# Multicast and Unicast Real-Time Video Streaming Over Wireless LANs

Abhik Majumdar, *Student Member, IEEE*, Daniel Grobe Sachs, *Student Member, IEEE*, Igor V. Kozintsev, *Member, IEEE*, Kannan Ramchandran, *Associate Member, IEEE*, and Minerva M. Yeung, *Member, IEEE*

***Abstract*—In this paper, we address the problem of real-time video streaming over wireless LANs for both unicast and multicast transmission. The wireless channel is modeled as a packet-erasure channel at the IP level. For the unicast scenario, we describe a novel hybrid Automatic Repeat reQuest (ARQ) algorithm that efficiently combines forward error control (FEC) coding with the ARQ protocol. For the multiple-users scenario, we formulate the problem of real-time video multicast as an optimization of a maximum regret cost function across the multicast user space. The proposed solution efficiently combines progressive source coding with FEC coding. We present a theoretical analysis of the unicast and multicast cases, as well as experimental results that demonstrate the performance advantages of the proposed algorithms over existing methods.**

***Index Terms*—Channel coding, forward error correction, IEEE 802.11, multicast channels, multimedia communication, video coding, wireless local area networks.**

## I. Introduction

Audio and video streaming over wired networks, such as the Internet, have been popular now for quite some time. However, with the development of broadband wireless networks, attention has only recently turned to delivering video over wireless networks. In this paper, we focus on the wireless Local Area Network (LAN), which can operate at high enough bit rates to allow transmission of high quality video data. Specifically, we investigate the IEEE 802.11b wireless LAN, though the ideas that we present are applicable to other wireless networks as well.

As is well known, video streaming is very different from data communication due to the inherent delay constraints, as late arriving data is not useful to the video decoder. In fact, it is better to drop such data at the sender rather than attempt to send it after the deadline has passed. For this reason, TCP/IP, which is designed to deliver data reliably (but asynchronously) and works very well for data communication, is not always the best solution for real-time video streaming. To this end, several industrial streaming media architectures (e.g., by Microsoft and Real Networks[1]) have been developed. However, the existing solutions were not designed specifically for the harsh conditions that are prevalent in a wireless channel.

With wireless networks gaining prominence and acceptance, especially the LANs based on the IEEE 802.11 standards, it is foreseeable that streaming of audio/video will be a critical part of the wireless digital infrastructure. Yet, to the best of our knowledge, video streaming applications have not been studied extensively for IEEE 802.11 based wireless networks. There are two major challenges for video streaming over wireless LANs: 1) fluctuations in channel quality and 2) high bit-error rates compared with wired links. One important contribution of this paper is the introduction of packet-erasure channel models for wireless LAN channels that allow us to develop video streaming algorithms that take into account the specifics of the IEEE 802.11 networks. In addition to wireless channel-related challenges, when there are multiple clients, we have the problem of heterogeneity among receivers, since each user will have different channel conditions, power limitations, processing capabilities, etc., and only limited feedback channel capabilities.[2] This paper contributes to the analysis of both the single-user and multi-user case in detail, and proposes practical solutions to address the afore-mentioned challenges.

There has been a substantial amount of prior work in the area of video streaming. The single-user scenario has been addressed by many researchers (see, e.g., [1]–[4] and references therein). In this paper, we introduce a new method that combines the reliability and fixed delay advantages of forward error control (FEC) coding with the bandwidth-conserving channel-adaptive properties of Automatic Repeat ReQuest (ARQ) protocol. Hybrid ARQ protocols of different flavors have been extensively studied in the literature [5]. However, to the best of our knowledge, there has been no work addressing the specifics of media streaming over wireless LANs. Our main contribution to the single-user case is a novel Hybrid ARQ algorithm that improves the performance of real-time video streaming over wireless LANs compared to existing techniques.

In a multicast scenario, to tackle the problem of heterogeneity and to ensure graceful quality degradation, the use of multi-resolution-based scalable bitstreams has been previously suggested in [6]–[8]. However, such a bitstream is sensitive

Manuscript received August 15, 2001; revised April 15, 2002.

A. Majumdar was with Intel Labs, Santa Clara, CA 95054-1537 USA. He is now with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (abhik@eecs.berkeley.edu).

D. G. Sachs was with Intel Labs, Santa Clara, CA 95054-1537 USA. He is now with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, 61801 USA (e-mail: sachs@uiuc.edu).

I. V. Kozintsev and M. M. Yeung are with Intel Corporation, Santa Clara, CA 95054-1537 USA (e-mail: igor.v.kozintsev@intel.com; minerva.yeung@intel.com).

K. Ramchandran is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: kannanr@eecs.berkeley.edu).

Publisher Item Identifier 10.1109/TCSVT.2002.800315.

[1]Third-party marks and brands are the property of their respective owners.

[2]In this paper, we do not address issues related to differences in power and processing capabilities among multicast users.

to the position of packet loss, i.e., the received quality is a function of which packets are erased. To overcome this problem, the Priority Encoding Transmission scheme was proposed in [1], which allows different resolution layers to be protected by different channel codes based on their importance. Substantial work has been made toward finding an algorithm for optimizing the amount of parity bits used to protect each resolution layer [2], [3], [9]. A near-optimal, $O(N)$ (where $N$ is the number of packets) complexity, algorithm was proposed in [9] to solve this problem. Termed multiple description coding using FEC codes, or the *MDFEC algorithm*, it partitioned the scalable video bitstream into a series of resolution layers and then applied varying amounts of protection on each of the layers, depending on the importance of the layer. To find the optimal amounts of protection, the MDFEC algorithm used a Lagrangian optimization-based $O(N)$ algorithm.

In this work, we focus on the video multicast scenario in the "last hop," i.e., between the base station and the mobile devices because it is likely to be the bottleneck of the whole video streaming system. As such, the ideas of multicast trees as proposed in [7] or the ideas of reliable and resilient multicast [10], [11] are not applicable here, since there is only one hop between the source (base station) and the destination (mobile devices). We use a maximal regret criterion to formulate the problem of optimal source and channel coding for the multicast case, similar to [12], [13]. In [13], the authors formulated the general multiple-user broadcasting problem for both the power and rate constraints. In contrast, we consider only the rate constraint case, but we explicitly formulate the problem of robust streaming of a progressive video bitstream over a wireless 802.11b LAN and derive the properties of the optimal solution for the case of $N = 2$ and $N > 2$ users. Our optimization algorithm takes advantage of the fact that the MDFEC algorithm automatically tries to match both the source and channel characteristics. The main contribution of our paper to the multicast case is in formulating the problem of real-time multicast of video using the maximum regret criterion, and providing the optimal solution for a special case of progressive source coder and MDFEC channel coder. To the best of our knowledge, no comparable schemes have been proposed in the literature for video multicast over wireless LANs in a similar setting.

The rest of the paper is organized as follows. Section II gives an overview of the related background information, including source and channel coding and the IEEE 802.11b Wireless LAN, as well as the motivation behind our design choices. In Section III, we present our novel Hybrid ARQ algorithm for real-time video unicast over wireless LAN. In Section IV, we formulate the problem of real-time video multicast over wireless LANs, and propose the optimal solution for a progressive source coder and MDFEC channel coder. Finally, we conclude with discussions and future directions in Section V.

## II. Background and Motivation: Source and Channel Coding for Video Over Wireless LANs

In this section, we offer an overview of the related background information and the motivation for our design choices. In Section II-A, we discuss the existing advanced source coding methods for multimedia. Single-resolution and progressive source coding are introduced, as well as distinctive characteristics of compressed real-time media. In Section II-B, we describe the IEEE 802.11 standard for wireless LANs and discuss the details relevant for multimedia streaming applications, as well as important differences of WLANs and wired LANs. In Section II-C, we discuss the packet-erasure channel model, that is used in this work to represent the Wireless LAN channel. In contrast to existing approaches that model wireless channels on RF level (using, e.g., AWGN or Rayleigh models), we approximate the WLAN channel at the IP level as it is "seen" by multimedia applications. Finally, Section II-D describes the existing channel-coding methods (FEC and ARQ) for data transmission over packet-erasure channels.

### A. Source Coding Methods for Multimedia Data

Multimedia data are normally compressed before being transmitted or stored. In this paper we concentrate on video but other multimedia data types (speech, audio, image, and graphics) can potentially be handled within the proposed framework. Multimedia compression methods can be broadly classified as follows. The traditional source-coding methods, referred to here as fixed-rate (or single resolution) coding, compress the data to a target bandwidth or a target storage size. Examples of such schemes are DPCM speech codecs, JPEG for image coding, MPEG 1 and 2 for video compression, etc.[3] Many such techniques exploit data dependency to achieve high compression ratios. However, this also introduces interdependency of data in the compressed form. Consequently, bit/packet errors in the compressed data due to unreliable transmission may render as significant artifacts after the compressed data is decoded. Similarly, when the available bandwidth/storage size increases, fixed-rate coding techniques are unable to take advantage of this. This type of single layer coding method works well for application domains, such as VCDs and DVDs, where a certain media storage size is targeted, or cable broadcasting applications, where dedicated bandwidths are assigned to each data channel. However, it does not always fit well for wireless packet networks where bandwidth is highly varying with time or/and across different receivers. As shown in Fig. 1, the fixed-rate coding achieves the highest (for this coding scheme) quality of data only at a single point—the target bitrate. The quality remains constant, even if available channel rate increases.

For a packet network with dynamic bandwidth, a different class of source coding techniques called progressive coding is better suited. In progressive coding (also referred to as scalable coding), multimedia data is divided into coding units [e.g., a video or audio frame(s)] and within those units, the data is arranged such that the most important part is placed in the beginning followed by the parts with successively lower importance. The decoding of the data within a coding unit can be *partial*, and the more data that can be decoded, the better the quality of the decoded multimedia data. Therefore, a single progressively coded multimedia stream can be served over a dynamic network where the bandwidth may vary from time to time for

[3]In advanced profiles, some of these coding methods allow generation of several resolutions of data at different bit rates or sampling frequencies. These profiles, however, are rarely used in real life.

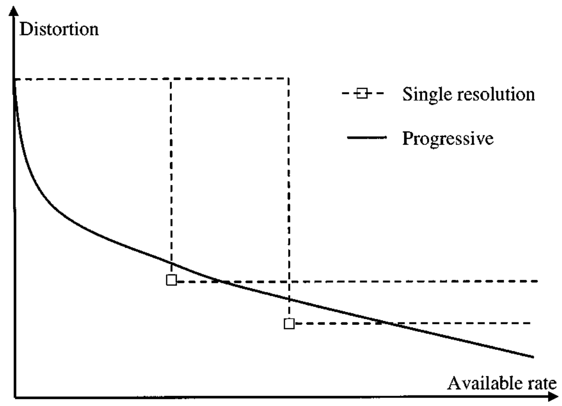



Fig. 1. Operational rate distortion characteristics of single resolution and progressive coder. Different points for a single resolution case correspond to separate coding schemes, while all the points on the progressive coder characteristic are obtained using a single coder.

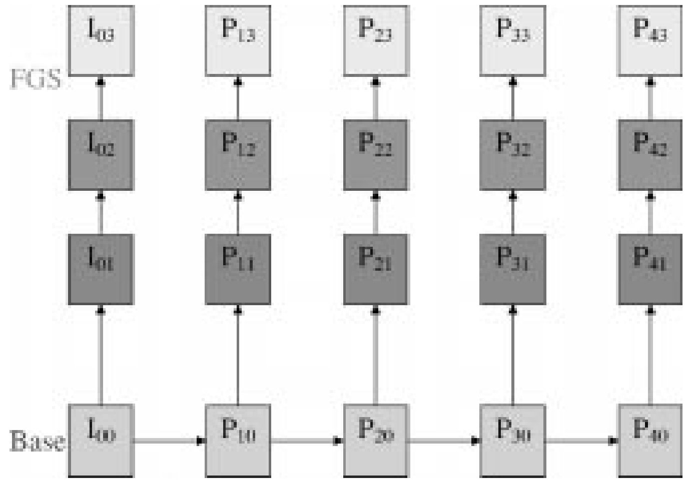



Fig. 2. Directed graph representing dependencies between data units in MPEG4 FGS stream. Horizontal dependencies in the base layer are due to predictive coding. Vertical dependencies correspond to successive bitplanes in a single frame.

a given receiver or vary from one receiver to other receivers. In Fig. 1, the distortion–rate characteristic of a progressive coding scheme is shown to be somewhat inferior to the performance of the fixed-rate compression scheme—a price we pay for the possibility of having a single coding scheme to cover a range of bit rates.

An important example of a progressive coding scheme for video that we use later on in this paper is the fine grain scalability (FGS) mode of MPEG4 [14]. In MPEG4-FGS, there are two separate bitstreams representing the video data. The base stream is an MPEG4-compliant bitstream. The refinement layer contains additionally a progressively coded bitstream that improves the quality of the base layer video. As long as the base layer is present, the refinement layer can be used to improve the quality of each video frame where it is present.

A useful abstraction for representing media data was proposed in [4]. The multimedia data is represented as a set of "data units," e.g., video frames, images, audio frames. Each data coding unit $l$ is associated with its size (in bits) $B_l$, distortion $d_l$ (corresponding to the amount of decoded signal distortion reduction due to decoding of unit $l$ on time), and the deadline $t_l$ by which the unit has to be decoded to be useful. Different coding units are proposed to be modeled as an acyclic directed graph that signifies the dependencies between data units. An example of such a graph for the MPEG 4 FGS video stream is shown in Fig. 2, where in addition to the dependencies in the base layer due to predictive coding, there are also dependencies in the refinement layers due to successive coding of bitplanes within a single frame. In [4], the graph model was used to solve a dynamic programming problem of distortion/rate optimal data unit transmission policy. Our work is different, since we specifically address the wireless channel model and we extend our investigation to include the important multicast scenario.

### *B. IEEE 802.11b WLAN Standard*

The IEEE 802.11 standard [15] along with the "b" annex define a 2.4-GHz spread-spectrum wireless LAN capable of operation at bit rates of 1, 2, 5.5, and 11 Mbits/s (using a spread-spectrum BPSK modulation) up to 11 Mbits/s (using different modulation methods). The IEEE 802.11 standard uses the same logical link layer as other 802-series networks (including the 802.3 wired Ethernet standard), and uses compatible 48-bit hardware Ethernet addresses to simplify routing between wired and wireless networks. As in the wired Ethernet, corrupted packets are dropped at the link layer; therefore, the wireless link appears as a packet loss network to applications running on it (note, that packets with bit errors are unavailable to a multimedia application!). However, significant differences between the properties of wired and wireless networks demand a very different media access control (MAC) layer for 802.11 wireless networks. Hence, we describe the relevant features of IEEE 802.11.

Using radio transceivers for the network physical layer is complicated by the inability of radio transceivers to detect collisions as they transmit, and the potential for devices outside the network to interfere with network transmissions. Communication is also hampered by the hidden node problem; two widely spaced nodes on the network may be unable to communicate with each other directly, and yet still interfere with transmissions to an intermediate point. To address these limitations, a complex media access control (MAC) that includes retransmissions of corrupt packets and collision avoidance is used. This complex MAC is required for adequate TCP performance, as TCP responds to packet losses by throttling the transmission rates. Therefore, packet losses not caused by congestion must be hidden by the MAC layer to allow TCP and TCP-friendly congestion control algorithms to choose an appropriate transmission rate.

Fig. 3 shows a typical example of two mobile stations communicating with an Access Point (AP). The Request To Send (RTS)/Clear To Send (CTS) mechanism is often used to avoid a hidden node problem. Station 2 accesses the channel (after sensing the channel as free for some interval of time) by sending an RTS frame. At this time, all other stations that received the RTS frame set the state of the channel as "busy" for the duration of time advertised in the RTS frame. The AP broadcasts the CTS frame so that all other stations that are out of range of station 2

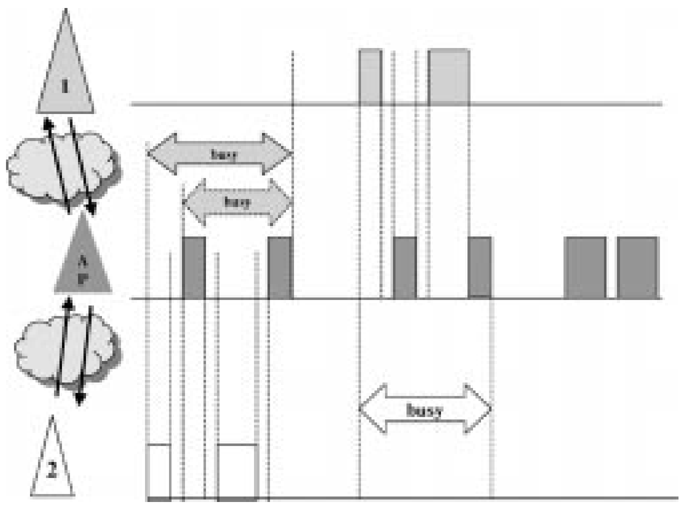



Fig. 3. Example of IEEE 802.11b LAN operation.

("hidden" nodes) can also set the channel state as "busy." After this procedure Station 2 sends a data frame that is acknowledged by the AP. A similar process is then repeated by station 1 when it needs to transmit (the time between the acknowledgment frame and the RTS frame from station 1 is available for all stations to initiate transmission). Finally, the AP multicasts several unacknowledged data packets (in this example not using the RTS/CTS mechanism).

At this point we would like to highlight important features specific to WLAN that influence the design of media streaming system. Notice that in a unicast traffic all wireless data frames are acknowledged by a recipient.[4] Acknowledgment (ACKs) are not "free," and therefore, they have to be used efficiently. Also notice that even UDP unicast frames are sent asynchronously to hide the impairments of a wireless channel from the transport layer protocols—a significant difference from wired IP networks. On the other hand, in case of multicast or broadcast traffic, the data packets are not acknowledged (and, hence, not retransmitted on the MAC/Logical link layer). This mode of communication delegates all error-control operations to the application layer, and therefore is used in the rest of this paper, even for the case of point-to-point streaming.

### *C. Packet-Erasure Model of IEEE 802.11 LAN*

The IEEE 802.11 MAC/logical link control (LLC) and physical (PHY) layers represent two lower layers in the open system interconnect (OSI) reference model, i.e., the data link and physical layers. In real life, we do not have direct access to the physical (or even MAC) layer. Furthermore, most of the successful wireless networks adopt the IP as a network layer simplifying the integration of wireless networks into the Internet networks. In this scenario, user applications see the wireless channel as an IP packet channel with erasures—much like the wired Ethernet. Therefore, in designing our algorithms (which run at the application layer), we model the wireless network channel as a packet-erasure channel at the network layer level.

[4]Link layer uses a well-known stop-and-wait ARQ [5].

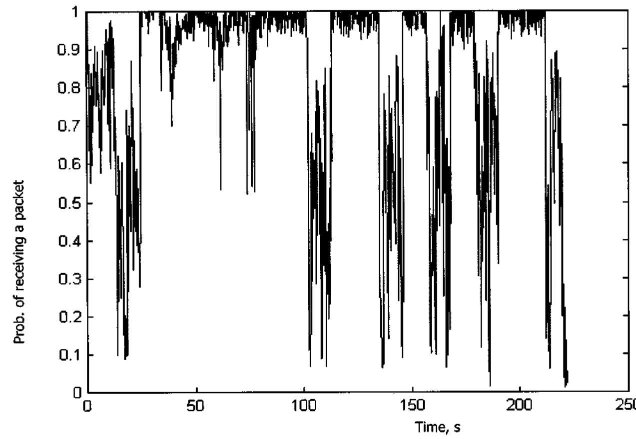



Fig. 4. Packet-erasure probability as a function of time in IEEE 802.11b trace.

Fig. 4 shows a sample trace[5] we obtained using the 802.11 AP and mobile station. The AP was sending UDP multicast packets at a constant rate and the receiving station was recording the sequence numbers of the correctly received packets. Each point on the trace corresponds to an average of about one hundred packets received. Note that the channel behavior in this experiment is the result of the interaction between the actual RF channel, the hardware responsible for PHY, and the software/firmware responsible for MAC/LLC and IP implementation. This approach leads to a more accurate model for the application layer algorithms that are the focus of this paper.

The simplest model to approximate a packet-erasure channel is to assume that the erasures are independent and identically distributed (i.i.d.) and have probability $P_e$. This model is justified if, for example, packet interleaving to a sufficient depth is used in the transmission (e.g., if long FEC codes are used). There is, however, gain to be explored by relaxing this i.i.d. assumption and attempting to model the correlations between channel properties in consecutive moments of time.

It is important to discuss one simple channel model based on a Markov chain because it can be useful in interactive applications and, more importantly, it draws an interesting connection between multiuser and a single user scenarios. Let us introduce hidden channel states $(S)$ that form a Markov chain with the initial distribution $\pi(s)$ and the transitional probabilities $P(s_i|s_{i-1})$. The probability of a packet erasure is represented by a conditional density $P_e(s) = \mathrm{Prob}(erasure|state = s)$. The number of states and the actual values of the erasure probabilities can be chosen in a number of ways. For instance, we may fix the number of states and try to choose the erasure probabilities to approximate the distribution of the erasure probabilities in the sample. Another reasonable approach is to let all states have equal probabilities, i.e., $P(s_i) = P(s_j)$. This last model allows us to draw a connection between the unicast and the multicast cases. If we make an additional assumption of no information about the channel at the transmitting side (due to absence of feedback channel, or in the case when channel changes too fast to be tracked at the transmitter), then *sending packets to a single user* over the described multistate

[5]The trace shows a channel with a rather poor quality because of the significant distance used in this measurement.

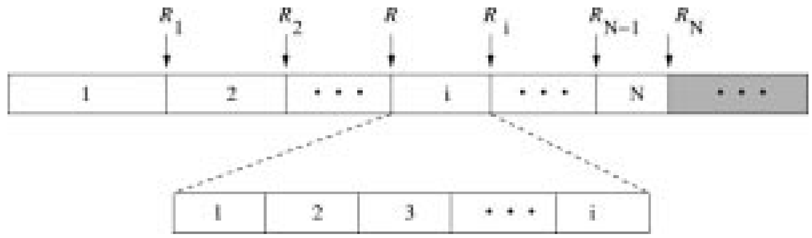



Fig. 5. A scalable bitstream partitioned into $N$ layers. The $i$th layer can be further decomposed into $i$ parts for channel coding.

packet-erasure channel *is analogous to sending data to multiple users* over different i.i.d. packet-erasure channels. This observation is an additional motivation to consider the problem of video multicast in a wireless LAN environment. In this paper, we use an independent packet-erasure model for the channel; however, our algorithms can be extended for the Markov model described above.

### D. Coding for Packet-Erasure Channels

In order to reliably communicate over packet-erasure channels, it is necessary to exert some form of error control. Two classes of communication protocols are used in practice to communicate data over packet networks: synchronous and asynchronous. Asynchronous communication protocols, such as ARQ, are reliable but have unbounded delay. ARQ operates by dividing the data into packets and appending a special error check sequence to each packet for error detection purposes. The receiver decides whether a transmission error occurred by calculating the check sequence. For each intact data packet received in the forward channel, the receiver sends back an acknowledgment. Thus, ARQ requires a two-way communication channel to be present. While this model works very well for data communication, it is not suitable for multimedia streams with hard latency constraints. The maximum delay of the ARQ mechanism is unbounded, and in multimedia applications it is usually preferable and, in the case of live streaming, necessary to interpolate late-arriving or missing information rather than insert a delay in the stream playback.

In synchronous protocols, the data are transmitted with a bounded delay but generally not in a channel adaptive manner. To provide for some measure of reliability, FEC coding is employed. FEC codes are applied to a group of source data packets. The FEC codes are designed to protect data against channel erasures by introducing parity packets. No feedback channel is required. If the number of erased packets is less than the decoding threshold for the FEC code, the original data can be recovered perfectly. However, FEC techniques cannot guarantee that the receiver receives all the packets without error.

Note that existing Internet streaming media servers and clients are based on a partially-synchronous version of the ARQ protocol. These applications maintain a record of the approximate round-trip time for a packet and its acknowledgment, and use this information to determine at the server if a packet is likely to arrive at the destination before its deadline. In this way, the unbounded delay of ARQ protocols can be avoided. However, even with this change, the ARQ-based protocols still require a small overall packet loss rate and low round-trip latency to achieve an acceptably small probability of stream transmission failure.

In this paper, we will focus on transmission protocols employing a particular class of erasure correction codes, namely Reed–Solomon (RS) codes. RS codes are described by two numbers $(n, k)$, where $n$ is the length of the codeword and $k$ is the number of data symbols in the codeword. Each symbol is drawn from a finite field of $2^s$ elements, where $s$ (we use 8) is the number of bits to be represented in each symbol. The total number of words in the code equals $2^s - 1$. RS codes can be used to correct errors, erasures, or both. Particularly efficient decoding algorithms based on Vandermonde matrices [16] exist if only erasures are to be corrected. In this case, each parity symbol can correct *any* one missing data symbol. This means that we can recover the original codeword, and hence the original data, if at least $k$ of the original $n$ symbols are received.

## III. Streaming Video Over Wireless LAN: A Single User Case

In this section, we present algorithms for streaming a video bitstream over a wireless LAN. The first approach described is the purely FEC-based MDFEC algorithm [9], which assumes as input a scalable video bitstream. On the other hand, the Hybrid ARQ protocol, is a combination of FEC and ARQ techniques and does not assume a scalable video bitstream as input, and therefore is readily applicable to existing video content stored on DVDs and VCDs. The problem we address in this section is that of finding the parameters for source and channel coding schemes for a single server and a single client, to maximize the overall data quality (or, equivalently, minimize the distortion) subject to a communication delay constraint.

### A. MDFEC

MDFEC is a transcoding mechanism to convert a prioritized multiresolution bitstream (see Fig. 5) into a nonprioritized multiple description bitstream (see Fig. 6) using efficient FEC codes.

Let $\mathbf{d}$ be an $N$-dimensional distortion vector (also called the distortion profile) where $d_k$ reflects the distortion attained when $k$ out of $N$ packets are received. The progressive bitstream is marked at $N$ different positions (that form $N$ resolution layers), which correspond to achieving the distortion levels $d_k$, as shown in Fig. 5. The $i$th resolution layer is split into $i$ equal parts and an $(N, i)$ RS code is applied to it to form the $N$ packets as shown in Fig. 6. Since every packet contains information from all the $N$ resolution layers, they are of equal priority. The RS code ensures that the $i$th resolution layer can be decoded on the

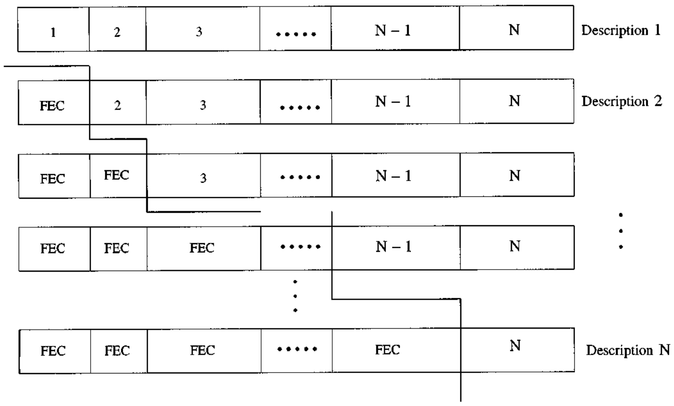



Fig. 6. Conversion of the prioritized scalable bitstream into an unprioritized one through unequal channel codes. Each packet offers an unprioritized equivalent description of the source.

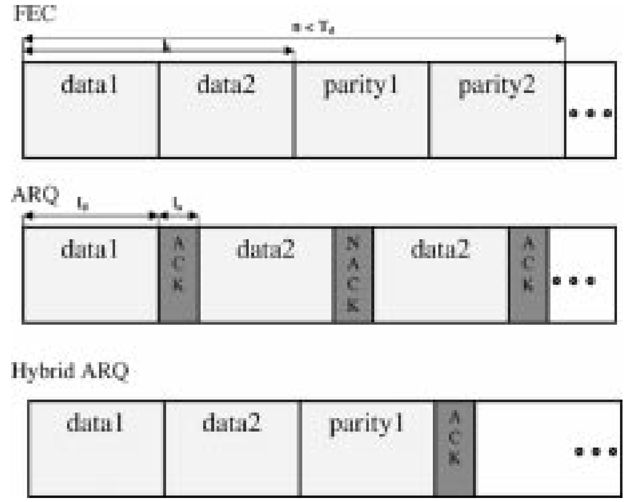



Fig. 7. FEC, ARQ, and Hybrid ARQ coding schemes.

reception of at least $i$ packets. Since the distortion-rate function $D(r)$ for a source is a one-to-one function of the rate $r$, finding the $N$ dimensional distortion vector $\mathbf{d}$ corresponds to finding the rate partition $\mathbf{R} = (R_1, \ldots, R_N)$ of the multiresolution bitstream (see Figs. 5 and 6). We note that, in this framework, the decoded quality at the receiver is strictly a function of *how many* packets are received and not *which* packets are received. Depending on the channel conditions at hand, the rate partition $\mathbf{R}$ can be optimized so as to match the given channel. A fast, near-optimal algorithm, of complexity $O(N)$ (where $N$ is the number of packets), based on Lagrangian principles, to solve this problem is described in [9]. The inputs to the algorithm are the source rate-distortion curve, a total transmission rate constraint and the transmission (channel) profile. Given these inputs, the algorithm outputs a quality profile (i.e., the reconstruction quality experienced at the receiver as a function of the number of packet losses) that attains the best expected average distortion for the given transmission rate.

### *B. Hybrid ARQ*

The MDFEC method is an attractive solution, but it requires progressive video input. Here we propose a way to combine the ARQ and FEC error control methods to improve the performance of unicast communications of single resolution video over packet-erasure channels. Hybrid ARQ schemes have been extensively studied in the literature for various communication channels. In this work we do not attempt to survey all of them (the reader is referred to [5] for a textbook treatment) but rather we propose a scheme that specifically addresses the problem of video streaming over 802.11 networks. Further, to the best of our knowledge, there has been little investigation of Hybrid ARQ schemes in a wireless LAN setting. As described in Section II-B, the acknowledgment have to share the same channel with the data and consequently too many ACKs can have a significant effect on the throughput. The proposed Hybrid ARQ scheme attempts to address this issue by efficiently reducing the amount of ACKs.

Our scheme is inspired by a similar idea proposed in [17] for rate-compatible punctured convolutional (RCPC) codes. The idea is illustrated in Fig. 7 (bottom). We start by splitting our multimedia data into "packet groups," consisting of $k$ packets each, and then, for each packet group, appending $n - k$ RS parity packets to the group as in the FEC coding scheme described above. However, unlike in the pure FEC scheme, we initially send only the first $k$ data packets to the receiver. Then transmitter starts sending parity packets until one of the following two events occurs: either an acknowledgment[6] from the receiver arrives, or the deadline for the transmission is reached. Once at least $k$ packets are received intact, the receiver sends an acknowledgment. Once the acknowledgment is received, the transmitter continues with the next $k$ data packets. One significant advantage of this algorithm is that it does not break down even when acknowledgments are lost. Instead, the transmitter simply assumes that more parity is needed.

The Hybrid ARQ scheme is a general algorithm and can be adjusted to fit specific cases as is appropriate. For instance, although interleaving is not described above, it is used in practice to improve bandwidth utilization during the time when an ACK is going from receiver to sender (i.e., after sending all the data packets for the current group of packets, the data/parity packets from other groups are used to interleave the parity packets of the current group).

To evaluate the performance of the Hybrid ARQ algorithm, it was both calculated analytically and implemented as a transport protocol for MPEG streams over 802.11b wireless networks [18]. The results are presented below.

### *C. Hybrid ARQ Throughput Analysis*

To compare the throughput of the proposed Hybrid ARQ scheme with the conventional ARQ and FEC schemes described above, we assume a memoryless packet-erasure channel with only a single user, and with $P_e$ being the probability of packet erasure. The throughput is defined as the ratio of time (in the channel) of $k$ data packets to the average time it actually takes to send them. Additionally, we only count those $k$ data packets that have transmission time less than some desired time $n$. This

[6]The acknowledgment is generated by a multimedia application. This is not to be confused with an ACK generated by the 802.11 MAC.

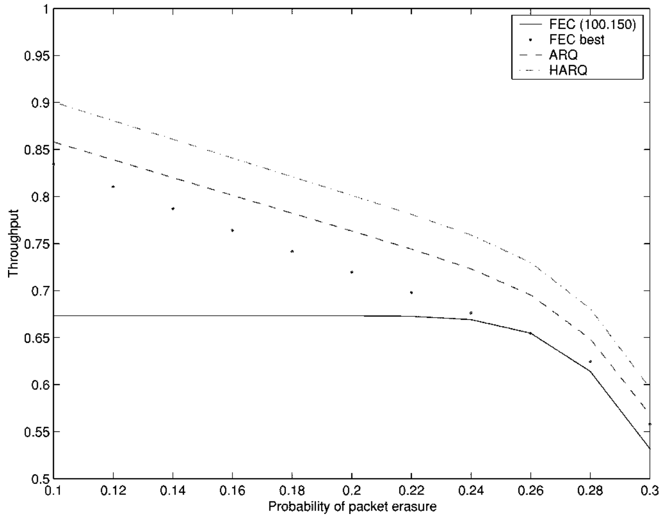



Fig. 8. Throughput for FEC, ARQ, and HARQ schemes for $l_d/l_a = 20$, $n = 150$ and $k = 100$.

last constraint specifically addresses real-time requirements of video.

Let $I_e^n$ denote a random variable that represents the number of packet erasures in a group of $n$ packets. Assuming an $(n, k)$ RS channel code, the throughput of the FEC coding scheme is equal to

$$\begin{aligned} T_{\text{FEC}} &= \frac{k}{n}\,\text{Prob}(I_e^n \le n-k) \\ &= \frac{k}{n}\sum_{j=0}^{n-k} P_e^j (1-P_e)^{(n-j)} \frac{n!}{j!(n-j)!} \end{aligned}$$

where $\text{Prob}(I_e^n \le n-k)$ is the probability of getting $n-k$ or fewer erasures in $n$ packets. The parameter $k$ can be chosen optimally for a given $P_e$.

To estimate the throughput of an ARQ scheme used in the 802.11 link layer, we assume that $l_d$ is the transmission time (in the channel) of a data packet, and $l_a$ is the transmission time of an acknowledgment, and we assume that no erasures occurs on the return channel. Let $E$ denote a random variable representing the total number of packets sent in a *successful* transmission of $k$ data packets. The throughput of the ARQ scheme is equal to

$$T_{\text{ARQ}} = \sum_{e=k}^{n} \frac{kl_d}{el_d + el_a}\,\text{Prob}(E=e)$$

where

$$\text{Prob}(E=e) = (1-P_e)^k P_e^{e-k}\,\frac{(e-1)!}{(e-k)!(k-1)!}.$$

Assuming, as before, no erasures in the return channel, independent erasures in the forward channel, the throughput of the described HARQ system can be found as

$$T_{\text{HARQ}} = \sum_{e=k}^{n} \frac{kl_d}{el_d + l_a}\,\text{Prob}(E=e).$$

The throughput for different coding schemes is shown in Fig. 8 for the case of $l_d/l_a = 20$, $n = 150$ and $k = 100$. Those parameters are chosen to be close to the parameters of a real system described in the next sections. "FEC best" corresponds to the convex hull of different FEC schemes with the same $n$ but different $k$ (we also include one particular $(n, k)$ combination performance on the plot). Notice that the throughput of the proposed HARQ method is better than that of the ARQ system because fewer ACKs are sent (see also Fig. 7 for a pictorial illustration of this). Both ARQ and HARQ outperform FEC in this setup, however, FEC will become optimal as the blocksize $(n)$ increases.

### D. *Experimental Results for Hybrid ARQ*

The physical network used for our testing was an otherwise unloaded network based on Cisco Aironet 340–series hardware. We found that the network was reliable at short range, occasionally dropping single packets. However, at longer ranges, fades would render the channel unreliable (and sometimes unusable) for seconds at a time, as shown in Fig. 4. In our tests, we also observed that the uplink from the PC card to the access point was somewhat less reliable than the downlink from the access point to the PC card, which is expected due to a significant difference in the form factor (i.e., antenna size).

The software and protocols were designed to transmit MPEG-encoded data from a fixed server, through the wireless access point and the 802.11b wireless network, and to a mobile receiver. In our implementation, the server reads the MPEG data from a file on its hard disk drive. The packet-erasure correction is based on the fast RS erasure correction codes described in [16]. MPEG decoding and display is done using the Intel Media Processing Library [19].

The testbed divides the MPEG data into groups of 100 packets, each of length 1024 B.[7] For each group of 100 packets, 50 parity packets are then created, in effect creating a (150, 100) RS code using 8-bit symbols. A sequence number is added to each transmitted packet to allow proper identification of each packet at the receiver and to ensure proper operation of the erasure-correction code.

Upon the receipt of each packet, the decoder checks the identity of the received packet. The receiver maintains spaces for each of the 150 possible packets in each group. If and when one of the packets from the current group is received, it is placed into the correct space. Once any 100 packets are received in a group, an acknowledgment is sent to the transmitter. If, due to severe fading, fewer than 100 packets are received out of the 150 sent, the RS decoder does not attempt to decode the packet group. Missing packets are dropped, along with all MPEG data until the next MPEG resync header.

We found that the Hybrid ARQ algorithm allowed the system to maintain the video stream during fades, even if the fade prevented data from being transmitted from the mobile to the access point. Both the pure FEC and the Hybrid ARQ scheme resulted in fewer video breakups than the use of a simple, ARQ-based TCP connection; in addition, the Hybrid ARQ algorithm effectively prevented the transmission of unnecessary FEC packets when channel conditions were good.

[7] The number of packets in the group and the protection level was chosen to compensate for channel erasures under typical conditions. An example of a channel trace in shown in Fig. 4.

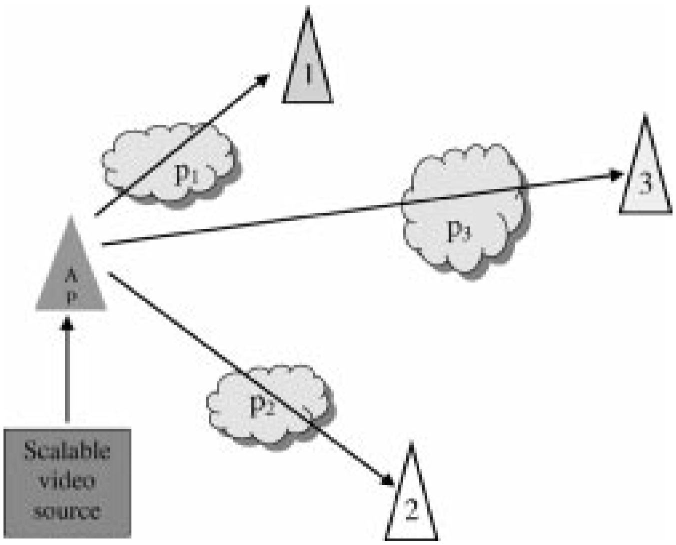



Fig. 9. Wireless multicast scenario with three users having different probabilities of packet erasure ($p_1 < p_2 < p_3$).

## IV. Streaming Video Over Wireless LAN: A Multiple User Case

In this section, we address the problem of multicasting a video bitstream over a wireless LAN (Fig. 9). It should be noted that ARQ-based schemes are less appropriate for the multicast case for two reasons: ACK explosions and the requirement to retransmit different packets to all users. For significant packet loss rates, each user will require frequent packet replacement, and different users are most likely to require different packets. To respond to requests by multiple users, we may have to resend a significant fraction of the original data even for small loss rates.

However, for small multicast networks, the hybrid ARQ scheme can alleviate the problem of sending different correction packets to each user. Because each parity packet can replace any missing data packet, there is no need for each user to identify which packet is missing. Instead, each user can simply transmit an acknowledgment when it receives enough parity to decode the transmitted data. When acknowledgment packets have been received from all known multicast users, the transmitter can move on to the next packet group. In this work, however, our approach is to use progressive video coding as described below.

### *A. Problem Formulation for Wireless Video Multicast*

When there is only a single client, it is clear that any coding scheme should seek to maximize the received user quality given a total rate constraint and a transmission profile. Since the scheme that maximizes the received quality for one client may not be the optimal one for other clients (since different clients will have different channel profiles and rate constraints), in the multicast scenario, it is desirable to maximize some composite delivered quality criterion, given the total rate constraint and the transmission profile.

It is difficult to arrive at an overall quality criterion for the multi-user case and any number of schemes, such as a weighted averaging scheme or simply designing for the worst-case receiver, could be suitable to a particular application scenario. Of course, the weighted averaging scheme can be easily mapped to the single user case and can be directly solved by the MDFEC algorithm. In this paper, we instead focus on a maximal regret criterion. Similar to the ideas proposed in [13], [12], we propose that the optimal coding scheme is the one that minimizes the following criterion:

$$\Delta(\mathbf{R}) = \max_i (E[d_i(\mathbf{R})] - E[d_i]_{\min})$$

where $\mathbf{R}$ is the rate partition as defined in Section III-A, $E[d_i]_{\min}$ is the minimum expected distortion for the $i$th client, achieved by using the optimal coding scheme when it is the only client, and $E[d_i(\mathbf{R})]$ is the expected distortion for the particular coding scheme being used. Such an overall quality criterion is fair in the sense that it minimizes the maximum penalty that any client suffers.

### *B. Proposed Solution*

We assume that all the clients have the same total rate constraint $R_{\text{tot}}$. For any rate partition $\mathbf{R}$, the total rate $R_t$ used is

$$R_t = \sum_{j=1}^{N} \alpha_j R_j \qquad \text{where } \alpha_j = \frac{N}{j(j+1)}, \\ \text{for } j = 1, \ldots, N-1 \text{ and } \alpha_N = 1.$$

Now for each client, we find the minimum expected distortion or $E[d_i]_{\min}$. This optimization problem is then formulated as

$$\min_{\mathbf{R}} E[d_i(\mathbf{R})]$$

subject to

$$R_t \le R_{tot} \quad \text{(resource constraint)} \tag{1}$$

$$R_1 \le R_2 \le \cdots \le R_{(N-1)} \le R_N \quad \text{(embedding constraint)} \tag{2}$$

where

$$E[d_i(\mathbf{R})] = q_{0i}E + \sum_{j=1}^{N} q_{ji} D(R_j)$$

and where $q_{ji}(N)$ is the probability of the $i$th client receiving $j$ out of $N$ packets, $D(r)$ is the source rate-distortion function, $E$ is the source variance, and $\mathbf{R} = (R_1, R_2, \ldots, R_n)$ is the rate partition (see Figs. 5 and 6). The above optimization problem can be solved using the MDFEC algorithm [9]. Just as in [9], we have relaxed the integer constraints on the $R_i$-s. After finding the $E[d_i]_{\min}$ for each client we are ready to solve the problem for the multi-user case. We state the problem as follows:

$$\min_{\mathbf{R}} \Delta(\mathbf{R}) \text{ subject to (1) and (2).}$$

Assuming that rate-distortion curve is convex (not always true in practice for operational rate-distortion curves), the expected distortion must also be convex in the rate partition, as the weighted sum of convex functions with positive weights is another convex function. Hence, $\Delta_i(\mathbf{R}) = (E[d_i(\mathbf{R})] - E[d_i]_{\min})$ is also convex in the rate partition. Consequently, $\Delta(\mathbf{R})$ has to

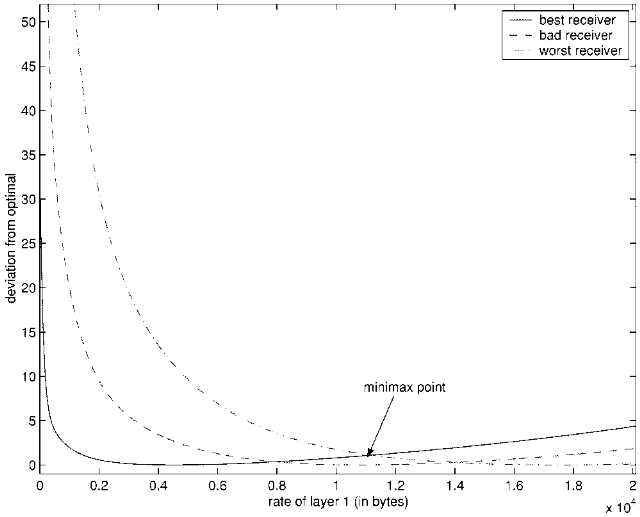



Fig. 10. Deviation from optimal performance (regret) as a function of the rate of layer 1.

be convex in the rate partition, as the maximum/supremum of any set of convex functions is convex. Further, the constraints on $\mathbf{R}$, i.e., (1) and (2), are also convex constraints and hence the problem of finding the minimax regret is one of convex optimization.

- Two client case: since $\Delta_i(\mathbf{R})$ is convex in $\mathbf{R}$ and the minimum value of $\Delta_i(\mathbf{R}) = 0$, $\Delta_1(\mathbf{R}) = \Delta_2(\mathbf{R})$ for the optimal rate partition. So, for the particular case of two clients, the algorithm finds the rate partition at which both the users are equally satisfied or equally disappointed. In this sense, the algorithm is fair to both users.
- For the rate partition that maximizes the overall quality criterion for more than 2 users, we will again have $\Delta_l(\mathbf{R}) = \Delta_j(\mathbf{R})$, where $l$ and $j$ are the two users for whom the $\Delta_i(\mathbf{R})$ versus rate partition curves (surfaces) intersect at the highest point relative to the intersection points of any pair of curves, and this will be the maximum disappointment or least satisfaction for any client (see Fig. 10). Thus, the $N > 2$ users problem can be solved by analyzing the users pairwise.

To actually find the rate partition that maximizes the overall quality criterion, we use a simplex optimization method although other standard techniques may be used [20]. While the computational complexity of the simplex algorithm may be high, we do not expect that the algorithm will be run for a very large number of users. This is due to the fact that the algorithm runs at each access point in the wireless LAN network and we do not expect many users to be present in vicinity of any AP.

To illustrate our approach, we performed some representative simulations. The source distortion rate functions are approximated as a weighted sum of decaying exponential functions as suggested in [21], fitted to match the operational rate-distortion curve for the Football video sequence, obtained using the 3-D SPIHT video coder [22]. The packet size was fixed at 2048 B. The channel is modeled as an independent packet-erasure channel with packet-erasure probability $p_i$. The parameters are that of a 802.11b wireless LAN channel estimated from real traces (see Fig. 4). The algorithm is run for a rate constraint of 50 packets/s. Table I shows the numerical results of the penalty $\Delta_i$ for each of the three users in the example above. "Worst case" refers to the MDFEC scheme designed for the worst channel, and "minimax" refers to the MDFEC scheme designed using our algorithm. Notice that as expected, in the worst case design the user with the most reliable channel is heavily penalized (by a factor of 4.6 compared to the minimax design), while under the minimax design, all three users suffer moderate increase in distortion. Fig. 10 illustrates the operation of the algorithm in the case of three clients as a function of the rate partitions used.[8] In Fig. 10, we have plotted the curves of $E[d_i] - E[d_i]_{\min}$ (deviation from the optimal) for all three users.

TABLE I
COMPARISON OF PENALTY IN DISTORTION FOR THREE USERS FOR MDFEC DESIGNED FOR THE WORST-CASE CHANNEL, AND FOR MDFEC DESIGNED FOR THE MINIMAX COST

| *user* | $p_i$ | $E[d_i]_{min}$ | worst case MSE penalty | minimax MSE penalty |
|---|---|---|---|---|
| 1 | 0.113 | 137.5322 | 32.2641 | 6.9583 |
| 2 | 0.156 | 162.8519 | 14.8577 | 0.3199 |
| 3 | 0.197 | 193.5069 | 0 | 6.9583 |

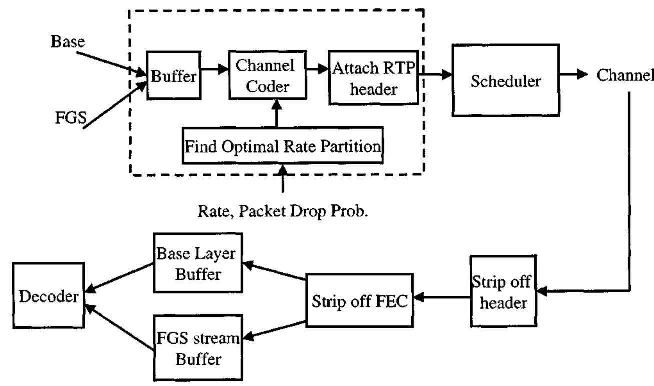



Fig. 11. Block diagram of experimental multicast system.

## C. Experimental Details and Results

The hardware used in the implementation of the above algorithm is the same as that used for the Hybrid ARQ case. However, for multicast, we use an MPEG4 FGS bitstream as the input scalable video bitstream. The server reads the MPEG data from two files (one for base and the other for FGS) stored in the hard disk. The block diagram of the system is shown in Fig. 11. The MPEG4 FGS video bitstream is multicast to a set of heterogeneous receivers over the 802.11 wireless LAN through a UDP socket. At the transmitter end, the frames are read into a buffer. This data is then passed on to the channel coder, which applies the relevant channel codes using the algorithm described in Section IV. The transmission profiles are obtained from the feedback channel. The protected bitstream is then packetized similar to the specifications in [23] and the Real-time Transport Protocol (RTP) [24] headers are attached to each of the packets.

[8]Here, we use only two descriptions/layers. Since there are only two layers, and the number of packets and the length of each packet is fixed, the only variable is the rate of the first layer.

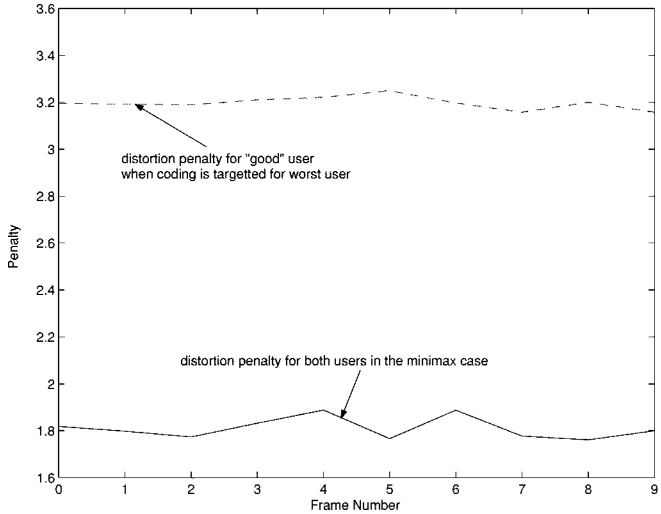



Fig. 12. Distortion penalty as a function of frame number for "akiyo" sequence.

The packets are subsequently multicast over the wireless network (802.11), through a UDP socket.

At the client end, the RTP headers are stripped off from the packets and channel decoding is applied. The bitstream is then separated into base and enhancement layers, and the base stream is written to the buffer for the base layer, while the enhancement layer stream is written to the FGS buffer. These buffers are then passed to the video decoder which displays the bitstream.

Fig. 12 demonstrates the measured performance (regret) of the proposed multicast system as a function of frame number for "akiyo" sequence encoded with CIF resolution at about 5 Mbits/s total rate. Notice that the proposed minimax-optimal algorithm evenly distributes the regret between the "good" (nearby) and the "bad" (distant) user and is, in this sense, fair to each user as opposite to the worst-case design where "good" user suffers significant performance degradation.

## V. Discussion and Future Work

In this work, we have studied the problem of real-time video streaming over wireless LANs, specifically the IEEE 802.11b wireless LAN. We addressed both the unicast and multicast cases in detail. For the multicast case, we have introduced a problem formulation that is fair to each user in the sense of quality degradation due to targeting multiple receivers. We derived the optimal solution that efficiently combines FEC coding with progressive source coding by extending the existing ideas of MDFEC to the multicast case. In the unicast case, we have proposed a novel algorithm for combining the advantages of FEC coding with those of the ARQ protocol. We have verified the performance of the proposed algorithms in a real-time streaming testbed.

A possible direction of future work would be to investigate whether the idea of Hybrid ARQ can be combined with the unequal error protection ideas of MDFEC for application to a multicast kind of scenario. How the proposed algorithm for the multicast case scales to larger networks is another possible angle that may be explored. In this work, we have only looked at the network between the wireless base station to the mobile client. It would be interesting to look at the video multicast case, where some parts of the network are wired and the rest wireless. Here, it might be possible to incorporate the ideas of multicast trees as described in [7] with our proposed algorithm.

## Acknowledgment

The authors thank anonymous reviewers who helped to improve this paper.

## References

[1] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.

[2] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *Proc. SPIE Conf. Wavelet Applications to Digital Image Processing*, Denver, CO, Aug. 1996.

[3] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Graceful degradation over packet erasure channels through forward error correction," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1999.

[4] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," IEEE Trans. Multimedia, 2001, submitted for publication.

[5] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[6] A. Balachandran, A. T. Campbell, and M. E. Kounavis, "Active filters: Delivering scalable media to mobile devices," in *Proc. 7th Int. Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97)*, St. Louis, MO, 1997, pp. 125–134.

[7] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM '96*, Stanford, CA, Aug. 1996, pp. 117–130.

[8] D. Wu, T. Hou, and Y.-Q. Zhang, "Scalable video transport over wireless IP networks," in *IEEE Int. Symp. Personal, Indoor and Mobile Radio Communication*, Sept. 2000, pp. 1185–1191.

[9] R. Puri, K. W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control framework for video streaming in the internet," *IEEE Trans. Multimedia*, vol. 3, pp. 18–32, Mar. 2001.

[10] S. Floyd, V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," in *Proc. ACM SIGCOMM '95*, Cambridge, MA, pp. 342–356.

[11] X. R. Xu, A. C. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous-media applications," in *Proc. 7th Int. Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97)*, St. Louis, MO, 1997, pp. 183–194.

[12] I. Kozintsev, "Signal processing for joint source-channel coding of digital images," Ph.D. dissertation, University of Illinois at Urbana-Champaign, May 2000.

[13] L. Qian and D. Jones, "Minimax disappointment wireless video broadcasting," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2001, pp. 449–452.

[14] *Final Draft Amendment 4*, ISO/IEC JTC1/SC29/WG11, MPEG00/N33904, 2001.

[15] *Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks-specific requirements—Part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications*, IEEE 802.11 standard, 1997.

[16] L. Rizzo. On the feasibility of software fec. DEIT Technical Report LR-970 131. [Online]. Available: http://www.iet.unipi.it/luigi/softfec.ps.

[17] J. Hagenauer, "Rate compatible punctured convolutional codes (RCPC-codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1988.

[18] D. G. Sachs, I. Kozintsev, M. Yeung, and D. L. Jones, "Hybrid ARQ for robust video streaming over wireless LANs," in *Proc. Inform. Technol.: Coding and Computing, ITCC2001*, 2001, pp. 317–321.

[19] M. Yeung, "Media processing library," in *Intel Developer Forum*, Aug. 2000.

[20] D. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[21] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Kozintsev, "Joint source channel matching for a wireless communications link," in *Proc. IEEE Int. Conf. Communication*, 1998, pp. 482–486.

[22] B. J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1374–1387, Dec. 2000.

[23] IETF, (2001). An RTP payload format for erasure-resilient transmission of progressive multimedia streams. [Online]. Available: ftp://ftp.nic.ad.jp/internet-drafts/draft-ietf-avt-uxp-00.txt

[24] *RFC 1889, Standards Track*, RTP: A transport protocol for real-time applications, 1996.

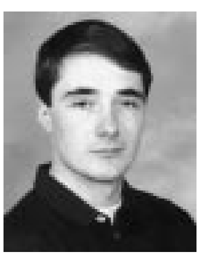

**Abhik Majumdar** (S'01) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology (I.I.T.), Kharagpur, India, in 2000. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.

His research interests include multimedia networking and wireless communications.

Mr. Majumdar was awarded the Institute Silver Medal from I.I.T. Kharagpur for being the best student in the graduating class of Electronics and Electrical Communication Engineering in 2000.

**Daniel Grobe Sachs** (S'01) received the B.S. degree in computer science in 1998 and the M.S. degree in electrical engineering in 2000 from the University of Illinois at Urbana–Champaign (UIUC), where he is currently working toward the Ph.D. degree.

He is currently a Research Assistant with the Coordinated Sciences Laboratory, UIUC. During the summer of 2000, he was with Intel Microprocessor Research Labs, Santa Clara, CA, and in the summer of 2001, he was with MIT Lincoln Laboratories, Lincoln, MA. His research interests include multimedia and wireless systems and networks, and end-to-end system optimization for wireless and portable media devices.



**Igor V. Kozintsev** (S'00–M'00) received the Diploma (with honors) in electrical engineering from Moscow State Technical University, Moscow, Russia, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC) in 1997 and 2000.

Since 1996, he has been a Research Assistant at the Image Formation and Processing Laboratory, Beckman Institute for Advanced Science and Technology, UIUC. In May 2000, he joined Intel Labs, Intel Corporation, Santa Clara, CA, where he is a Senior Researcher. His research interests include multimedia processing, digital signal processing, wireless communications, and networking.

**Kannan Ramchandran** (A'01) received the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1984 and 1993, respectively.

From 1984 to 1990, he was a Member of Technical Staff in the area of telecommunications at AT&T Bell Labs. From 1990 to 1993, he was a Graduate Research Assistant at the Center for Telecommunications Research, Columbia University. From 1993 to 1999, he was a Research Assistant Professor in the ECE Department, University of Illinois at Urbana-Champaign (UIUC), and a Research Assistant Professor at the Coordinated Science Laboratory, Beckman Institute, UIUC. Since 1999, he has been an Associate Professor in the EECS Department at the University of California at Berkeley. His research interests include distributed algorithms for signal processing and communications, multi-user information theory, audio-visual signal compression and transmission, wavelet theory and multirate signal processing, and unified algorithms for multimedia signal processing, communications, and networking.

Dr. Ramchandran was an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He was a recipient of the 1993 Elaihu I. Jury Award at Columbia University for the best doctoral thesis in the area of systems, signal processing, or communications. His research awards include the NSF CAREER award in 1997, and ONR and ARO Young Investigator Awards in 1996 and 1997, and the Okawa Foundation Award at UC Berkeley in 2000. In 1998, he was selected as a Henry Magnusky Scholar by the ECE Department at UIUC, to recognize excellence among junior faculty. He is the co-recipient of two Best Paper Awards from the IEEE Signal Processing Society, the 1996 Senior Paper Award and the 1998 Signal Processing Magazine Paper Award. He is a member of the Technical Committees of the IEEE Image and Multidimensional Signal Processing Committee and the IEEE Multimedia Signal Processing Committee.

**Minerva M. Yeung** (S'93–M'96) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, and the Ph.D. degree from Princeton University, Princeton, NJ.

She is a Principal Engineer with Intel Corporation, Santa Clara, CA, and leads the R&D of next-generation digital media and signal processing technology, computing models, and platform architecture in Intel's Microprocessor Research Labs. Previously, she was a Research Staff Member with IBM T.J. Watson Research Center, Yorktown Heights, NY. Her current research interests include distributed computing and signal processing, media security/management, wireless communications in ubiquitous computing, and computer architecture for emerging computing platforms. She has (co-)authored over 50 technical and invited papers, and holds eight U.S. patents and more than ten pending applications. Her work has been featured in such publications as the *New York Times*, *EE Times*, and *IEEE Spectrum* (April 2001).

Dr. Yeung served as an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA (1999–2001). She Co-Chaired the Wireless Multimedia Industrial Forum and the Wireless Communications Session in the Frontiers of Engineering Symposium of the National Academy of Engineering.