tandem-connected IEEE 802.11 wireless LANs and the Internet. For this network configuration, we model the end-to-end delay and packet loss rate as a function of the automatic repeat request (ARQ) and forward error correction (FEC) error control mechanisms that are employed at the application and wireless link layers. The analytical model is used as the basis of a delay-constrained error control algorithm that adapts the protection level at the application and link layers so that the end-to-end packet loss rate is minimized. With extensive simulations, we validate the efficiency of the proposed cross-layer error control methodology for delay-sensitive pre-compressed video streaming.**

***Index Terms*—802.11 wireless LAN, ARQ, cross-layer design, FEC, hybrid wireless/wired network, wireless multimedia streaming.**

I. Introduction

ONE important characteristic of compressed multimedia streams is their vulnerability to errors. In wireless local area networks (WLANs), this problem is very difficult to address since the unreliability and time-varying nature of the wireless channel, user mobility, and contention at the medium access control (MAC) layer can cause both bit errors and packet erasures. Even with QoS enhancements to the wireless LAN protocols like the IEEE 802.11e [1], packet losses that are caused by wireless errors are unavoidable, and they can only be reduced by error control mechanisms that are employed at different layers of the protocol stack [2]. The most well-investigated techniques that deal with errors in multimedia streaming applications include error resilient source coding, error concealment, and channel coding with forward error correction (FEC) and automatic repeat request (ARQ). In this paper, we are interested in transport-level mechanisms and so we focus our attention in the later two.

The least complex of the two techniques is ARQ that is usually employed either at the application or link layers. The disadvantages of ARQ are the requirement of a reverse channel and the variable network delay. However, it can offer extremely low error rates for a given amount of redundancy. The performance of application-layer ARQ for video streaming applications has been studied in a wide range of channel conditions [3], [4]. For real-time video communications delay constraints are very strict, making thus the retransmission of lost packets not particularly useful in a practical setup. To overcome this problem, FEC is usually employed [5]–[8]. On the other hand, the disadvantage of FEC systems is that they require the transmission of a large amount of redundant data to overcome poor channel conditions. However, they can offer a fixed network delay. An interesting analysis that presents possible tradeoffs between the degree of reliability and the resulting delay for wireless video transmission can be found in [9]. In most practical systems, FEC is usually employed with an interleaver at the source encoder so that a burst loss is spread over many FEC codewords. The disadvantage is that the interleaver introduces additional delay that must be taken into account by the streaming application [10].

To combine the best features of the two schemes, several works have suggested the use of hybrid approaches that use either application-layer ARQ or FEC depending on the channel conditions [11], [12]. For pre-compressed video streaming, a generalized rate-distortion framework for studying different streaming and error control scenarios at the application layer was presented in [11]. Works that follow similar principles can be found in [13], [14]. For real-time video encoding and streaming, the same hybrid ARQ/FEC approach was also shown to be very effective [12]. There has also been an interest in using cross-layer information from the protocol stack in order to perform error control more efficiently. For example in [15], the authors use information from the application layer so that the number of link-layer retransmissions is adapted depending on the importance of each packet. An analytical cross-layer method that considers application-layer FEC and wireless link-layer ARQ, but only for the protection of scalable encoded video, is thoroughly described in [16].

While in the previous works the various flavors of ARQ and FEC systems have been demonstrated as an effective way to combat packet erasures, thorough performance analyses of the interactions between the error control techniques that are employed at different layers of the protocol stack are limited. In this paper, we develop a new cross-layer performance model that considers the effect of wireless link-layer ARQ, application-layer ARQ, and application-layer FEC, on the average packet loss rate and delay of transmitted media packets. This model is developed for a channel that considers a tandem connection between a wireless LAN and a wireline IP-based backbone. The analytical expressions that we derive, are used for real-time adaptation of the basic error control schemes that are employed at the application and wireless link layers. Our goal is to demonstrate that the proposed scheme can contribute to more efficient adaptation by avoiding duplicate effort across layers.

Manuscript received March 19, 2007; revised March 21, 2008. Current version published October 24, 2008. This work was presented in part at the International Workshop on Multimedia Signal Processing (MMSP), Chania, Greece, October 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Qian Zhang.

The author is with Philips Research, Eindhoven 5656 AE, The Netherlands (e-mail: anargyr@ieee.org).

Digital Object Identifier 10.1109/TMM.2008.2001371

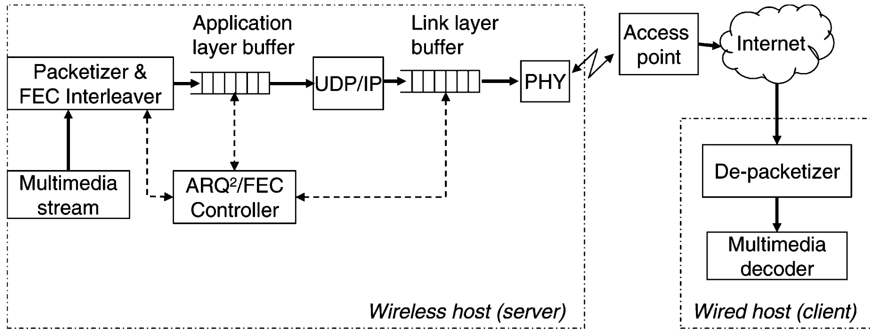

Fig. 1. Proposed wireless multimedia streaming system with cross-layer error control. Dashed lines indicate cross-layer signaling while solid lines indicate data flow.

This paper is organized as follows. An overview of the proposed system is presented in Section II. Based on the adopted system model, we systematically derive the end-to-end delay in Section III. In Section IV we present the performance model that characterizes the overall end-to-end packet loss rate. Based on our previous analysis, in Section V we define the problem of cross-layer error control and we provide a simple solution algorithm. In Section VI we present simulation results that stress-test the validity of the model and the actual performance of the proposed algorithm for video streaming experiments. Finally we present our conclusions in Section VII.

II. System Overview

Fig. 1 depicts the main components of the proposed system architecture for wireless multimedia streaming with cross-layer error control. The media source, which can be a real-time encoder or a pre-compressed media file, generates media packets that are initially sent to the application-layer FEC encoder. In this work, we implement FEC by considering systematic Reed-Solomon (RS) codes, even though the proposed modeling framework could be easily extended to work with other codes. Fig. 2 depicts how FEC is applied for a group of media packets that are transmitted with our streaming system. FEC is applied across the source packets so that each generated transport packet contains parts both of the source payload and of the parity bits. The RS(n,k) encoder generates $n-k$ additional packets for k input source packets. If π is the packet erasure rate before error recovery, the probability of RS decoding failure is equal to:

$$P^{\mathrm{fec}} = 1 - \sum_{i=0}^{n-k} \binom{n}{i} \pi^i (1-\pi)^{n-i}. \tag{1}$$

For a source with a data rate of R_s packets/s, the resulting FEC bitrate is $R_s(n-k)/k$ packets/s. FEC will also introduce an additional delay since k packets must be buffered before RS encoding. Furthermore, in case of a single packet loss, n transport packets must be buffered before RS decoding takes place. To avoid additional delays in the playback process at the client, RS encoding is applied on k source packets that will have the same playback deadline at the media decoder. After the RS encoding is over, the n transport packets are placed in the application buffer, while a copy of each packet is also stored in the retransmission buffer. Each packet is assigned a maximum

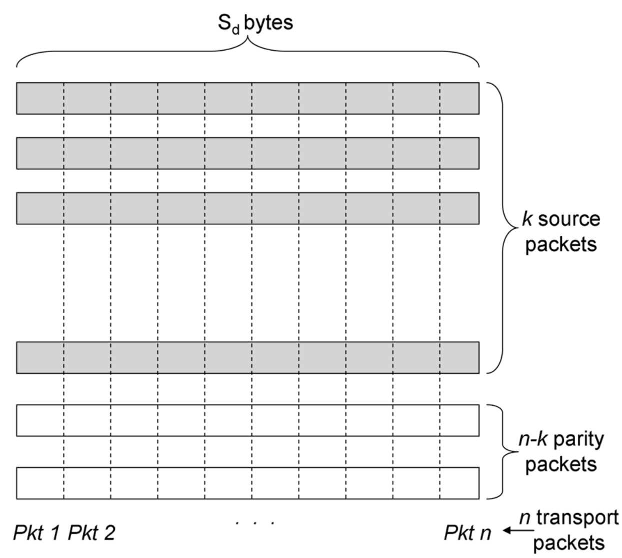

Fig. 2. Packetization with forward error correction across k source packets. Each one of the n generated transport packets has an associated maximum number of application and link-layer retransmissions.

number of application-layer retransmissions denoted as N_a^m. Negative acknowledgments (NACKs) are also used at the application layer for reporting gaps in the sequence numbers of the received media packets. When a retransmission request arrives through a NACK, the requested packet that is stored in the retransmission buffer has priority over the regular application buffer, and is transmitted as soon as the network allows it. Subsequently, a UDP/IP header is added by the network layer. The IP packet is sent next to the 802.11 MAC layer which appends a header of 28 bytes, and creates an MAC protocol data unit (MPDU) for wireless transmission. Each MPDU can be retransmitted up to N_w^m times by the link layer. Once the packets are correctly transmitted at the wireless physical layer, they are forwarded by the access point in the wireline IP network.

A. Channel Model

To capture the error process in the end-to-end path that is illustrated in Fig. 1, we adopt a joint model between the wireless and the wireline channels. In this paper, we model each of the two channels with a two-state Markov chain, since it has been demonstrated that this is a good approximation of the error process both in wireless fading channels [17] and the Internet

[5]. The behavior of a two-state Markov chain, is characterized by good and bad states, with the bad state denoting a failed packet transmission. In the case of wireless networks, the bad state represents usually bad conditions of the wireless medium, while in the Internet case it corresponds to congestion. The parameters that define the behavior of the Markov chain are the transition probabilities between good and bad states that we denote as P_{gb} and P_{bg}. The stationary probabilities of being in the good and bad states are given by the terms $P_{bg}/(P_{gb}+P_{bg})$ and $P_{gb}/(P_{gb}+P_{bg})$ respectively. The last expression actually provides the packet loss rate that is denoted from now on as p_w and p_o for the wireless and wireline networks respectively. We must note here that we consider MPDU erasures in the wireless network, since in practice if errors are detected by the link layer check sequence the complete MPDU is discarded [16].

Based on the channel model that we just described, we can express the total end-to-end of packet erasure rate as follows:

$$\pi = 1 - [1 - p_w][1 - p_o]. \tag{2}$$

III. Delay Analysis

Deriving an analytical expression for the end-to-end packet erasure rate is not enough for characterizing the performance of our system since ARQ is employed both at the application and link layers. Therefore, it is important to model the effect of retransmissions on the delay and eventually the overall error rate experienced by the media streaming application. If we consider all the delay components that affect the transmission of a single packet, the one-way end-to-end delay for a video packet l can be written as

$$L_l = L_{\text{enc/dec}} + L_{\text{fec}} + L_{\text{pkt}} + L_w + L_N. \tag{3}$$

$L_{\text{enc/dec}}$ is the encoding and decoding delay, while L_{pkt} is the packetization and de-packetization delay. These two delay components are generally constant and can be ignored [12]. The FEC encoding delay is constant for given values of n and k. On the other hand, the delay on the wireless link and the Internet are random variables and are denoted as L_w and L_N. In this section, we will calculate analytically the last two parameters since they are the dominant factors in (3).

A. Delay in the Wireline IP Network

In this work, we model the random variable of the one-way Internet delay L_N with a Gamma distribution. This distribution can properly account for the main reason of Internet delays, that is due to buffering in the routing infrastructure or other network elements like proxies and gateways [18]. Therefore, the Gamma probability distribution function with rightward shift γ and parameters ν and α is given as follows:

$$f_{L_N}(t) = \frac{\alpha}{\Gamma(\nu)}(a(t-\gamma))^{(\nu-1)}e^{(-\alpha(t-\gamma))} \text{ for } t > \gamma. \tag{4}$$

From the network perspective, ν expresses the number of routers, γ is the total end-to-end processing time, and α is the parameter of exponentially distributed waiting time in each router that is modeled as an M/M/1 queue. Therefore, this model considers that packets traverse a network of ν M/M/1 queues, where each of them has a mean processing time plus waiting time equal to $(\gamma/\nu)+(1/\alpha)$ and variance $(1/\alpha^2)$. Furthermore, the forward trip time has a mean equal to $\mu_F = \gamma + (\nu/\alpha)$, and variance equal to $\sigma_F = (\nu/\alpha^2)$. Both α and ν are calculated by periodically estimating the mean and variance of the forward and backward trip time at the receiver, and transmitting their values to the sender which uses the following equations [11]:

$$\alpha = (\mu_F - \sigma_F)/\sigma_F^2 \tag{5}$$

$$\nu = (\mu_F - \sigma_F)/\alpha. \tag{6}$$

However, the aforementioned approach is used for modeling and estimating the one way delay, which means that the delay added from retransmissions has to be recalculated when packet erasures take place. Therefore, the overall delay experienced by a single packet, that includes subsequent retransmissions and immediate negative acknowledgments, will be characterized by the joint distribution of the forward and backward trip delays. Furthermore, according to the adopted network delay model, the delay experienced by a lost packet is correlated to the delay of the last successfully sent packet since packets are sent back-to-back [11]. Therefore, for i application-layer retransmissions the probability that a packet l was late can be obtained if we consider the joint distribution of the total forward and backward trip time:

$$P_r\left\{L_{l,N}^{(i)} > \tau_l\right\} = \int_{\tau_l}^{\infty} f_F^{(1)} * \cdots f_F^{(i+1)} * f_B^{(1)} * \cdots f_B^{(i)} dt. \tag{7}$$

In this equation τ_l is the maximum allowed delay for packet l, while f_F and f_B are the probability distribution functions of the forward and backward trip delays respectively. Note that the above equation corresponds only in the delay experienced over the Internet path.

B. Delay Analysis for the 802.11 WLAN

The next step in our analysis is the derivation of the transmission delay for every 802.11 MPDU in the wireless link when the point coordination function (PCF) is used. With the distributed coordination function (DCF) in IEEE 802.11, the delay that each packet experiences depends on the number of contending users at the MAC layer. In this paper, we limit our analysis to the performance of a single user where packet losses are caused by wireless link errors since the PCF mode offers contention-free channel access. Regarding the detailed operation of the IEEE 802.11 MAC, the interested reader is referred to [19], [20]. In this section we will analyze the MAC protocol actions during successful and unsuccessful transmissions of an MPDU since they have different impact on the overall delay.

We denote the transmission delays experienced by a single MPDU during a loss and no-loss event as L_1 and L_2 respectively. Also, S_l is the size of the source packet l, S_o the combined header overhead, and R_m the data rate for the selected 802.11a PHY transmission mode with $m \in 1, ..., 8$. With PCF, when the wireless station transmits a data MPDU, it receives an acknowledgment after a duration of a short inter-frame space (SIFS) [19]. When the 802.11 ACK frame is received, the sender transmits a new data MPDU after another period equal to SIFS. Therefore, the transmission delay is simply equal to

$$L_2 = \mathrm{SIFS} + \frac{S_l + S_o}{R_m} + \frac{S_{\mathrm{ack}}}{R_1}. \tag{8}$$

An MPDU transmission can fail either with the loss of the data or the acknowledgment. When a data MPDU transmission fails, the sender does not receive an ACK after a duration of SIFS, and it retransmits the data MPDU after a duration equal to a PCF inter-frame space (PIFS) [19] (note that PIFS>SIFS). This sequence of actions means that the transmission delay in this case will be

$$L_{1a} = 2\frac{S_l + S_o}{R_m} + \mathrm{PIFS} + \mathrm{SIFS} + \frac{S_{\mathrm{ack}}}{R_1}. \tag{9}$$

On the other hand, when the ACK MPDU transmission fails, the sender is able to sense the failed ACK transmission (busy medium). When this transmission ends and the medium is free, the sender waits for SIFS and retransmits the data packet. The transmission delay in this case will be equal to

$$L_{1b} = 2\frac{S_l + S_o}{R_m} + 3\mathrm{SIFS} + 2\frac{S_{\mathrm{ack}}}{R_1}. \tag{10}$$

The combined packet loss probability for these two cases is given by $p_w = p_{1,w}p_{2,w}$. The packet loss probabilities due to wireless link erasures $p_{1,w}$ and $p_{2,w}$ can be obtained from the underlying MAC layer which maintains this information. The average delay caused by any of these types of loss events can be written as $L_{w1} = p_{1,w}L_{1a} + p_{2,w}L_{1b}$.

Now the probability that i retransmissions are needed for successful transmission of an MPDU, out of the N_w^m allowed is given by

$$\pi_w = \frac{(1-p_w)p_w^i}{1-p_w^{N_w^m+1}}. \tag{11}$$

Also, given N_w^m and the wireless packet erasure rate p_w, we can calculate the average number of link-layer retransmissions as follows:

$$\bar{N}_w = \frac{1-p_w^{N_w^m+1}}{1-p_w} - 1. \tag{12}$$

For application-layer ARQ, π_a and $\bar{N}_a$ will have the same form because the same assumptions hold for the underlying erasure channel. If we add the MPDU transmission delay for the case where there was a delivery failure even after the N_w^m-th retransmission, we obtain the average transmission delay for N_w^m-truncated link-layer ARQ as follows:

$$L_w = \sum_{i=0}^{N_w^m} \pi_w(iL_1 + L_2) + (N_w^m + 1)\,L_2. \tag{13}$$

Note that this expression is the average value of the delay in the wireless link which means that the sender has to re-calculate it by measuring variations in the included parameters.

C. End-To-End Delay

To calculate the overall end-to-end delay experienced by a single packet, we have to consider the effect of all the error control mechanisms in use. First, we calculate the probability that a transport packet m_l will miss the playback deadline of the corresponding source packet l, after i application-layer retransmissions:

$$P_{r,m_l}^{(i)} = P_r\left\{L_N^{(i)} + iL_w > \tau_{m_l}\right\} = P_r\left\{L_N^{(i)} > \tau'_{m_l}\right\}. \tag{14}$$

Since τ'_{m_l} is constant for given i, the probability distribution function of the previous equation is a time-shifted version of (7). Nevertheless, we have to calculate the probability that i application-layer retransmissions are needed for the correct transmission of a single transport packet. In the same way we derived this expression for the link-layer retransmissions, we can write

$$\pi_a = \frac{(1-\pi)\pi^i}{1-\pi^{N_a^m+1}}. \tag{15}$$

So the probability that the transport packet m_l will miss the playback deadline, for any possible number of retransmissions, will be

$$P_{r,m_l}\{L_{m_l} > \tau_l\} = \sum_{i=0}^{N_a^m} \pi_a P_{r,m_l}^{(i)} \tag{16}$$

However, because of FEC, for k source packets n transport packets are transmitted by the application. If more than $n-k$ transport packets arrive late for the playback deadline then RS decoding will fail. Therefore, the probability that the source packet will miss its playback deadline will be

$$P_r\{L_l > \tau_l\} = \sum_{j=n-k+1}^{n} \binom{n}{j} \prod_{m_l=1}^{j} P_{r,m_l}\{L_{m_l} > \tau_l\} \tag{17}$$

since any combination of more than $n-k$ lost packets will result in an RS decoding failure and of course a missed playback deadline. This equation reveals an interesting case where we might have an RS decoding failure not because of a packet erasure, but because of a late packet arrival.

IV. End-to-End Packet Loss Rate

Recall that p_w and p_o are the packet erasure rates of the wireless and wired links respectively without considering retransmissions. Also, N_a^m and N_w^m are the maximum allowed retransmission attempts at the application and link layers respectively. With the adopted channel model, the packet loss rate for N_w^m-truncated ARQ will be equal to $p_w^{N_w^m+1}$. After application-layer ARQ is also applied, the total end-to-end packet loss rate will be equal to $\pi^{N_a^m+1}$. With both application and link-layer ARQ, (2) will become

$$P_{e2e}^{\mathrm{arq}^2} = \pi^{N_a^m+1} = \left(1 - \left[1 - p_w^{N_w^m+1}\right]\left[1-p_o\right]\right)^{N_a^m+1}. \tag{18}$$

Therefore, with FEC, application and wireless link-layer ARQ in the hybrid end-to-end path, after substituting (18) in (1), we obtain the overall end-to-end packet erasure rate denoted as $P_{e2e}^{\text{arq}^2\_\text{fec}}$.

If the maximum allowed delay for packet l is τ_l, the overall packet loss rate that takes into account late packets can be obtained from (17). The final formula for the end-to-end effective packet loss rate will be

$$\epsilon_l = P_{e2e}^{\text{arq}^2\_\text{fec}} + \left(1 - P_{e2e}^{\text{arq}^2\_\text{fec}}\right) P_r\{L_l > \tau_l\}. \qquad (19)$$

V. Optimal Cross-Layer Error Control

The analytical expressions derived in the previous section are used by a simple cross-layer adaptive error control algorithm. The objective of this algorithm is to minimize the effective packet loss rate given the maximum allowed end-to-end delay τ_l and available channel rate R_T. The parameters that the algorithm can adjust during this process, are the retransmission retry limits N_w^m and N_a^m, and the RS code for a group of k source packets. The objective can be formally written as follows:

$$\begin{aligned} &(N_w^m, N_a^m, n/k)^* = \arg\min(\epsilon_l) \\ \text{s.t.}\quad & L_l \leq \tau_l \text{ and } R_s + R_{\text{arq}} + R_{\text{fec}} \leq R_T. \end{aligned} \qquad (20)$$

R_{arq} and R_{fec} denote the channel rate dedicated to application/link-layer ARQ and FEC respectively, while R_T denotes the available channel bandwidth. Note from (19) that both the end-to-end delay and ϵ_l can be increased as the selected protection strategy becomes stronger. The proposed model is used precisely for fast online calculation of the optimal error control combination. It is also possible that the multimedia streaming application configures specific values for both the maximum τ_l and ϵ_l. Finally, the network parameters that are used by the models are estimated both at the sender and the receiver. The receiver is mainly responsible for measuring the mean and variance of the overall end-to-end delay and also π. The sender measures p_w and calculates the wireless link-layer delay L_w according to (13) and the wireline packet erasure rate p_o according to (2).

VI. Performance Evaluation

Our initial goal is to evaluate the accuracy of the overall packet loss rate model. Furthermore, we use the developed model for exercising cross-layer error control in a real video streaming experiment. For the two aforementioned experiments, we now describe certain aspects of the evaluation setup that are similar. We used the ns-2 network simulator to simulate an 802.11a WLAN connected in tandem with a wireline IP network. The end-to-end connection includes the media server which is located in the wireless network, and the receiver that is located in the Internet. The capacity of the wireless link is set to 12 Mbps, that corresponds to a QPSK modulation scheme. The IEEE 802.11a MAC protocol operates in the PCF mode. The timing parameters were set as follows $\text{SIFS} = 16\ \mu\text{sec}$, $\text{PIFS} = 25\ \mu\text{sec}$. The combined header overhead (application, transport, network, MAC) is 68 bytes.

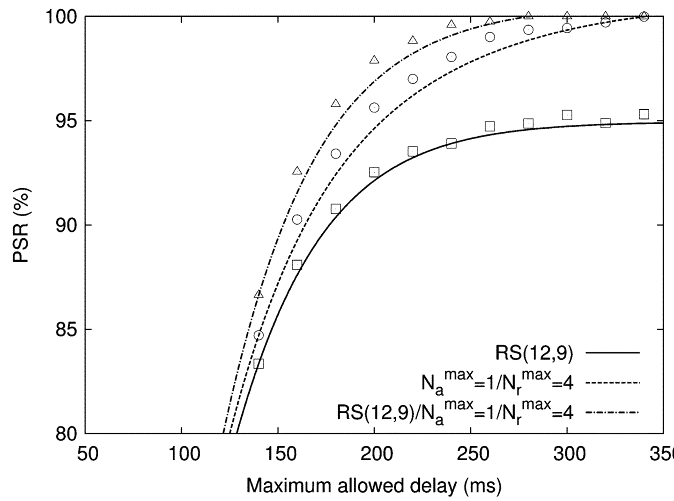

Fig. 3. Analytical and simulation results of the packet success rate (PSR) for different values of the maximum allowed end-to-end delay. $R_s = 512$ Kbps and $R_T = 3R_s$.

A. Model Accuracy

For evaluating the accuracy of the end-to-end packet loss rate model, we tested three statically configured error control combinations. For this set of simulations, we set the average end-to-end delay and bandwidth of the wireline IP channel at 100 ms and 10 Mbps respectively. All the results for this group of simulations were averaged for 100 runs.

The cumulative distribution function of the number of delivered packets (i.e., packet success rate $\text{PSR} = (1-\epsilon_l)\%$), can be seen in Fig. 3(a). We present analytical and simulation results for $p_w = 10\%$ and $p_o = 5\%$. The first important observation that can be made from this figure, is that the use of link-layer ARQ $(N_r^m = 4)$ leads to substantial performance increase since p_w dominates the overall packet loss rate. The additional application-layer ARQ/FEC mechanisms improve slightly the performance below 200 ms. After nearly 250 ms, link-layer ARQ reaches the maximum error correcting capability, and the use of an application-layer error control mechanism can improve performance. Our simulation results in this figure are characterized by the same trend with the proposed model. However, we observe that the simulation results for the stand-alone FEC scheme are closer to what the model predicts. The reason for this results is that FEC introduces a constant delay in our simulation, while with ARQ the end-to-end delay estimate is updated continuously.

B. Video Streaming Experiments

For the actual video streaming experiments, the wireline packet loss rate p_o is set equal to 5%. The bandwidth, and average forward and backward delays are set equal to 10 Mbps, 80 ms and 40 ms respectively. Occasionally, cross-traffic was introduced so as to create small delay fluctuations but without generating packet losses. The QCIF sequence Claire was encoded at 30 fps with the H.264/AVC JM12.2 software [21]. We used only I and P frames and we set the GOP size at 64 frames. The QP was set equal to 14 in order to get constant video quality. Furthermore, the maximum source video packet

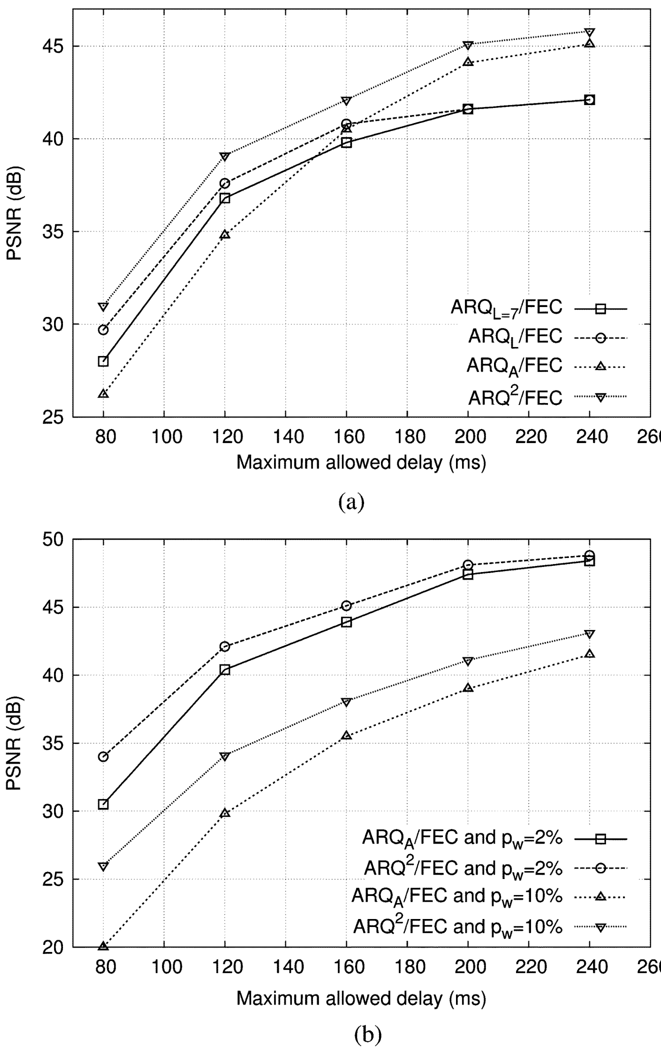

Fig. 4. Average PSNR for uplink video streaming simulations. (a) Results for all schemes with $p_o = 5\%$ and $p_w = 5\%$. (b) Results for ARQ$_A$/FEC and ARQ^2/FEC with $p_o = 5\%$.

size is set at 1000 bytes. With this frame size, the encoded P frames are packetized into three transport packets. For obtaining comparative results, we implemented four adaptive error control schemes: The hybrid ARQ$_L$/FEC scheme where FEC is applied at the application layer and ARQ is applied at the wireless link layer. The ARQ$_{L=7}$/FEC system that employs only application-layer FEC, while at the wireless link layer the retry limit is set equal to 7 since it corresponds to 802.11 real-life implementations. The ARQ$_A$/FEC scheme where both mechanisms are employed at the application layer, while the default 802.11 retry limit is used at the link layer. Finally, the proposed system where the error control schemes are employed in a joint optimization setting is denoted as ARQ^2/FEC.

In Fig. 4 we present the average quality in terms of the peak signal-to-noise ratio (PSNR) at the video decoder versus the allowed end-to-end delay for 20 simulation runs. The first striking observation about the results in Fig. 4(a) is that below a specific delay constraint ARQ$_A$/FEC performs worse than all the other schemes. The reason is that for a strict delay constraint, ARQ$_A$/FEC can use FEC while at the link layer the default retry limit that is equal to 7, results into several retransmissions that are useless due to the low delay constraint. On the other hand, for a low delay constraint the proposed ARQ^2/FEC is using extensively FEC at the expense of link-layer ARQ since the arrival of these retransmissions is expected to be late. The performance improvement suggests that it is crucial to explore not only the available tradeoffs between ARQ and FEC at the application layer, but also at the wireless link layer. However, when the allowed delay is increased the use of application-layer ARQ from both systems results in significant improvement of the video quality, since most retransmissions arrive on-time and the delay at the wireless link layer has minimal effect. On the other hand, ARQ$_L$/FEC performs very well for low delay but it under-performs above 160 ms because of the constant overhead introduced by FEC. At the same time, in the operational area above 200 ms, ARQ^2/FEC can start using application-layer retransmissions. This result demonstrates clearly that the proposed scheme is able to combine the best characteristics of the basic error control mechanisms when cross-layer signaling is allowed.

It is also interesting to note in the same figure the performance of the ARQ$_{L=7}$/FEC system, that is similar with the ARQ$_L$/FEC scheme for a delay constraint higher than 180 ms. This result should be expected because the optimal value of the retry limit for the later scheme, coincides in most cases with the hard-coded value in the 802.11 MAC. But below 180 ms, the ARQ$_L$/FEC system performs better since it defers from using aggressively link-layer retransmissions, but instead it uses the available bandwidth for stronger FEC protection.

In Fig. 4(b), we examine the performance of the ARQ$_A$/FEC and ARQ^2/FEC systems with $p_o = 5\%$ but for different values of p_w. Purpose of these simulations is to highlight quantitatively the effect of packet erasure rate variations on the system performance. The first thing that can be said is that the differences in performance between the two schemes are more significant for the higher wireless PER of 10%, since link-layer retransmissions are used more frequently. We also observed that hard-coding the link-layer ARQ at the maximum value of 7, that is also the case for the ARQ$_A$/FEC system, did not created performance problems for a channel p_w equal to 2%. However, for a higher p_w equal to 10% the number of actual retransmissions was increased considerably and it had negative impact on delay of the wireless link L_w. This means that for a low delay constraint, the higher value of L_w results in quite low packet success rate and eventually low video quality in terms of PSNR. However, with the adaptive link-layer ARQ mechanism, the retry limit is set to a lower value instead of persistently being equal to 7.

Table I presents representative results for the optimal values of all the error control parameters that were calculated by the ARQ$_A$/FEC and ARQ^2/FEC mechanisms in the previous experiments. These results demonstrate more accurately the relative decisions that were made by the tested error control schemes. Two observations can be made from the results in this table. First, we see that ARQ^2/FEC is reducing aggressively the use of link-layer retransmissions since a high retry limit introduces delay of several milliseconds contrary to FEC. Second, the proposed scheme makes increased use of link-layer retransmissions

TABLE I
CALCULATED ERROR CONTROL (EC) PARAMETERS FOR DIFFERENT CHANNEL ERASURE RATES AND ALLOWED END-TO-END DELAY τ

EC scheme	Channel erasure (p_o, p_w)	EC parameters vs. delay τ $N_w/N_a/(n,9)$			
		τ:120	τ:160	τ:200	τ:240
ARQ_A	(5%,2%)	7/0/18	7/0/18	7/0/18	7/1/10
+	(5%,5%)	7/0/18	7/0/18	7/0/18	7/1/10
FEC	(5%,10%)	7/0/18	7/0/18	7/0/18	7/1/10
ARQ^2	(5%,2%)	2/0/24	4/0/22	2/1/18	4/1/14
+	(5%,5%)	3/0/24	5/0/20	3/1/16	5/1/12
FEC	(5%,10%)	4/0/22	6/0/20	3/1/16	6/1/10

as p_w and τ are increased but it still scales back the retry limit in order to accommodate application-layer retransmissions. This last case happens for τ equal to 200 ms.

VII. CONCLUSION

In this paper, we presented a cross-layer error control framework that can contribute to improved robustness for delay-sensitive multimedia streaming in tandem-connected wireless LANs and the Internet. We developed an analytical model that expresses the end-to-end delay and packet loss rate as a function of both the application-layer ARQ/FEC and wireless link-layer ARQ error control algorithms. Subsequently, we proposed the use of this model for adaptively controlling the aforementioned error algorithms depending on the channel conditions. Our simulations showed that for minimizing the effective end-to-end packet loss rate, it is necessary to consider the delay introduced by each of the employed error control schemes. Furthermore, we demonstrated that the proposed scheme is able to calculate optimal error control configurations that cannot be identified when a subset of the available options is considered.

REFERENCES

[1] IEEE 802.11e/d13.0. Draft Supplement to Part 11: Wireless Medium Sccess Control (MAC) and Physical Layer (PHY) Specifications: MAC Enhancements for Quality of Service (QoS) Jan. 2005.

[2] M. van der Schaar and P. Chou, *Multimedia Over IP and Wireless Networks*. New York: Academic, 2007.

[3] C.-H. Wang, R.-I. Chang, J.-M. Ho, and S.-C. Hsu, "Rate-sensitive ARQ for real-time video streaming," in *Global Telecommunications Conf. (GLOBECOM)*, San Francisco, CA, Dec. 2003, vol. 6, pp. 3361–3365.

[4] H.-C. Wei, Y.-C. Tsai, and C.-W. Lin, "Prioritized retransmission for error protection of video streaming over WLANs," in *ISCAS*, 2004.

[5] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC based error control for Internet telephony," in *INFOCOM*, 1999.

[6] J. Rosenberg, L. Qiu, and H. Schulzrinne, "Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet," in *INFOCOM*, 2000.

[7] P. Frossard and O Verscheure, "Joint source/FEC rate selection for quality optimal MPEG-2 video delivery," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1815–1825, Dec. 2001.

[8] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Model-based delay-distortion optimization for video streaming using packet interleaving," in *36th Asilomar Conf. Signals, Systems, and Computers*, 2002.

[9] M.-T. Sun and A. R. Reibman, *Compressed Video Over Networks*. New York: Marcel Dekker, Sep. 2001.

[10] F. Borgonovo and A. Capone, "Efficiency of error-control schemes for real-time wireless applications on the Gilbert channel," *IEEE Trans. Veh. Technol.*, vol. 54, no. 1, pp. 246–258, Jan. 2005.

[11] P. A. Chou and Z. Miao, Rate-Distortion Optimized Streaming of Packetized Media , Microsoft Research Tech. Rep. MSR-TR-2001-35, 2001.

[12] F. Zhai, Y. Eisenberg, 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. 40–53, Jan. 2006.

[13] F. Hartanto and H. R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks," in *IEEE Tenth Workshop on Local and Metropolitan Area Networks (LANMAN)*, 1999.

[14] H. Seferoglu, Y. Altunbasak, O. Gurbuz, and O. Ercetin, "Rate distortion optimized joint ARQ-FEC scheme for real-time wireless multimedia," in *Int. Conf. Image Processing*, 2005.

[15] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–291, Apr. 2004.

[16] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1752–1763, Dec. 2003.

[17] H. S. Wang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 353–357, May 1996.

[18] A. Mukherjee, On the Dynamics and Significance of Low Frequency Components of Internet Load Univ. Pennsylvania, Philadelphia, Tech. Rep. MS-CIS-92-83, Tech. Rep., 1992.

[19] *Part 11: Wireless LAN Medium Access Control and Physical Layer (PHY) Specifications*, IEEE 802.11-1999 Standard, Aug. 1999.

[20] *Part 11: Wireless LAN Medium Access Control and Physical Layer (PHY) Specifications High-Speed Physical Layer in the 5 GHZ Band*, IEEE 802.11a Supplement to IEEE 802.11 Standard, Aug. 1999.

[21] JVT Reference Software [Online]. Available: http://bs.hhi.de/suehring/tml/download/ Available from

Antonios Argyriou (S'00–M'07) received the Diploma in electrical and computer engineering from Democritus University of Thrace, Greece, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering as a Fulbright scholar from the Georgia Institute of Technology, Atlanta, USA, in 2003 and 2005, respectively.

Currently, he is a Research Scientist at Philips Research, Eindhoven, The Netherlands. From 2004 until 2005, he was also a research engineer at Soft. Networks, Atlanta, GA, and was involved in the design of MPEG-2 and H.264 video compression algorithms. His current research interests include wireless networks and networked multimedia systems. He has published more than 30 papers in international journals and conferences.