Optimal Packet Scheduling for Wireless Video Streaming with Error-Prone Feedback

Dihong Tian, Xiaohuan Li, Ghassan Al-Regib, Yucel Altunbasak, and Joel R. Jackson
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332–0250

Abstract— **In wireless video transmission, burst packet errors generally produce more catastrophic results than equal number of isolated errors. To minimize the playback distortion, it is crucial for the sender to know the packet errors at the receiver and then optimally schedule next transmissions. Unfortunately, in practice, feedback errors result in inaccurate observations of the receiving status. In this paper, we develop an optimal scheduling framework to minimize the expected distortion by first estimating the receiving status. Then, we jointly consider the source and channel characteristics and optimally choose the packets to transmit. The optimal transmission strategy is computed through a partially observable Markov decision process. The experimental results show that the proposed framework improves the average peak signal-to-noise ratio (PSNR) by 0.6-1.3dB upon using a traditional system without packets scheduling. Moreover, we show that the proposed method smoothes out the bursty distortion periods and results in less fluctuating PSNR values.**

I. Introduction

Multimedia communications over packet networks has experienced phenomenal growth over the last decade. Recently, the continuing growth in wireless communications has attracted considerable applications as well as researches to transmit video over wireless channels. In practice, wireless video communications face several challenges such as high error rates, bandwidth variations and limitations, and processing capability constraints on the handhold devices. Among these, the unreliable and error-prone nature of the wireless channel is the major challenge to stream video over wireless channels. Wireless channels are afflicted by time-varying fading and interference conditions, which may lead to burst packet corruptions. A typical architecture of end-to-end MPEG-4 video streaming over a wireless channel is depicted in Figure 1.

Many studies have been conducted on quality of service (QoS) protection for wireless video transmission. Error control schemes using forward error correction (FEC) codes are investigated in [1], [2]. Because FEC codes are ineffective when errors are bursty, which is the case in fading wireless channels [3], there is a trend to use more intelligent automatic retransmission request (ARQ) techniques to protect the transmitted video stream. In [4], different video frame fields are protected by unequal ARQ schemes with different retransmission policies.

The streaming video client typically employs error detection and concealment techniques to mitigate the effects of corrupted packets [5]. Generally, streaming media systems do not rely on the transport-layer protocols for media transport [6]. Instead, they implement their own application-level transport methods to provide the best end-to-end delivery while adapting to the changing network conditions. Recent advances include application-layer error-correction coding [10], path diversity transmission [11], application-level packet scheduling [7], [8], and rate-distortion optimized streaming [12], [13].

A priority-based packet scheduling algorithm for wireless video streaming is proposed in [8]. In this method, video packets are scheduled for transmission based on their deadline thresholds, which are assigned to reflect the packets relative importance within the video stream based on heuristics. Nevertheless, no optimization is performed in [8].

In [12], [13], Chou et al. proposed rate-distortion optimized streaming methods. These algorithms provide a flexible framework where time and bandwidth resources are allocated among packets in order to minimize a Lagrangian cost function. This cost function depends on both transmission rate and distortion. Packetization of the media is abstracted as a single directed acyclic dependency graph and the overall distortion is computed based on such a dependency graph. And the data packet transmission processes are assumed to be independent. An *optimal transmission policy* is determined based on the channel statistics, the packets' interdependencies, and the reduction in distortion achieved by each packet if it is successfully decoded.

The framework in [12] proposes a sender-driven streaming algorithm to deliver video over a best-effort network. In wireless fading channels, however, burst block errors appear as a very important feature. The assumption in [12] that the

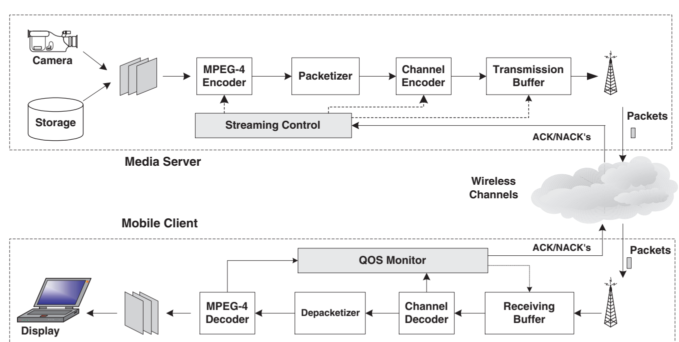

Fig. 1. An architecture of streaming MPEG-4 video sequences over wireless channels.

 0-7803-8344-3/04/$20.00

packet transmission processes are independent is no longer appropriate. Furthermore, for predictively coded video, burst packet errors generally result in larger distortion than equal number of isolated errors [14]. To minimize the reconstructed distortion, the sender should take into account not only the remaining packets but also the delivery results of passed packets and then schedule next transmissions.

In this paper, we develop an optimal packet scheduling framework for streaming video over a wireless link with error-prone feedback. A general scenario is considered where packetized video is transmitted on demand from the sender to the mobile host and the acknowledgements (ACK's or NACK's) are sent to the sender as feedback. The sender estimates the possible receiving states by a probability distribution, which is determined according to the transmission history, the observations on feedback, and the channel statistics. After that, the optimal transmission strategy for the remaining packets is computed using a partially observable Markov decision process (POMDP), following the principle of minimizing the expected distortion.

To simplify the presentation, during the analysis we assume that packet delays on the wireless link are fixed or ignorable. Nevertheless, the framework allows easy generalization to other scenarios where the random packet delay should be considered. The innovation in this work, namely, modeling the uncertainty of the receiving state to stream video optimally, can be integrated with the previous research advances such as rate-distortion optimized streaming over lossy networks [12].

The rest of this paper is organized as follows. Section II describes the channel model used in this paper. In Section III, we show the effect of packet errors on the compressed video and summarize how to estimate the distortion produced by a packet error pattern. Section IV studies in detail the optimal packet scheduling framework with a POMDP process. Some experiments are conducted on MPEG-4 video sequences and the results are presented in Section V. Section VI concludes this paper.

II. MODELING WIRELESS FADING CHANNELS

The behavior of block errors invoked on data transmitted over fading channels has been investigated in [16], [17]. A first-order Markov model, although not rigorously proven, has been found to accurately approximate the block error process in wireless channels; both for slow fading (successive samples are very correlated) and for fast fading (successive samples are almost independent).

The system parameters which affect the Markov description are defined by a transition matrix

$$M(x) = \begin{pmatrix} p(x) & q(x) \\ r(x) & s(x) \end{pmatrix} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}^x,$$

where $p(x) = 1 - q(x)$ (and similarly, $r(x) = 1 - s(x)$) is the probability that the i-th block is successfully transmitted, given that the transmission of the $(i-x)$-th block was successful (unsuccessful).

There are essentially two independent parameters to completely specify this first-order Markov model. In the literature, the transition probability $r = P\{success|failure\}$, and the steady-state packet error probability,

$$\varepsilon = 1 - \frac{r}{1-p+r},$$

are studied instead of evaluating p directly. This choice is motivated by the fact that these two parameters have an immediate physical significance. ε, as mentioned above, is the average packet error rate (PER), measuring how often a packet is corrupted and $1/r$ is the average error burst length (EBL), and gives a measure of how clustered the errors tend to be. The quantities ε and r can be either computed through analysis or evaluated directly from the second-order joint distribution computed by simulation [18].

In this paper, we consider a scenario of sender-driven transmission over the wireless channel with error-prone feedback. The forward and backward channels are modeled by two independent first-order Markov models, which are described by two transition matrices,

$$M_f = \begin{pmatrix} p_f & q_f \\ r_f & s_f \end{pmatrix} \text{ and } M_b = \begin{pmatrix} p_b & q_b \\ r_b & s_b \end{pmatrix}, \tag{1}$$

respectively. Equivalently, we represent the two channels by two pairs of parameters, namely $(\varepsilon_f, 1/r_f)$ and $(\varepsilon_b, 1/r_b)$.

III. ANALYSIS OF PACKET ERRORS FOR THE COMPRESSED VIDEO

In this section, we analyze the impact of the packet error on the reconstructed distortion of the compressed video, and introduce models to estimate the distortion associated with a packet error pattern. For simplification, in this section and throughout this paper we assume that each predictively coded frame (P-frame) is coded into a single packet, so that the corruption of a packet corresponds to the loss of an entire frame. And a simple error concealment scheme is assumed where the corrupted frame is replaced by the previous frame at the decoder output. The results and the streaming framework proposed in next section can be extended to the case where each frame is coded into multiple packets and the corruption of one packet does not result in the loss of an entire frame.

Figure 2 plots the measured distortion produced by a burst error of length two as well as two errors separated by a short lag l, $0 < l < L$, where L is the INTRA *update interval*. We show the results for the video sequences: FOREMAN, MOTHER-DAUGHTER and AKIYO, coded using MoMuSys MPEG-4 visual reference software. For each error pattern, the distortion has been averaged over all possible error realizations. All the measured distortions are normalized by the distortion of the length-two burst error. From Figure 2, one can observe that the distortion resulted from the burst packet error is larger than the distortions produced by two separated errors, and moreover, the average overall distortion of two errors with a shorter lag is larger than that of two errors with a longer lag.

In general, different packet error patterns result in unequal distortions [14]. Accurate estimation of the distortion is therefore crucial for a sender-driven transmission mechanism to

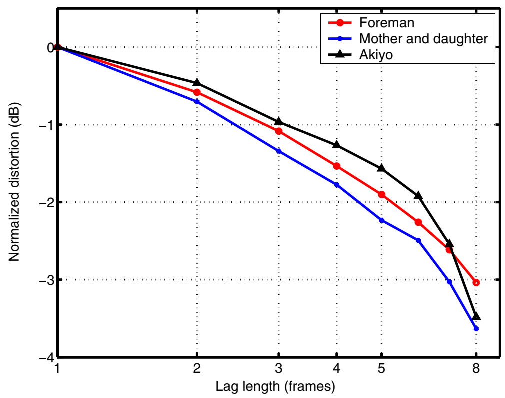

Fig. 2. Overall distortions produced by two packet errors, normalized by the overall distortion of the burst error with length two; the INTRA update interval is $L = 30$ frames.

maximize the receiving quality. Next, we first analyze the distortion of a single error by accounting for error propagation, intra refresh, and spatial filtering. We then introduce models for estimating the effect of multiple packet errors.

The original video signal is a discrete space-time signal denoted by $s[x, y, k]$, where $(x, y) \in Z^2$ is the spatial coordinate and $k \in Z$ is the temporal index. To simplify notation, the 2-D array of $M = M_1 \times M_2$ pixels in each frame k are sorted in a 1-D vector $f[k]$ (of length M) in line-scan order. We use 1-D vector $f[k]$ to represent an original video frame, $\widehat{f}[k]$ to denote the error-free reconstruction of the frame, and $g[k]$ to denote the reconstruction at the decoder after error concealment. With our error concealment assumption, the initial error frame introduced by a *single* error at frame k is

$$e_s[k] = g[k] - \widehat{f}[k] = \widehat{f}[k-1] - \widehat{f}[k],$$

which is also a 1-D vector. Note that the quantization error is not included here since our concern is the effect of channel error. Assuming the error frame $e_s[k]$ to be a zero-mean process, its variance is equal to its mean square error (MSE), given by

$$(e_s^T[k] \cdot e_s[k])/M = \sigma_s^2[k].$$

The above MSE quantifies the error power introduced in the initial error frame, but it does not include the effect of error propagation to subsequent frames. With an INTRA update period of L, if a single error is introduced at k with an MSE of $\sigma_s^2[k]$, and the power of the propagated error at $k+l$ is represented by $\sigma^2[k+l]$, then the *total distortion* resulted from this single error is

$$D_s[k] = \sum_{i=k}^{\infty} \sigma^2[i] = \sum_{l=0}^{L-1} \sigma^2[k+l]. \quad (2)$$

Note $\sigma^2[k] = \sigma_s^2[k]$ here and $\sigma^2[k+l] = 0, l > L$, since we assume that the error is completely removed by intra update after L frames.

The variance of the propagated error signal is calculated by

$$\sigma^2[k+l] = \sigma^2[k]\frac{1-\beta l}{1+\gamma l}, \quad (3)$$

for $0 \leq l < L$, where the *leakage* γ accounts for the effect of spatial filtering, and $\beta = 1/L$ is the INTRA rate [19]. Finally, the total distortion of a single error at frame k is

$$D_s[k] = \sigma_s^2[k] \sum_{l=0}^{L-1} \frac{1-\beta l}{1+\gamma l}. \quad (4)$$

Now we consider the effect of a packet error pattern with multiple errors in an INTRA update period. It can be modeled as the superposition of multiple independent errors. In other words, the total distortion of multiple errors is approximated by the sum of the distortions produced by the corresponding single error patterns, which are calculated individually using (4). A more complicated model, which takes into account the correlation between adjacent frames and therefore is more accurate especially in the situations when the packet errors occur in burst, has been proposed by Liang et al [14]. Because of the space limitation, we skip the details of this model. Interested readers are referred to [14] for further details.

IV. Packet scheduling with a partially observable Markov decision process

The above analysis shows that different packet error patterns result in unequal distortions, and the burst packet error generally produces larger distortion than equal number of isolated errors. It implies that the receiving status (packet error pattern at the receiver) should be involved in the sender's decision on transmitting the remaining packets for minimized expected distortion. Although the accurate receiving status is generally not observable due to feedback errors, the sender can maintain an estimate with a probability distribution over a set of possible states. This probability distribution, which we call *belief* throughout this paper, is obtained based on the channel statistics and the streaming history, i.e., which packets have been transmitted and what are the observations on feedback.

One can consider the *belief* as a measurement of the uncertainty of the receiving state, as illustrated in Figure 3. During the streaming process, the sender can only partially observe the status of the receiving application due to feedback corruptions. With a probability distribution, such an intractability can be quantified and used by the sender for next transmission decisions. In the remainder of this section, we study in detail, using a partially observable Markov decision process [15], how to compute the optimal transmission strategy with estimate of the receiving status.

Consider scheduling a group of packets with group size N. We define a *state* as a possible packet error pattern at the

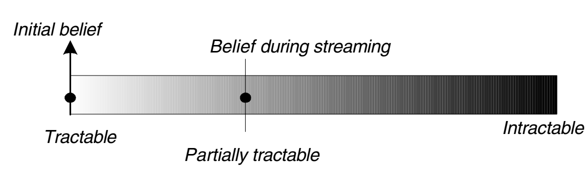

Fig. 3. *Belief*: modeling the intractability of the receiving status; the higher gray scale denotes the more uncertainty on the receiving status (more errors occurred on feedback transmission).

receiving application. And the *state space*, which contains 2^N states in total, is expressed by

$$\begin{aligned} \mathbf{S} &= \{\Phi, (1), (2), ..., (N), ..., (1, 2, ..., N)\} \\ &\triangleq \{\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{2^N-1}\}, \end{aligned} \tag{5}$$

where each state $\mathbf{s}_i$ stands for one packet error pattern by indicating which packets are missing. For instance, state $\mathbf{s}_0 = \Phi$ denotes the state that all packets have been correctly received, whereas state $\mathbf{s}_{2^N-1} = (1, 2, ..., N)$ denotes the state that none of the N packets has been correctly received. The corresponding overall distortion of state $\mathbf{s}_i$ is denoted by $D[\mathbf{s}_i]$, which can be approximated with pre-measured distortions of single errors as analyzed in Section III.

A *belief*, represented by $\mathbf{b} = [p_0, p_1, ..., p_{2^N-1}]$, is a probability distribution over the state space $\mathbf{S}$. In other words, the sender has an estimate that the receiving application is in state $\mathbf{s}_i$ with probability p_i, where $i = 0, 1, ..., 2^N - 1$. Apparently, $\sum_{i=0}^{2^N-1} p_i = 1$.

Figure 4 shows the decision trellis of a POMDP process associated with our problem. Each node represents a belief. The sender is in the initial belief at time t_0, choosing to send one of the N packets. Every time after selecting one packet and sending it, the sender makes an observation just prior to the next transmission opportunity. There are three possible observations: a positive acknowledgement (ACK), a negative acknowledgement (NACK), or neither of them (denoted by ϕ in Figure 4). Here for simplification, we have ignored the random packet delays in the channel, so that the observation ϕ is equivalent to the fact that the transmission on the backward channel is corrupted.

After transmitting a packet and making an observation, the process enters a unique new belief at next transmission opportunity. In particular, if at time t_0 the sender resides in belief $\mathbf{b}_0 = [p_0, p_1, ..., p_{2^N-1}]$, transmits packet k, and after that observes z, $z \in \{ACK, NACK, \phi\}$, then the resulted belief $\mathbf{b}' = [p'_0, p'_1, ..., p'_{2^N-1}]$ is obtained by updating the probabilities in belief $\mathbf{b}_0$ as follows. For each pair of states $\mathbf{s}_i$ and $\mathbf{s}_j$ where $k \notin \mathbf{s}_i$ and $\mathbf{s}_j = \mathbf{s}_i \cup k$, i.e., packet k is a lost packet in state $\mathbf{s}_i$ but is correctly received in state $\mathbf{s}_j$, we adjust the corresponding probabilities such that

$$p'_j = \begin{cases} p_j, & \text{if } z = NACK, \\ p_j + p_i, & \text{if } z = ACK, \quad \text{and} \\ p_j + p_i(1 - \epsilon_f(t_0)), & \text{if } z = \phi, \end{cases}$$
$$p'_i = \begin{cases} p_i, & \text{if } z = NACK, \\ 0, & \text{if } z = ACK, \\ p_i \epsilon_f(t_0), & \text{if } z = \phi, \end{cases} \tag{6}$$

where $\epsilon_f(t_0)$ is the error probability on the forward channel at time t_0, which can be calculated with the Markov transition matrix (1) and the most recent time slot whose transmission result is known to the sender (the acknowledgement has been correctly received).

To simplify the representation, we denote the above transformation by

$$\mathbf{b}' \triangleq T(\mathbf{b}_0, k, z, t_0). \tag{7}$$

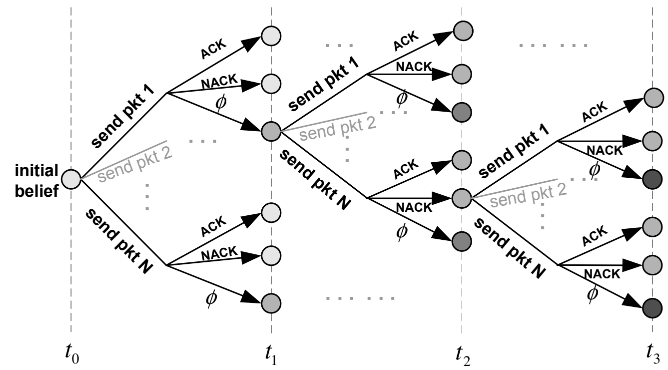

Fig. 4. Trellis for a partially observable Markov decision process. Each node stands for a *belief*, and the one with higher gray scale implies more intractability on the receiving status.

Distortion at the receiver is expected to be reduced after the sender transmits packet k in belief $\mathbf{b}_0$. The following equation calculates this expected distortion reduction.

$$\begin{aligned} E_s\{d_1(k|\mathbf{b}_0, t_0)\} &= \sum_{i=0}^{2^N-1} p_i \cdot \Delta D(k|\mathbf{s}_i, \mathbf{b}_0, t_0) \\ &= (1 - \epsilon_f(t_0)) \sum_{i:k \notin \mathbf{s}_i} p_i (D[\mathbf{s}_i \cup k] - D[\mathbf{s}_i]), \end{aligned} \tag{8}$$

where $\mathbf{s}_i \cup k$ denotes the resulted state (error pattern) if the receiver correctly receives packet k at state $\mathbf{s}_i$.

$$\Delta D(k|\mathbf{s}_i, \mathbf{b}_0, t_0) = (1 - \epsilon_f(t_0))(D[\mathbf{s}_i \cup k] - D[\mathbf{s}_i])$$

is the distortion reduction for successfully delivering packet k if the receiver is at state $\mathbf{s}_i$. Finally, $E_s\{d_1(k|\mathbf{b}_0, t_0)\}$ gives the expected distortion reduction for transmitting packet k in belief $\mathbf{b}_0$ at t_0.

The maximum expected distortion reduction for sending one packet at t_0 is simply the maximum of $E_s\{d_1(k|\mathbf{b}_0, t_0)\}$:

$$H_1(\mathbf{b}_0, t_0) = \max_{k \in \mathcal{G}(t_0)} \{E_s\{d_1(k|\mathbf{b}_0, t_0)\}\}, \tag{9}$$

where we use $\mathcal{G}(t)$ to denote the set of packets considered for transmission at time t; in particular, $\mathcal{G}(t_0) = \{1, 2, \cdots, N\}$. Correspondingly,

$$\Pi_1(\mathbf{b}_0, t_0) = \arg\max_{k \in \mathcal{G}(t_0)} \{E_s\{d_1(k|\mathbf{b}_0, t_0)\}\} \tag{10}$$

is the optimal to-be-sent packet by which the sender maximizes the expected distortion reduction in a single step.

Our general goal is to find the optimal transmission *strategy* that maximizes the expected distortion reduction in n steps. Next, we show that this n-step transmission strategy can be constructed in an inductive way.

Suppose we have constructed a function, $H_{n-1}(\mathbf{b}, t)$, which computes the maximum expected distortion reduction for an $(n-1)$-step POMDP process beginning in belief $\mathbf{b}$ and at time t. Now the question is, what is the maximum expected distortion reduction in n steps, if the sender chooses to send packet k at time t_0? This is actually the instant distortion reduction by sending packet k plus the maximum expected distortion reduction by next $(n-1)$ transmissions. That is,

$$E_s\{d_n(k|\mathbf{b}_0,t_0)\} = E_s\{d_1(k|\mathbf{b}_0,t_0)\} + \sum_{z\in\mathcal{O}} P(z,t_0)H_{n-1}(T(\mathbf{b}_0,k,z,t_0),t_1), \tag{11}$$

where $\mathcal{O} = \{NACK, ACK, \phi\}$ denotes the set of possible observations, and $P(z,t_0)$ is the probability of observing z after sending packet k at time t_0, which can be easily calculated given the Markov channel description. Recall that $T(\mathbf{b}_0,k,z,t_0)$ represents the resulted belief if the sender transmits packet k at time t_0 and observes z after that; $H_{n-1}(T(\mathbf{b}_0,k,z,t_0),t_1)$ gives the maximum expected distortion reduction that the sender could achieve by next $(n-1)$ transmissions.

Similar to (9) and (10), we obtain the maximum expected distortion reduction of the n-step POMDP process,

$$H_n(\mathbf{b}_0,t_0) = \max_{k\in\mathcal{G}(t_0)} \{E_s\{d_n(k|\mathbf{b}_0,t_0)\}\}, \tag{12}$$

and the optimal packet that should be transmitted instantly is

$$\Pi_n(\mathbf{b_0},t_0) = \arg\max_{k\in\mathcal{G}(t_0)} \{E_s\{d_n(k|\mathbf{b}_0,t_0)\}. \tag{13}$$

In summary, an n-step POMDP process can be accomplished using a recursive or a dynamic programming algorithm. The optimal transmission strategy computed by this process contains an instant to-be-sent packet at t_0, which is given by (13), and $(n-1)$ future transmissions. These sequenced transmissions will be finally determined each at a transmission opportunity, according to the previous transmission and the corresponding observation.

For a group of packets, the above scheduling framework is optimal in the sense of minimizing the final distortion. Nevertheless, this process has high computing complexity when the group size is large. Moreover, for streaming a video presentation, it is neither feasible nor practical to schedule the whole packet sequence at one time.

Finding heuristics to simplify the decision trellis or algorithms to speed up the computation is currently under research. In this paper, we implement a *sliding-window* based algorithm to investigate the performance of the proposed framework. This algorithm utilizes a packet window sliding along the time horizon. At each transmission opportunity, only the packets within the window will be scheduled for transmission.

V. Experimental results

This section presents the experimental results we obtained. We conduct experiments on two standard video test sequences, Foreman and Coastguard, coded using MoMuSys MPEG-4 visual reference software. These video sequences are in the format 4:2:0, 176×144 pixels per frame and 30 frames per second. Each video sequence is repeated to 120 seconds long (12×300 frames) and is transmitted over the simulated network with time-varying channel conditions. Four 30-second long channel phases are experienced. The detailed information of each channel phase is listed in Table I. The size of the packet window is fixed to $N=5$, i.e., five packets are scheduled at each transmission opportunity.

TABLE I
The parameters for first-order Markov channel model in the simulation.

	Forward		Backward	
	PER (ϵ_f)	EBL ($1/r_f$)	PER (ϵ_b)	EBL ($1/r_b$)
Phase 1	5%	2.5	15%	4.0
Phase 2	10%	3.0	10%	3.0
Phase 3	15%	4.0	5%	2.5
Phase 4	20%	4.0	20%	4.0

TABLE II
Performance comparison between the heuristic method and the POMDP-based framework for two video sequences.

		Phase 1	Phase 2	Phase 3	Phase 4
		Foreman			
PSNR (dB)	Heuristic	30.75	30.57	30.41	29.77
	Proposed	31.34	31.22	31.19	31.03
Variance (σ^2)	Heuristic	5.09	5.82	6.54	7.73
	Proposed	1.03	1.69	1.09	1.31
		Coastguard			
PSNR (dB)	Heuristic	29.03	28.70	28.43	27.89
	Proposed	29.62	29.27	29.31	28.95
Variance (σ^2)	Heuristic	4.15	4.91	7.22	8.87
	Proposed	0.13	1.25	0.96	2.33

We compare the proposed framework with the conventional heuristic method, which simply transmits the packets in the same order as they will be displayed at the receiving application. We assume such a scenario where each packet can have two transmission opportunities at most. Two quantities are used to perform objective comparisons: Average peak signal-to-noise ratio (PSNR), and the variance of PSNR among INTRA update intervals.

The simulated results for two video sequences are listed in Table II. We observe that the proposed framework outperforms the heuristic method by about 0.6-1.3 dB on the average PSNR, which shows that the POMDP-based scheduling can effectively reduce the end-to-end distortion. Performance gains are especially significant when the channel is experiencing very bad condition. In the case that both the forward and backward channels have high error rates, the average PSNR of the heuristic method is dramatically decreased, whereas the proposed framework only slightly degrades.

Figures 5 and 6 plot for Foreman the average PSNR values of different INTRA update periods and the frame loss ratios when the channel is in phase 1 and phase 4, respectively. We can see the improvement on the average PSNR clearly from Figure 5 (a) and Figure 6 (a). Moreover, Figure 5 (b) and Figure 6 (b) show that scheduling leads to much lower loss rates for earlier frames whereas much higher loss rates for latter frames. This is because the earlier frames in an INTRA update period are generally more important than the latter ones and will produce larger distortion if they are lost.

Another observation from Table II is that with the proposed framework, the variances of PSNR among different INTRA update periods are much smaller than the corresponding results of the heuristic method. In other words, our scheduling scheme has an effect of smoothing out the bursty distortion periods, which results in less fluctuating PSNR values compared to the heuristic method. This smoothing effect, also shown in Figure 5 (a) and Figure 6 (a), provides much better visual quality than the conventional system because for most human observers, distortion bursts result in more severe visual degradation than random distortions.

VI. Conclusions and Future Work

The problem of streaming video over wireless channels with error-prone feedback is addressed in this paper. We show that the sender's optimal transmission decision depends on the receiving state. An optimal packet scheduling framework is proposed based on a partially observable Markov decision process (POMDP). The experimental results on MPEG-4 video sequences verify the significant improvement of the proposed method on the video quality.

The framework proposed in this paper is analyzed in detail for a wireless scenario, where for simplification the packet delay is not considered. Nevertheless, the framework can be further generalized to other wireless or wired network scenarios where the random packet delay should be taken into account. We expect further research will focus on finding effective heuristics and fast algorithms to simplify the computation of the decision process.

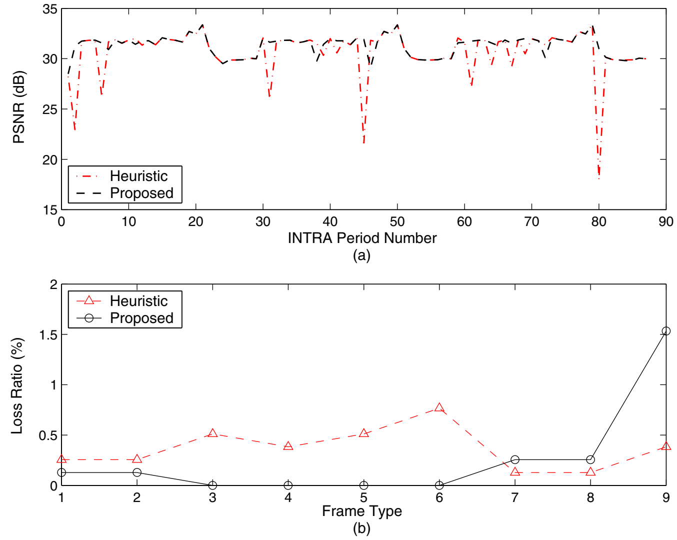

Fig. 5. Average PSNR values and frame loss ratios comparison between the POMDP-based streaming and the heuristic method for channel phase 1.

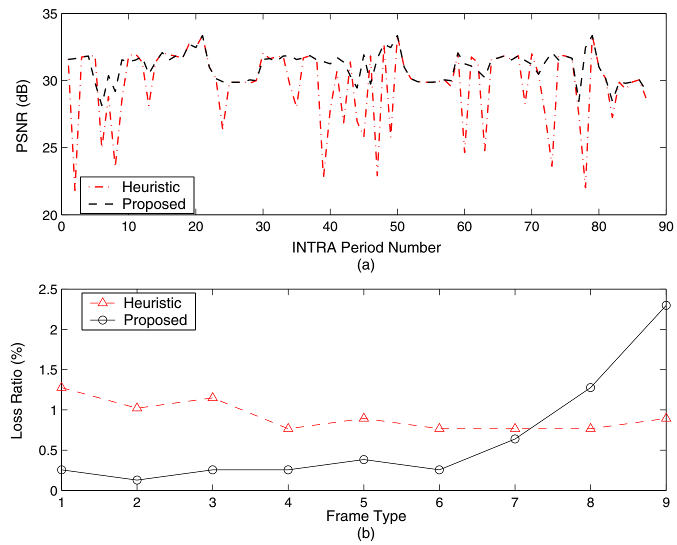

Fig. 6. Average PSNR values and frame loss ratios comparison between the POMDP-based streaming and the heuristic method for channel phase 4.

References

[1] D. Qiao and K. G. Shin, "A two-step adaptive error recovery scheme for video transmission over wireless networks," in *Proc. of IEEE INFOCOM2000*, pp. 1698-1704, 2000.

[2] J. Cai, Q. Zhang, W. Zhu, and C. W. Chen, "An FEC-based error control scheme for wireless MPEG-4 video transmission," in *Proc. of IEEE WCNC2000*, vol. 3, pp. 1243-1247, 2000.

[3] M. Zorzi, "Performance FEC and ARQ error control in bursty channels under delivery constraints," in *Proc. of IEEE VTC'98*, Ottawa, Canada, May 1998.

[4] C.-C. Liu and S. S. Chen, "Providing unequal reliablity for transmitting layered video streams over wireless networks by multi-ARQ schemes," in *Proc. of IEEE ICIP'99*, vol. 3, pp. 100-104, 1999.

[5] Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," in *Proc. of the IEEE,* pp. 974-997, 1998.

[6] B. Girod, M. Kalman, Y. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," *Wireless Communications and Mobile Computing,* vol. 2, no. 6, pp. 549-552, September 2002. *Invited paper.*

[7] Z. Jiang and L. Kleinrock, "A packet selection algorithm for adaptive transmission of smoothed video over a wireless channel," *Journal of Parallel and Distributed Computing,* vol. 60, pp. 494-509, 2000.

[8] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Packet Video 2002,* Pittsburgh, April 2002.

[9] W. Tan and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. on Multimedia*, vol. 1, No. 2, pp. 172-186, 1999.

[10] J. Chakareski and P. A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," Microsoft Research Technical Report MSR-TR-2002-81, August 2002.

[11] J. Chakareski and B. Girod, "Rate-Distortion Optimized Packet Scheduling and Routing for Media Streaming with Path Diversity", to appear in *Proc. IEEE Data Compression Conference*, Snowbird, UT, Apr. 2003.

[12] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of pacetized media", *IEEE Trans. on Multimedia,* submitted.

[13] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Proc. Packet Video Workshop*, Pittsburg, PA, April 2002.

[14] Y. J. Liang, John G. Apostolopoulos and B. Girod, "Analysis of packet loss for compressed video: does burst-length matter?" in *Proc. of IEEE ICASSP-2003*, Hong Kong, Apr. 2003.

[15] http://www.cs.brown.edu/research/ai/pomdp

[16] H. S. Wang, Pao-Chi Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 45, No. 2, May 1996.

[17] M. Zorzi, R. R. Racd, L. B. Milstein, "A Markov model for block errors on fading channels", in *Proc. Personal, Indoor and Mobile Radio Communications, PIMRC'96., IEEE*, Oct 1996.

[18] M. Zorzi, R. R. Rao, L. B. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels", in *Proc. Universal Personal Communications, Fourth IEEE International Conference on*, Nov 1995.

[19] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, June 2000.