Efficient Cross-Layer Error Control for Wireless Video Multicast

Ivan V. Bajić
School of Engineering Science, Simon Fraser University
8888 University Drive, Burnaby, B.C. V5A 1S6, Canada
Tel: 1-604-268-7159
Fax: 1-604-291-4951
ibajic@sfu.ca

Abstract

In this paper we develop cross-layer schemes combining source rate selection, power management, and error control for wireless multicast of SNR scalable video. In particular, two error control schemes are analyzed and compared. The first one, type 2 hybrid ARQ/FEC, is based on incremental redundancy chosen to recover losses and provide the required level of video quality. The second one, called noncausal error control, relies on error concealment from future video data whose quality is adaptively adjusted to bring the video quality up to the desired level for all users. Our results indicate that noncausal error control can be significantly more efficient than type 2 hybrid ARQ/FEC.

Index Terms

Wireless video multicast, error control, error concealment, power management.

I. INTRODUCTION

Error control schemes for unicast video transmission include Forward Error Correction (FEC), Automatic Repeat reQuest (ARQ), and error concealment [1]. FEC is an open-loop error control scheme that adds redundancy to the bitstream prior to transmission. It is suitable for communication scenarios where the feedback from the receiver is not feasible, such as interactive conversational services. Because no feedback is assumed, packet loss needs to be predicted, and redundancy is designed for a range of possible packet loss realizations. However, only one of the possible loss realizations actually happens, so FEC usually ends up being either inefficient, or ineffective [2]. When feedback is feasible, as in video streaming, ARQ schemes can be more efficient than FEC [3]. With ARQ, packet loss need not be predicted as with FEC, but simply observed and reported to the server. The redundancy is then chosen to exactly match the observed loss.

Error concealment has traditionally been thought of as the “last line of defense” against packet loss [4]. There are several reasons for this. Error concealment is an application-layer technique, so it can be employed over any communication infrastructure, and activated whenever other error recovery methods fail. Further, error concealment relies on *estimation* of lost pieces of video data from the available ones, so it only provides approximate recovery of missing information. It might not always be possible to recover missing video data with sufficient quality. However, the error concealment method of [5], which we employ in one of the error control schemes described in this work, allows us to easily determine the quality of concealed data and decide whether it would be able to satisfy quality constraints. Hence, this error concealment method can form the “first line of defense,” and other error control schemes can be employed only if the server determines that the quality of the concealed data would be too low.

When multiple users need to receive the same data, multicast transmission can offer several advantages over unicast transmission, such as bandwidth savings and reduced resource requirements at the server and intermediate nodes. As the number of multicast users increases, the savings generally increase. But the large number of users also makes error control more challenging, because different users experience different losses. This is especially true in wireless multicast, where high error rates are common.

Our focus in this work is on centralized error control schemes for multicast, where the responsibility for providing error recovery rests with a single entity, typically the video server itself. The most efficient of these schemes seems to be type 2 hybrid ARQ/FEC [6] (T2HA/F). In this scheme, each user periodically reports the number of received packets out of previously transmitted k packets, so the server can calculate how many packets has each user lost. Suppose the user with the largest loss has lost r out of k transmitted packets. The server would then transmit r parity packets (constructed from, say, Reed-Solomon (n, k) code with $n - k \geq r$) on the multicast channel. These r parity packets will enable each user to correct its own losses. The following example illustrates why this scheme is more efficient than standard ARQ. Suppose six packets were sent to three users. In Table I, packet reception is indicated by 1 and packet loss by 0. In this case, if using standard ARQ, the server would need to transmit four packets (2, 3, 4, and 6), while only two parity packets would be needed if T2HA/F is used.

Packet #	1	2	3	4	5	6
User 1	1	1	0	1	1	1
User 2	1	1	1	0	1	1
User 3	1	0	1	1	1	0

TABLE I

SIX PACKETS SENT TO THREE USERS.

In this paper we consider a wireless video multicast streaming system whose goal is to maintain received video quality for all users above a certain pre-specified level, while minimizing the energy used to transmit the video. We present two error control schemes suitable for this task, along with the corresponding power management algorithms. Both schemes make use of the receiver-side buffers which are common in video streaming [1]. One of the schemes is T2HA/F mentioned above. The other one is an extension of noncausal error control (NCEC) [7] to the multicast case. NCEC is based on error concealment of lost data from future data using backward error concealment. Since future data is used as the reference for error concealment, its quality can be adjusted so that the required level of quality is achieved in the damaged portion of the video when error concealment is applied. The novel aspects of this work include:

- Cross-layer formulation of type 2 hybrid ARQ/FEC (T2HA/F) for wireless video multicast. Distortion-rate (D-R) information from the application layer is used to set power levels for packet transmission at the physical layer, thereby providing unequal error protection.

- Extension of another cross-layer error control scheme, the noncausal error control (NCEC) [7], to the multicast case. Loss of video packets is concealed from future video data, which plays a double role in our video multicast system - first, as the video data itself; and second, as the error control mechanism.
- Comparison of the energy efficiency of T2HA/F and NCEC in maintaining the required video quality at the receivers. The results indicate that NCEC can be significantly more efficient than T2HA/F.

The work described in this paper is an extension of our work in [9]. The paper is organized as follows. In Section II we provide an overview of a 3G wireless video multicast system. Section III describes the two error control schemes (T2HA/F and NCEC) that can be used in wireless video multicast. Their comparison is reported in Section IV, and conclusions are presented in Section V.

II. Wireless video multicast system

Figure 1 shows the key components of a generic 3G wireless video multicast system, which can be built on various technologies, e.g. UMTS [10] or cdma2000 [11]. Compressed video content is stored at the server in a scalable format. The server sends the video stream to radio network controllers (RNCs), which control the operation of base stations. Video content is transmitted on a broadcast channel. Depending on the content they wish to receive, users tune into the appropriate broadcast channel while maintaining a low-bandwidth dedicated connection to the base station. This dedicated channel is used for service and content updates, mobility management, and error control-related feedback, as described below.

A. Video coding, packetization, and decoding

The video coder used in this work is the fully scalable Motion-Compensated Embedded ZeroBlock Coder (MC-EZBC) [12], [13]. The basic coding unit is the group of pictures (GOP), typically 8 or 16 frames long. Video bitstream is packetized on a GOP-by-GOP basis into 512-Byte packets, which include a 4-Byte header, used as follows: 11 bits specifying GOP number, 5 bits specifying packet number within a GOP, and 16 cyclic redundancy check (CRC) bits. Application-level payload consists of 508 Bytes of compressed video data, so the number of bits

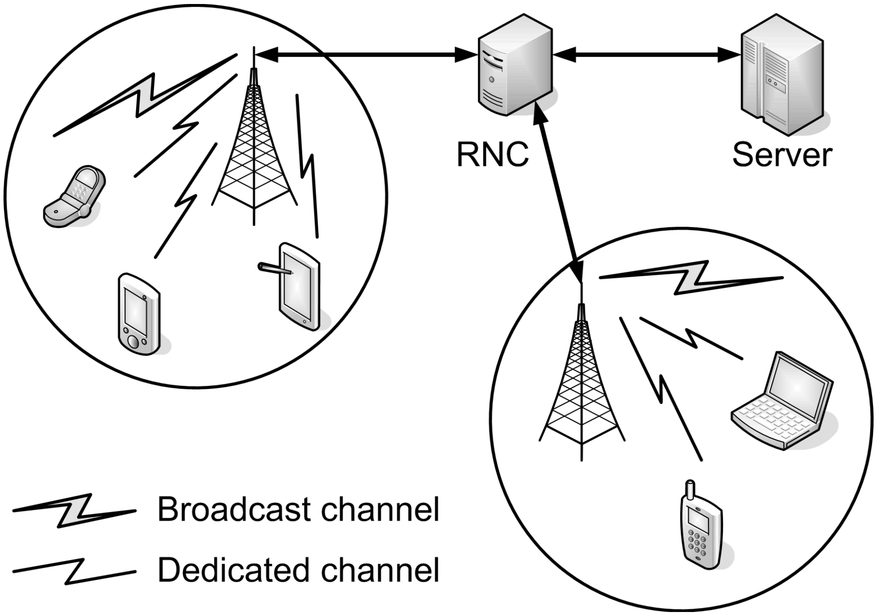

Fig. 1. Illustration of wireless video multicast.

in the payload is $B_p = 508 \times 8 = 4064$. The bandwidth expansion factor due to packet headers is $\eta = 512/508 = 1.008$.

Each user sends one small packet per GOP back to the server using its dedicated channel. This feedback packet indicates which packets were received in the previous GOP, so the server can determine packet loss for each user, and adjust its future actions accordingly. The loss of a packet is indicated by bit 0 in the feedback packet, while the reception is indicated by bit 1. If there are N packets in a GOP, feedback related to that GOP will require N bits. Using this feedback information the server decides on the number of parity packets that need to be transmitted, in the case of T2HA/F. In the case of NCEC, the server can find how much of the bitstream can each user decode, and then adjust the quality of future data which will be used as the reference for error concealment. In our experiments in Section IV, the number of packets per GOP never exceeded 16, so feedback information was limited to 16 bits. Since the amount of feedback information is so small, we assume the feedback packet is always received correctly, because its small size allows for heavy protection - either by increasing its transmission power, or using strong channel coding, or both.

The error concealment algorithm used here is the backward error concealment from [5], which makes use of the properties of motion-compensated temporal filtering to recover data lost in a given GOP (say GOP $i-1$) from the data in the following GOP (in this case, GOP i) . The quality of the recovered data depends on the number of decodable bits in both GOPs, which in turn depends on the number of packets received in both GOPs. Figure 2 shows an example of the operational distortion-rate (D-R) surface produced by this error concealment algorithm. The surface corresponds to the maximum frame distortion (as measured by mean squared error - MSE) within GOP $i-1$ after error concealment from GOP i, as a function of the number of packets received in these two GOPs, N_{i-1} and N_i, respectively. We denote such a surface as $\overline{D_c^{i-1}}(N_{i-1}, N_i)$. If the rate is expressed in terms of bits instead of packets, the surface could be denoted $\overline{D_c^{i-1}}(N_{i-1}B_p, N_i B_p)$. This example was produced using the grayscale QCIF sequence *Foreman* at 30 frames per second (fps), encoded using the 8-frame GOPs, with $i=2$. Operational D-R concealment surfaces are stored at the server, and used to adjust the quality of future data as necessary to maintain the received video quality above the specified level. Observe that the quality of the concealed data improves as the number of packets (and consequently the quality of the data) in the following GOP increases. This enables the server in the NCEC mode to choose the appropriate number of packets in the following GOP, which will bring the quality of the concealed data up to the desired level.

Both error control schemes described in this paper require compressed video data to be buffered at the receiver during the decoding process. This buffering enables the server to take error control action prior to video display. The receiver-side buffer needs to hold two GOPs - the current one (GOP i), and the previous one (GOP $i-1$), which is subject to error control. In the simulations described later in the text, we used QCIF sequences at 30 fps, with transmission rates of up to 215.0 kilobits per second (kbps). With 8-frame GOPs, the amount of memory needed to hold two compressed GOPs is no more than $2 \times (215/8) \times (8/30) \approx 14.4$ kiloBytes.

B. Wireless channel model

We adopted a simple and tractable wireless channel model in this work. We model the wireless channel as a Rayleigh fading channel with additive white Gaussian noise. The channel attenuation remains constant over one bit interval, but may vary between bits, and the sequence of fading levels is wide-sense stationary. Hence, if H_i is the random variable representing fading during bit i, we have $E[H_i] = const = E[H]$. Different users may have different values of $E[H]$. The

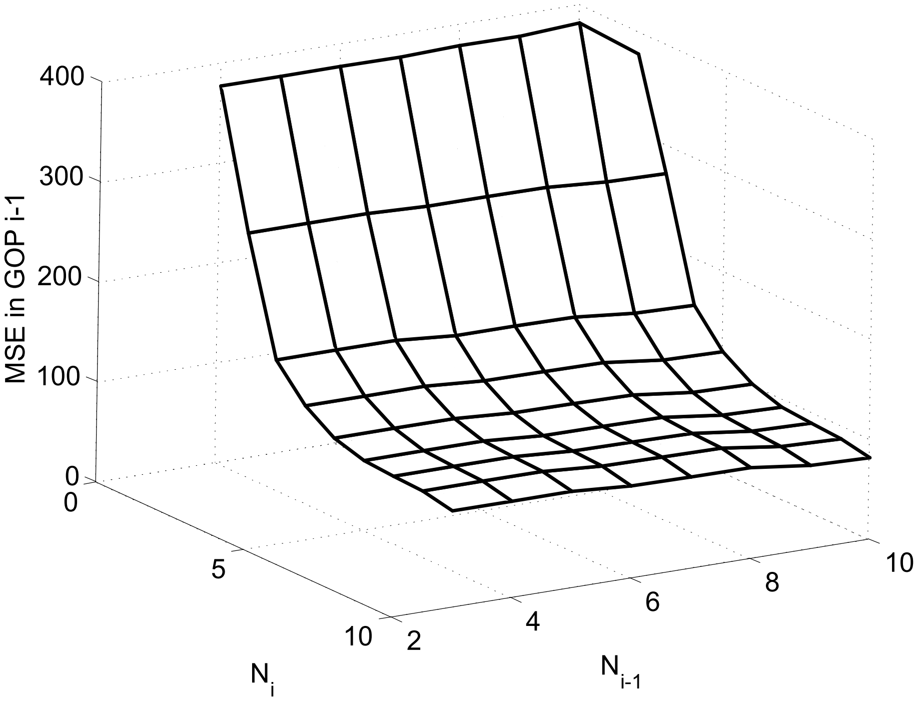

Fig. 2. Maximum frame distortion in terms of MSE in GOP $i-1$ as a function of the number of decoded packets N_{i-1} in GOP $i-1$ and the number of decoded packets N_i in GOP i, when backward error concealment is used.

modulation scheme considered here is binary differential phase-shift keying (DPSK). Bit error probability of binary DPSK over a Rayleigh fading channel [14] is $P_b = 1/(2+2E_b/N_0)$,where E_b is the received energy per bit period, and N_0 is the noise power spectral density. Transmission power is kept constant over the duration of a packet, but can be changed between packets. Hence, while packet n is being transmitted at a power level P_n, average received energy per bit is $E_{b,n} = P_n E[H]/R_t$, where R_t is the transmission rate. The expression for bit error probability in packet n now becomes $P_{b,n} = 1/(2+2P_n/G)$,where $G = N_0 R_t/E[H]$. The packet is discarded and considered lost if any of its bits are erroneously received. Hence, the loss probability of packet n is

$$p_n = 1 - (1 - P_{b,n})^{\eta B_p} = 1 - \left(1 - \frac{1}{2 + 2P_n/G}\right)^{\eta B_p}. \tag{1}$$

This relationship between transmission power P_n and packet loss probability p_n will be used in the following sections to compute appropriate transmission power levels.

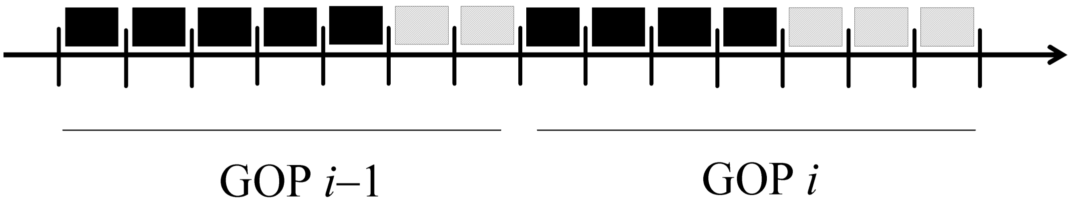

Fig. 3. Packets on the broadcast channel: solid blocks represent video packets, dashed blocks represent error control packets.

III. Power management and error control

The goal of power management is to minimize the energy used to transmit a given video while maintaining the expected distortion of the received video below a certain level $D_{\max}$. Let $|GOP|^i$ be the number of frames in GOP i, and f be the frame rate. For real-time transmission, we have a time slot of $|GOP|^i/f$ seconds to transmit all the information related to GOP i. With packet payload size B_p, bandwidth expansion factor η, and transmission rate R_t, the maximum number of packet slots in a time slot for GOP i is

$$M^i_{\max} = \left\lfloor \frac{R_t \times |GOP|^i}{\eta \times B_p \times f} \right\rfloor, \tag{2}$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than x. Some of these slots will be used for video information packets, while the rest will be used for error control packets related to the previous GOP, as shown in Figure 3.

Error control packets can either be parity packets used to correct losses in the previous GOP, or extra video packets used to improve the quality of the future video data, so that the quality of concealed data in the previous GOP is brought up to the required level. Depending on the packet loss and the error control scheme being used, there are three possibilities for choosing what to send as error control:

A. No loss in the previous GOP - in this case, no error control packets are sent, and all the packet slots for the current GOP are allocated to the video data from the current GOP. This problem is explored in Section III-A below.

B. Some loss in the previous GOP, followed by transmission of parity packets - in this case, explored in Section III-B, the server needs to determine how many parity packets need to be sent. This number equals the largest number of lost packets among all users.

C. Loss in the previous GOP, followed by backward error concealment - in this case, the server needs to determine how many packets from the following GOP need to be sent in order for error concealment to produce concealed data of sufficient quality. This can be done with the help of D-R concealment surfaces such as the one shown in Fig. 2. Section III-C describes this case.

In the T2HA/F mode, the server needs to consider cases A and B above. In the NCEC mode, all three cases need to be considered. Although error concealment is the first option for error recovery in NCEC, it might not always be able to provide the required quality of reconstruction, for example, when there is complex motion or a scene change in the video. In that situation the server may need to transmit parity packets for error recovery. Details are given in Section III-C.

A. No loss in the previous GOP

In this section we assume that no packets have been lost in the previous GOP, so the server only needs to send the data related to the current GOP. Let $\overline{D^i}(B)$ denote the maximum individual frame distortion in GOP i after decoding B bits. Operational distortion-rate (D-R) curves $\overline{D^i}(B)$ are easy to obtain with a SNR scalable codec - video needs to be encoded once, and decoded at multiple points of interest. By definition of $\overline{D^i}$, if $\overline{D^i}(B) \leq D_{\max}$ for some B, then after decoding B bits all frames in GOP i will have distortion of no more than $D_{\max}$. Let N^i be the minimum number of packets in GOP i needed to satisfy the distortion constraint $\overline{D^i}(N^i B_p) \leq D_{\max}$ in the loss-free case, i.e.

$$N^i = \min\{N \in \mathbb{N} : \overline{D^i}(NB_p) \leq D_{\max}\}. \tag{3}$$

If all packets up to packet n in GOP i are received, but packet n is lost, we can decode up to $(n-1)B_p$ bits in that GOP. Let p^i_j be the loss probability of packet j in GOP i. For independent losses, the expected maximum frame distortion at the receiver for GOP i is

$$E\left[\overline{D^i_R}\right] = \sum_{n=1}^{N^i} \prod_{j=1}^{n-1} (1-p^i_j) p^i_n \overline{D^i}\left((n-1)B_p\right) + \prod_{j=1}^{N^i} (1-p^i_j) \overline{D^i}\left(N^i B_p\right). \tag{4}$$

So, the expected distortion for GOP i is a function of the operational D-R curve $\overline{D^i}\left(\cdot\right)$ and the loss probabilities of packets in GOP i, which in turn depend on power levels used to transmit those packets, according to (1). Obviously, if $M^i_{\max} < N^i$, there are fewer packet slots available for GOP i than the number that is needed to satisfy distortion constraint in (3). Hence, in our

R_t (kbps)	$M^i_{\max}$	$\text{PSNR}_{\min}$ (dB)	N^i
61.4	4	*Claire*	
76.8	5		
92.2	6	25.0	2
107.5	7	27.5	2
122.9	8	30.0	$2-3$
138.2	9	*Foreman*	
153.6	10		
169.0	11	25.0	$3-5$
184.3	12	27.5	$4-7$
199.7	13	30.0	$5-9$
215.0	14	30.0	$5-9$

TABLE II

PARAMETERS USED IN THE SIMULATIONS.

simulations described in Section IV, the transmission rate of the broadcast channel in (2) is always chosen sufficiently high to support the required quality. Typical values of $M^i_{\max}$ and N^i for the two test sequences (*Claire* and *Foreman*) are listed in Table II, where distortion constraints are shown in terms of the minimum Peak Signal-to-Noise Ratio (PSNR) in dB.

If the number of available packet slots $M^i_{\max}$ is larger than the number of slots needed to support the required quality, some packets might be repeated in more than one slot. As demonstrated in [7], this leads to energy savings. Suppose packet j in GOP i is repeated M^i_j times ($M^i_j \geq 1$) during the transmission of GOP i. Then the decoder will be able to decode this packet as long as at least one of its M^i_j copies is received. The corresponding loss probability reduces to $\left(p^i_j\right)^{M^i_j}$, and one can write the equation for expected distortion $E\left[\overline{D^i_R}\right]$ as

$$E\left[\overline{D^i_R}\right] = \sum_{n=1}^{N^i}\prod_{j=1}^{n-1}\left(1-(p^i_j)^{M^i_j}\right)\left(p^i_n\right)^{M^i_n}\overline{D^i}\left((n-1)B_p\right) + \prod_{j=1}^{N^i}\left(1-(p^i_j)^{M^i_j}\right)\overline{D^i}\left(N^iB_p\right). \quad (5)$$

The goal is to minimize the total energy E^i_{tot} spent on transmission of GOP i, while keeping the expected distortion at the receiver below $D_{\max}$. For each packet $n = 1, 2, ..., N^i$ in GOP i, we want to find the power level P_n^i and the number of repetitions M_n^i so as to

$$\begin{aligned} \text{minimize} \quad & E_{\text{tot}}^i = \sum_{n=1}^{N^i} \frac{M_n^i \eta B_p}{R_t} P_n^i, \\ \text{subject to} \quad & E\left[\overline{D_R^i}\right] \leq D_{\max}, \\ & \sum_{n=1}^{N^i} M_n^i \leq M_{\max}^i, \end{aligned} \tag{6}$$

where the relationship between the packet loss probability p_n^i and power P_n^i is given by (1). This is a nonlinear mixed-integer problem. We could solve this problem by finding P_n^i's that minimize E_{tot}^i for each feasible combination of M_n^i's, and then choosing the combination of P_n^i's and M_n^i's that yields the minimal E_{tot}^i. However, this would be prohibitively expensive in terms of computation. To simplify, we decompose this problem into a sequence of two simpler problems, as in [7]. The first step is to find intermediate values of P_n^i (call them $P_{n,opt}^i$) that minimize the total transmission energy for $M_n^i = 1, n = 1, 2, ..., N^i$, subject to $E\left[\overline{D_R^i}\right] \leq D_{\max}$. This is a constrained nonlinear programming problem which we solve via sequential quadratic programming (SQP) [8]. Once $P_{n,opt}^i$ are found, we can compute the corresponding packet loss probabilities $p_{n,opt}^i$ from (1). The next step is to find the combination of M_n^i's that minimizes the total transmission energy, while maintaining the values of $p_{n,opt}^i$. Note that if packet n is repeated M_n^i times using the transmission power

$$P_n^i(M_n^i) = G\left[\frac{1}{2\left\{1 - [1-(p_{n,opt}^i)^{1/M_n^i}]^{\eta B_p}\right\}} - 1\right], \tag{7}$$

then each of it's M_n^i copies will have a loss probability of $\left(p_{n,opt}^i\right)^{1/M_n^i}$, but the effective loss probability of packet n remains $\left(\left(p_{n,opt}^i\right)^{1/M_n^i}\right)^{M_n^i} = p_{n,opt}^i$. Hence, equation (7) tells us how to determine the power level for each of the M_n^i copies of packet n in order to preserve its loss probability of $p_{n,opt}^i$. The energy used to transmit all of M_n^i copies of packet n is $M_n^i \eta B_p P_n^i(M_n^i)/R_t$. The number of feasible combinations of M_n^i's is effectively the number of terms in the multinomial expansion of the sum of N^i variables raised to the power $M_{\max}^i - N^i$, given by

$$\binom{M_{\max}^i - 1}{M_{\max}^i - N^i}.$$

For the parameters in Table II, which we believe are realistic for wireless video multicast over 3G networks, this number works out to be at most 924. Hence, we used exhaustive search to

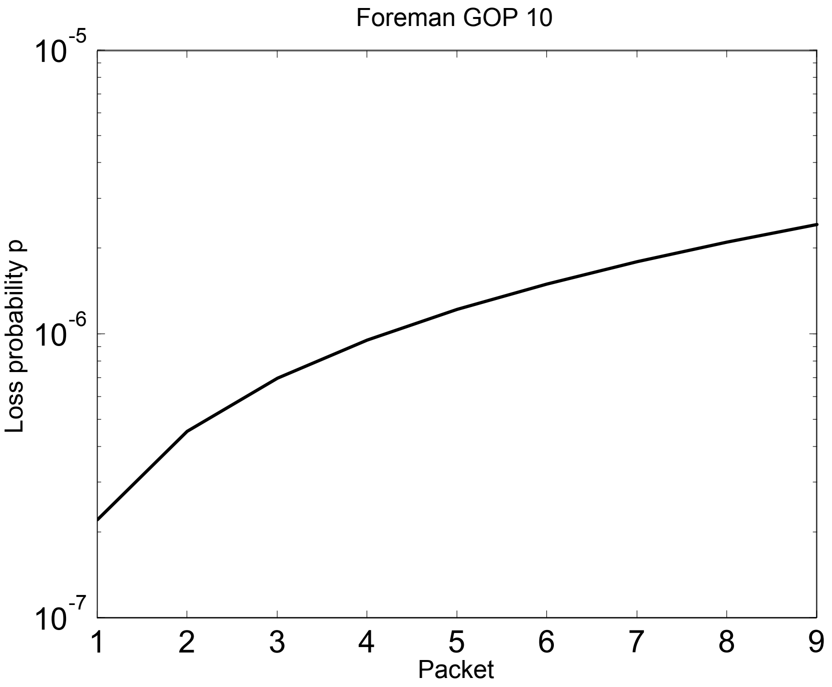

Fig. 4. Example of unequal packet loss probabilities for GOP 10 of *Foreman* with $D_{\max} = 65.025$, corresponding to minimum PSNR of 30.0 dB.

find the combination of M_n^i's that results in the lowest transmission energy.

The above procedure assigns power level P_n^i for the n-th packet in GOP i. This power level corresponds to packet loss probability p_n^i which is related to P_n^i through (1). An example of loss probabilities assigned to the packets of GOP 10 of the *Foreman* sequence, with $D_{\max} = 65.025$ (equivalent to minimum PSNR of 30.0 dB), is given in Fig. 4. Observe that packets are assigned unequal loss probabilities - packets near the start of the bitstream have lower loss probabilities than those near the end of the bitstream, just as it should be for a scalable bitstream.

B. Loss followed by transmission of parity packets

Now assume that packets from GOP $i-1$ have been sent, and that some of them are lost. Each user may have lost a certain number of packets. Let U be the number of users in the system, and let Λ_j^{i-1} be the number of packets from GOP $i-1$ that user j has lost. Let Λ^{i-1} be the maximum of these numbers, so that $\Lambda^{i-1} = \max\{\Lambda_j^{i-1} : j = 1, 2, ..., U\}$. In order to recover the losses in GOP $i-1$, the server will transmit Λ^{i-1} parity packets using some of the slots allocated to GOP i. This number of parity packets will enable each user to correct its own losses, since it is large enough even for the user with the largest loss. The goal is to minimize the total energy spent on transmission of packets of GOP i and transmission of parity packets for GOP $i-1$, while satisfying distortion constraint in both GOP i and GOP $i-1$.

Since different users are likely to lose different packets, each of them could have a different expression for the expected distortion in GOP $i-1$ in terms of parity packets' loss probabilities. The server could compute each of those, and then choose power levels for parity packets to satisfy the distortion constraint for each user separately. However, with a large number of users, this would be prohibitively expensive. Hence, we adopt a worst-case design strategy to find a single expression for $E\left[\overline{D_R^{i-1}}\right]$, which in turn will result in distortion constraint being satisfied for all users. Let $l_1, l_2, ..., l_L$ be the sequence of indices of packets lost among all users, sorted in ascending order. For packetized SNR scalable bitstreams, lower packet index indicates higher importance, because decoding of a packet depends on successful decoding of all previous packets. The worst-case design corresponds to the situation where the user with the largest loss (call her user A) is also the user which has lost the most important packet l_1. Note that $\Lambda^{i-1} \leq L$, because no single user could have lost more packets than the number of packets lost among all users. The server needs to send Λ^{i-1} parity packets. These packets will all be sent at the same transmission power level P_p^{i-1}. Let p_p^{i-1} be the corresponding loss probability for each of them. If user A receives all of the parity packets, she will be able to decode $N^{i-1}B_p$ bits in GOP $i-1$, otherwise she will decode $(l_1-1)B_p$ bits. The expression for expected distortion of GOP $i-1$ for user A is

$$\begin{aligned} E\left[\overline{D_R^{i-1}}\right] &= \left[1-(1-p_p^{i-1})^{\Lambda^{i-1}}\right]\overline{D^{i-1}}\left((l_1-1)B_p\right) \\ &\quad +(1-p_p^{i-1})^{\Lambda^{i-1}}\overline{D^{i-1}}\left(N^{i-1}B_p\right). \end{aligned} \tag{8}$$

Again, if j-th parity packet is repeated M_j^{i-1} times, its power level is chosen according to (7) to keep the loss probability the same. The optimization problem now has the following form:

$$\begin{aligned} \text{minimize} \quad & E_{\text{tot}} = E^{i-1} + E^i_{\text{tot}}, \\ \text{subject to} \quad & E\left[\overline{D_R^{i-1}}\right] \leq D_{\max}, \\ & E\left[\overline{D_R^{i}}\right] \leq D_{\max}, \\ & \sum_{j=1}^{\Lambda^{i-1}} M_j^{i-1} + \sum_{n=1}^{N^i} M_n^i \leq M^i_{\max} \end{aligned} \tag{9}$$

where E^{i-1} is the total energy spent on transmission of parity packets for GOP $i-1$, given by $E^{i-1} = (\eta B_p P_p^{i-1}/R_t) \sum_{j=1}^{\Lambda^{i-1}} M_j^{i-1}$, E_{tot}^i is the total energy spent on transmission of packets from GOP i, as in (6), and $E\left[\overline{D_R^i}\right]$ is the expected distortion in GOP i given by (5). As in the previous subsection, the first step is to find intermediate power levels $P_{p,opt}^{i-1}$ and $P_{n,opt}^i$ (and corresponding loss probabilities $p_{p,opt}^{i-1}$ and $p_{n,opt}^i$) that minimize the total transmission energy for $M_j^{i-1} = 1, j = 1, 2, ..., \Lambda^{i-1}$ and $M_n^i = 1, n = 1, 2, ..., N^i$. Then we find the best combination of M_j^{i-1} and M_n^i that minimizes the total transmission energy while maintaining loss probabilities found in the previous step.

C. Loss followed by error concealment

Noncausal error control (NCEC) operates as follows. Losses in GOP $i-1$ will be concealed from GOP i using backward error concealment. However, to satisfy distortion constraint in GOP $i-1$ in this way, one typically needs to transmit more data from GOP i than would be necessary in the absence of losses in GOP $i-1$. Suppose there are U users in the video multicast system. Each user may have lost a certain number of packets and, depending on its loss pattern, is able to decode a certain number of received packets. Suppose that user j is able to decode $N_{j,i-1}$ packets in GOP $i-1$. Similarly to the T2HA/F case, we adopt the worst-case design strategy. Based on feedback information, the server determines the largest number of packets that can be decoded by all users, which is equal to the number of decodable packets for the user with fewest decodable packets. Call this number N_{i-1}, so that $N_{i-1} = \min\{N_{j,i-1} : j = 1, 2, ..., U\}$. Using this number and the appropriate D-R concealment surface (such as the one shown in Fig. 2), the server can determine the number of packets needed to satisfy the distortion constraint in GOP $i-1$ through error concealment.

The following example based on the D-R concealment surface from Fig. 2 illustrates the procedure. Assume that in GOP $i-1$, Users $1, 2$, and 3 have received $3, 5$, and 8 packets, respectively. To find out how the concealed video quality for the three users depends on the number of packets in GOP i, we find intersections of the surface in Fig. 2 with planes $N_{1,i-1} = 3, N_{2,i-1} = 5$, and $N_{3,i-1} = 8$, respectively. These intersections define operational D-R concealment curves for each user, shown in Fig. 5. Suppose our goal is to maintain the maximum distortion below $D_{\max} = 120$ in terms of MSE. In that case, 5 packets from GOP i would be sufficient for User 1, but not for Users 2 and 3 - they would need 6 packets. Hence, to satisfy distortion constraint

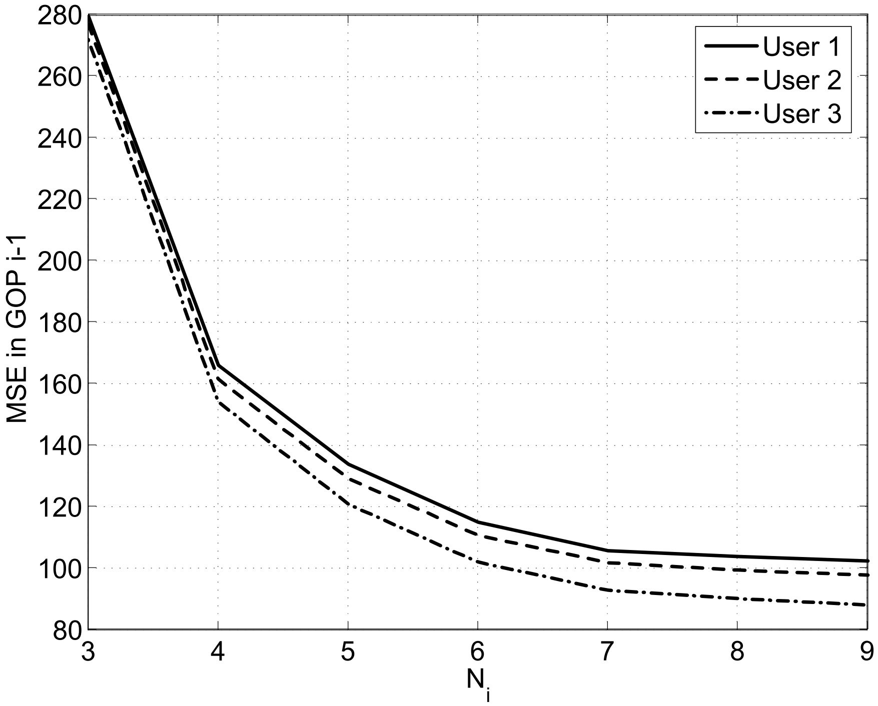

Fig. 5. MSE in GOP $i-1$ versus the number of packets in GOP i for three users, when backward error concealment is used.

in GOP $i-1$ for all users, we need at least 6 packets from GOP i in this case. This is also the worst-case design, because it is dominated by the user with the fewest decodable packets. Let the minimum number of packets from GOP i needed to satisfy distortion constraint in GOP $i-1$ for all users through error concealment be denoted by $N^i_{c,\min}$.

In addition to $N^i_{c,\min}$, the server can also determine N^i, the number of packets needed to satisfy the distortion constraint in GOP i (based on (3)). Let $N^i_c = \max(N^i_{c,\min}, N^i)$. This number of packets from GOP i will be sent by the server to the users. Since N^i is the greater of the two numbers ($N^i_{c,\min}$ and N^i), it will enable each user to satisfy distortion requirements in both GOP i and GOP $i-1$. Strictly speaking, this conclusion relies on the assumed monotonicity of operational D-R curves $\overline{D^i}(\cdot)$ and operational D-R concealment curves such as those shown in Fig. 5. It is well-known that theoretical D-R functions are non-increasing [15], but there is no similar result for operational D-R curves which depend on the combination of a particular codec and specific source information [16]. Nonetheless, our empirical observations show that this monotonicity is indeed satisfied for the MC-EZBC coder over the range of coding rates and sequences used in this work, and we expect it to hold for other SNR scalable coders as well.

The optimization problem now becomes

$$\begin{aligned} \text{minimize} \quad & E_{\text{tot}}^i = \sum_{n=1}^{N_c^i} \frac{M_n^i \eta B_p}{R_t} P_n^i, \\ \text{subject to} \quad & E\left[\overline{D_R^{i-1}}\right] \leq D_{\max}, \\ & E\left[\overline{D_R^{i}}\right] \leq D_{\max}, \\ & \sum_{n=1}^{N_c^i} M_n^i \leq M_{\max}^i, \end{aligned} \tag{10}$$

where $E\left[\overline{D_R^i}\right]$ is computed as in (5), and $E\left[\overline{D_R^{i-1}}\right]$ is computed using the appropriate operational D-R concealment surface $\overline{D_c^{i-1}}(\cdot,\cdot)$ as

$$\begin{aligned} E\left[\overline{D_R^{i-1}}\right] &= \sum_{n=1}^{N_c^i} \prod_{j=1}^{n-1} \left(1-(p_j^i)^{M_j^i}\right) \left(p_n^i\right)^{M_n^i} \overline{D_c^{i-1}}(N_{i-1}B_p,(n-1)B_p) \\ &\quad + \prod_{j=1}^{N_c^i} \left(1-(p_j^i)^{M_j^i}\right) \overline{D_c^{i-1}}(N_{i-1}B_p, N_c^i B_p). \end{aligned} \tag{11}$$

As before, we first find the power levels (and corresponding loss probabilities) that minimize the total transmission energy for $M_n^i = 1, n = 1, 2, ..., N_c^i$, and then find the combination of M_n^i that minimizes the total transmission energy for the loss probabilities computed in the precious step.

If the total energy obtained by solving (10) happens to be higher than the energy required to employ T2HA/F (obtained by solving (9)), the server will transmit parity packets; otherwise, it will send only the packets from GOP i and rely on error concealment. Hence, NCEC will be at least as efficient as T2HA/F, because it can always resort to T2HA/F if that happens to be the more efficient option. The question is - does error concealment ever become the more efficient option? The following section shows that error concealment usually is the more efficient option. Consequently, NCEC can be significantly more energy efficient than T2HA/F.

IV. Results and discussion

In the absence of constraints on the received video quality, error concealment would always be more efficient than the transmission of incremental redundancy. However, this is no longer the case if the received video quality is required to be above a certain level. Most error concealment methods cannot guarantee the quality of recovered video. One of the benefits of the error concealment method we employed in this work is that the quality of reconstructed video can be predicted using D-R concealment surfaces (e.g., Fig. 2). This enables future video data to be adjusted so that concealment can produce the required quality level. However, if the scene or motion are too complex, or if the required quality is too high, error concealment might not be able to provide it. Further, even if concealment with sufficient quality is possible, the quality of the reference data (the following GOP), which are used as the basis for concealment, may need to be so high (i.e. may require so many bits) that the transmission of incremental redundancy becomes the more efficient option. Hence, it is necessary to compare the energy required by T2HA/F and NCEC to maintain the received video quality above a certain pre-specified level.

We used two standard sequences in our experiments - *Claire* and *Foreman* - both grayscale, QCIF resolution, at 30 fps. They were encoded using the MC-EZBC video coder with GOP size of 8 frames. We simulated wireless multicast delivery of these sequences to three users. The $N_0/E[H]$ channel parameters for the three users were set as $1.2 \times 10^{-6}, 0.6 \times 10^{-6}$ and 0.24×10^{-6} W/Hz, respectively, similar to the values used in [17]. These parameters are used to compute packet loss realization for the three users. They correspond to the situation where, on average, the first user has the worst reception, while the third user has the best reception. The value of 1.2×10^{-6} is used in solving (6), (9), and (10). Hence, the average energy consumption is dominated by the user with the worst reception. However, the actual energy consumption for each particular GOP depends on the actual loss realization for each of the users in the previous GOP.

Multicast simulation is carried out in the following way. Each packet gets assigned a certain transmission power through optimization procedures described in the previous section. For each user, packet loss probability can be obtained from transmission power via (1), where constant G involves $N_0/E[H]$ parameter for that user. Hence, each user ends up with its own packet loss probability. Then, for each user, we generate a Bernoulli random variable with the corresponding success probability to indicate whether the packet has been lost or received.

Transmission rate R_t of the broadcast channel was varied in the range between 61.4 and 122.9 kbps for *Claire*, and between 92.2 and 215.0 kbps for *Foreman*. The reason for the difference in the range of transmission rates is that *Foreman* contains higher motion and is harder to compress (i.e. needs more bits) than *Claire*, so higher transmission rate is needed for the same quality. We tested error control with three values of maximum allowed distortion in terms of mean square error, $D_{\max} \in \{205.6271, 115.6326, 65.0250\}$. These settings correspond to minimum PSNR of 25.0 dB, 27.5 dB and 30.0 dB, respectively. To obtain statistically meaningful results, each sequence was transmitted 1000 times for each value of R_t and each $D_{\max}$. The results in Tables III and IV represent averages of these 1000 simulation runs.

The average energy per frame spent on error control by T2HA/F and NCEC is shown in Table III for *Claire* and Table IV for *Foreman*. Observe that at lower transmission rates, NCEC significantly outperforms T2HA/F in terms of energy efficiency. For example, for *Claire* at $R_t = 61.4$ kbps with minimum PSNR of 30.0 dB, NCEC is 55% more efficient than T2HA/F. At higher transmission rates, the number of available packet slots per GOP ($M^i_{\max}$ in the previous section) increases. Hence, packets can be repeated more often within the time slot allocated to the GOP, which reduces the required transmission energy. This makes both schemes more efficient, and the advantage of NCEC over T2HA/F reduces. The same trend was observed in the case of unicast video streaming [7].

Note that for *Foreman* with minimum required PSNR of 30.0 dB, the advantage of NCEC over T2HA/F is 0%. This means that incremental redundancy always ends up being more efficient than error concealment. As mentioned above, there could be several reasons for this. In some cases, due to high scene or motion complexity, error concealment cannot provide the required quality. In other cases, the quality of future data needed for high-quality error concealment may need to be too high and require too many bits, making T2HA/F the more efficient option. We would expect both of these situations to arise in sequences with complex motion. Our results agree with this observation.

To gain further insight into the operational differences between T2HA/F and NCEC, we extracted a sample simulation run from the experiments with *Foreman* sequence, with $R_t = 184.3$ kbps and minimum PSNR requirement of 27.5 dB. In this simulation run, user 1 loses the last two packets out of the four packets sent in GOP $\#2$ (frames $9-16$). During the transmission of the following GOP (GOP $\#3$), T2HA/F scheme sends four packets from GOP $\#3$ (which is enough to provide PSNR above 27.5 dB in GOP $\#3$) and two parity packets for loss recovery in GOP $\#2$. These two parity packets end up correcting the loss for user 1 in GOP $\#2$. On the other hand, NCEC scheme increases the quality of data in GOP $\#3$ by sending five packets

R_t (kbps)	PSNR$_{\min}$ (dB)	T2HA/F	NCEC	Gain
61.4	25.0	0.2354	0.1470	37.5%
76.8	25.0	0.0781	0.0545	30.2%
92.2	25.0	0.0390	0.0328	15.9%
107.5	25.0	0.0267	0.0242	9.4%
122.9	25.0	0.0197	0.0183	7.1%
61.4	27.5	0.4363	0.2302	47.2%
76.8	27.5	0.1473	0.1159	21.3%
92.2	27.5	0.0618	0.0507	18.0%
107.5	27.5	0.0398	0.0330	17.0%
122.9	27.5	0.0268	0.0238	11.2%
61.4	30.0	2.3149	1.0363	55.2%
76.8	30.0	0.3037	0.1802	40.7%
92.2	30.0	0.1552	0.1130	27.2%
107.5	30.0	0.0887	0.0778	12.3%
122.9	30.0	0.0537	0.0484	9.9%

TABLE III

AVERAGE ENERGY IN JOULES PER FRAME SPENT ON ERROR CONTROL BY TYPE 2 HYBRID ARQ/FEC (T2HA/F) AND NON-CAUSAL ERROR CONTROL (NCEC) FOR *Claire*.

for GOP #3. This is more than enough to keep the minimum PSNR of the frames in GOP #3 above 27.5 dB, but it also enables error concealment to produce the frames in GOP #2 whose PSNR is above 27.5 dB. Table V summarizes packet reception/loss for the three T2HA/F users and the three NCEC users. In the table, 1 indicates packet reception while 0 indicates packet loss. Packets 5 and 6 in GOP #3 for T2HA/F users are parity packets, indicated by (P).

The PSNR profile for all users is shown in Fig. 6. We can make three observations from this example:

- Both schemes kept the PSNR of all frames above 27.5 dB, which was the main objective.
- T2HA/F ended up sending six packets in the time slot allocated to GOP #3 (four packets from GOP #3 and two parity packets for GOP #2), while NCEC sent only five packets (all from GOP #3). Hence, packets could be repeated more often with NCEC than with T2HA/F, which improved energy efficiency of NCEC. The total transmission energy in GOP

R_t (kbps)	PSNR$_{\min}$ (dB)	T2HA/F	NCEC	Gain
92.2	25.0	26.2425	15.4232	41.2%
107.5	25.0	4.8062	3.1539	34.4%
122.9	25.0	1.3646	1.1519	15.6%
138.2	25.0	1.0098	0.9207	8.8%
153.6	25.0	0.2184	0.2085	4.5%
122.9	27.5	32.8405	18.3503	44.1%
138.2	27.5	19.4350	13.3550	31.3%
153.6	27.5	4.7485	3.8113	19.7%
169.0	27.5	3.7278	3.3077	11.3%
184.3	27.5	1.1532	1.0854	5.9%
153.6	30.0	922.2730	922.2730	0.0%
169.0	30.0	831.8051	831.8051	0.0%
184.3	30.0	326.7796	326.7796	0.0%
199.7	30.0	181.0173	181.0173	0.0%
215.0	30.0	31.8804	31.8804	0.0%

TABLE IV

Average energy in Joules per frame spent on error control by Type 2 Hybrid ARQ/FEC (T2HA/F) and non-causal error control (NCEC) for *Foreman*.

#3 was 0.0794 Joules per frame (Jpf) for T2HA/F, and 0.0749 Jpf for NCEC, as illustrated in Fig. 7. This represents a saving of 5.6%, slightly lower than the average saving of 5.9% measured in 1000 simulation runs under these conditions (Table IV).

- NCEC reaction to loss in GOP #2 was to improve the quality of GOP #3 (frames $17-24$). This benefited all users, since each of them decoded GOP #3 at a higher quality, while user 1 was able to compensate for its loss in GOP #2 through error concealment. Meanwhile, with T2HA/F, sending parity packets helped user 1 correct its losses, but did not help any of the other users.

Sample frames from this simulation run are shown in Fig. 8. From top to bottom, the figure shows the first and the last frame from GOP #2 (frames 9 and 16) and the first frame of GOP #3 (frame 17). From left to right, the first column corresponds to T2HA/F (all three users), the second column corresponds to users 2 and 3 with NCEC, while the third column shows the

	GOP #2				GOP #3					
Packet #	1	2	3	4	1	2	3	4	5 (P)	6 (P)
T2HA/F 1	1	1	0	0	1	1	1	1	1	1
T2HA/F 2	1	1	1	1	1	1	1	1	1	1
T2HA/F 3	1	1	1	1	1	1	1	1	1	1
Packet #	1	2	3	4	1	2	3	4	5	–
NCEC 1	1	1	0	0	1	1	1	1	1	–
NCEC 2	1	1	1	1	1	1	1	1	1	–
NCEC 3	1	1	1	1	1	1	1	1	1	–

TABLE V

A sample simulation run: 1 indicates packet reception, 0 indicates packet loss, (P) indicates parity packet.

frames decoded by user 1 with NCEC. Observe that the quality of frame 17 (bottom row in the figure) is higher for NCEC users than for T2HA/F users, since NCEC increases the quality of GOP #3 to compensate for the loss in GOP #2. The quality improvement is most visible along diagonal edges in the upper-left and upper-right part of the frame. PSNR of frame 17 is 29.5 dB for T2HA/F users and 31.3 dB for NCEC users. The first two rows in the figure show the first and the last frame of GOP #2 (frames 9 and 16) Here, users 2 and 3 with NCEC decode the same frames as T2HA/F users - all of them are able to decode the original four packets from GOP #2. The PSNR values for these two frames are 29.4 dB and 29.6 dB, respectively. Meanwhile, user 1 with NCEC (third column) uses error concealment to compensate for its loss. Observe how the increased quality of GOP #3 "spills over" into GOP #2 for this user through error concealment. The PSNR values for frames 9 and 16 for this user are 29.0 dB and 30.7 dB. As expected, frame 16 has the highest quality among all frames in GOP #2 because it is temporally closest to high-quality GOP #3 which is being used as the reference for error concealment. As we move backwards in time through GOP #2, away from GOP #3 and towards frame 9, PSNR values decrease, as illustrated in Figure 6. In terms of PSNR, frame 9 of NCEC user 1 has lower quality than frame 9 of other users - 29.0 dB versus 29.4 dB. However, visual comparison still seems to favor frame 9 of NCEC user 1 despite the lower PSNR value, because diagonal edges look much cleaner here than in other users' frames, albeit with some minor motion artifacts.

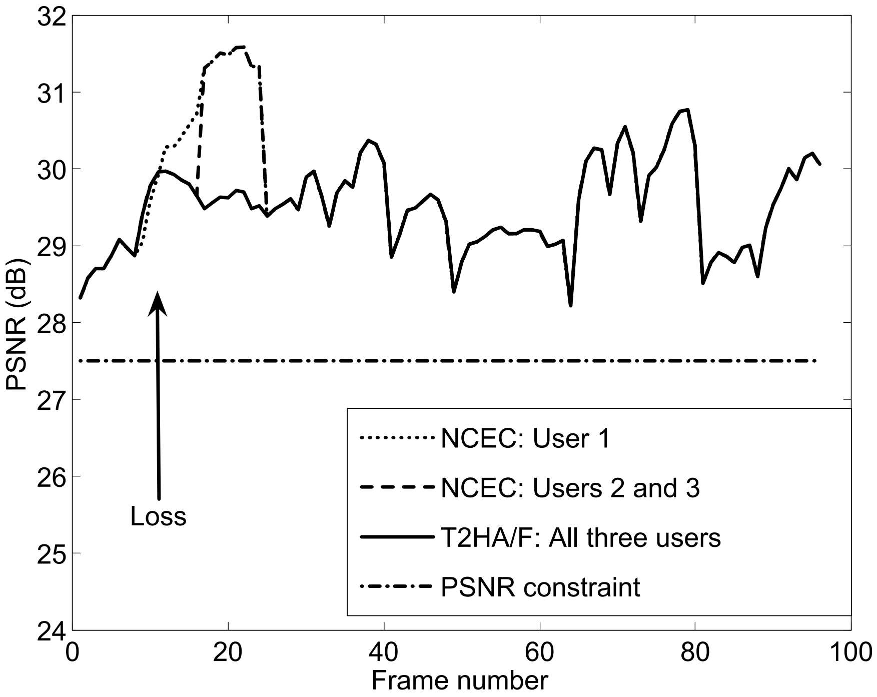

Fig. 6. PSNR in dB for *Foreman* sequence from one of the simulation runs.

V. Conclusions and future work

The focus of this paper was wireless video multicast. We have:

1) Developed joint source rate selection and power management for two error control schemes - type 2 hybrid ARQ/FEC (T2HA/F) and noncausal error control (NCEC).
2) Compared energy consumption of T2HA/F and NCEC in maintaining pre-specified video quality at the receiver for two standard sequences of different complexities and a variety of transmission rates.
3) Examined operational differences of the two error control schemes, and demonstrated that NCEC can be significantly more efficient than T2HA/F, especially for sequences with lower motion complexity.

T2HA/F can scale to a very large number of users [6], because it enables each user to correct its own losses from common data - parity packets in this case. NCEC shares this property with T2HA/F. With NCEC, losses are corrected from future video data, which is common to all users. However, as the results show, NCEC can be significantly more efficient than T2HA/F. The main

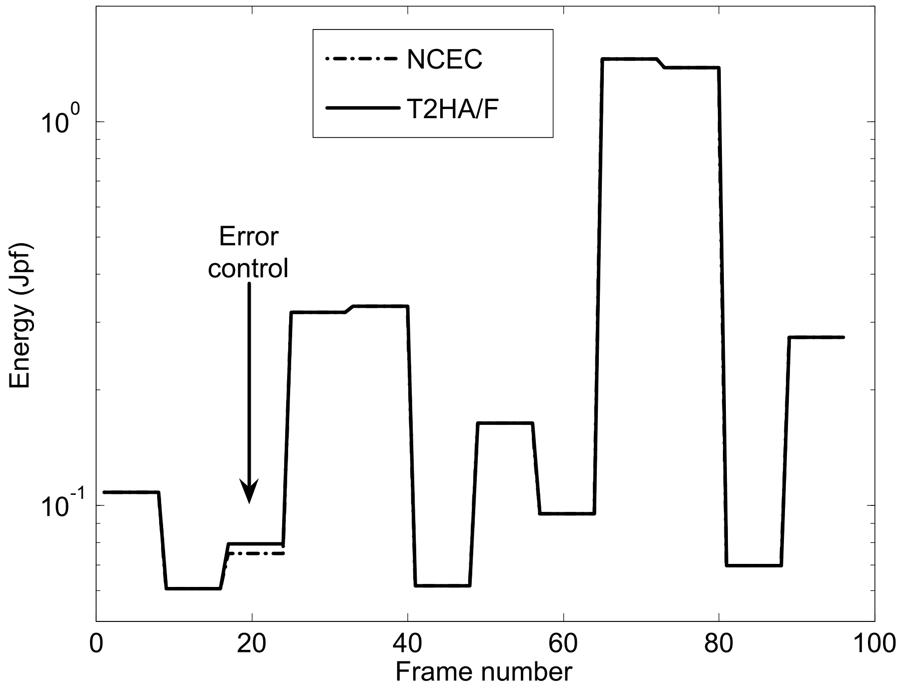

Fig. 7. Energy in Joules per frame (Jpf) corresponding to Fig. 6.

reason for this efficiency gain is that with NCEC, all resources (e.g. bandwidth, power) can be spent on transmitting future data, while with T2HA/F, part of the resources have to be spent on sending parity packets related to the previously transmitted data.

The two error control schemes we described in this paper were designed to minimize the power needed to maintain the received video quality above a certain pre-specified level. In practice, however, there are various regulatory limits on the base-station transmit power [18]. Hence, the power levels resulting from optimization procedures developed in Section III might occasionally be too high for a practical system. An important extension of this work would be the development of power management schemes for error control that incorporate power constraints. In this extended framework, the system would still try to minimize the power needed to maintain a certain video quality as described in Section III. If the resulting required power happens to be above the maximum allowed power level, the system would switch to a different mode and try to maximize the video quality for the given maximum allowed power level, as suggested in [19].

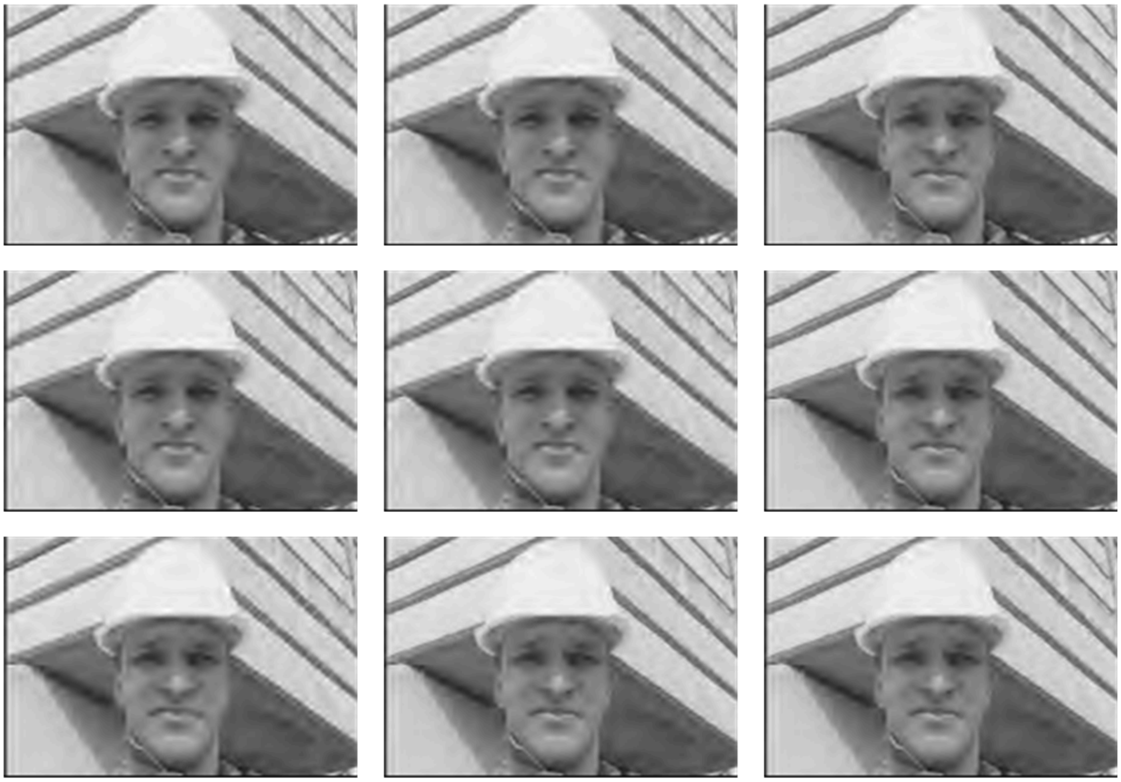

Fig. 8. Sample frames from the simulation shown in Fig. 6. Left to right: T2HAF (all users), NCEC (users 2 and 3), NCEC (user 1). Top to bottom: frames $9, 16$, and 17.

The key component of NCEC is the error concealment algorithm that recovers lost data from future video data. Coupled with the scalable coder, this methodology allows us to predict the video quality obtained through error concealment, and adjust the quality of future video data accordingly. More advanced error concealment algorithms in NCEC could further reduce energy consumption.

References

[1] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communications*, Upper Saddle River, NJ: Prentice-Hall, 2002.

[2] J. G. Apostolopoulos and M. D. Trott, “Path diversity for enhanced media streaming,” *IEEE Commun. Mag.*, vol. 42, no. 8, pp. 80-87, Aug. 2004.

[3] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, “Low bit-rate video transmission over fading channels for wireless microcellular systems,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 1-11, Feb. 1996.

[4] B. Girod and N. Färber, “Feedback-based error control for mobile video transmission,” *Proc. IEEE*, vol. 87, no. 10, pp. 1707-1723, Oct. 1999.

[5] I. V. Bajić and J. W. Woods, “Error concealment for scalable motion-compensated subband/wavelet video coders,” submitted for publication. [Online]. Available: www.sfu.ca/~ibajic/bw_sub_mcec.pdf

[6] M. S. Lacher, J. Nonnenmacher, and E. W. Biersack, “Performance comparison of centralized versus distributed error recovery for reliable multicast,” *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 224-238, Apr. 2000.

[7] I. V. Bajić, “Noncausal error control for video streaming over wireless packet networks,” to appear in *IEEE Trans. Multimedia*, Dec. 2007.

[8] The MathWorks, *Optimization toolbox user’s guide*, ver. 3.1, Natick, MA: The MathWorks, 2006.

[9] I. V. Bajić, “Efficient error control for wireless video multicast,” *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP’06)*, pp. 306-309, Victoria, BC, Canada, Oct. 2006.

[10] M. Hauge and Ø. Kure, “Multicast in 3G networks: Employment of existing IP multicast protocols in UMTS,” *Proc. ACM WoWMoM’02*, pp. 96-103, Sep. 2002.

[11] J. Wang, R. Sinnarajah, T. Chen, Y. Wei, and E. Tiedemann, “Broadcast and multicast services in cdma2000,” *IEEE Commun. Mag.*, vol. 42, no. 2, pp. 76-82, Feb. 2004.

[12] S.-T. Hsiang and J. W. Woods, “Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank,” *Signal Processing: Image Commun.*, vol. 16, pp. 705-724, May 2001.

[13] P. Chen and J. W. Woods, “Bi-directional MC-EZBC with lifting implementation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 10, pp. 1183-1194, Oct. 2004.

[14] S. G. Wilson, *Digital modulation and coding*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[15] T. M. Cover and J. A. Thomas, *Elements of information theory*, New York: Wiley, 1991.

[16] K. Sayood, *Introduction to data compression*, Second edition, San Francisco, CA: Morgan Kaufmann, 2000.

[17] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, “Joint source coding and transmission power management for energy efficient wireless video communications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 411-424, Jun. 2002.

[18] D. G. Steer, “Coexistence and access etiquette in the United States unlicensed PCS band,” *IEEE Personal Communications*, vol. 1, no. 4, pp. 36-43, 4th Quarter 1994.

[19] M. van der Schaar and S. Shankar N, “Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms,” *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp. 50-58, Aug. 2005.