Rate-Distortion Optimized Streaming of Packetized Media

Philip A. Chou, *Fellow, IEEE,* and Zhourong Miao, *Member, IEEE*

***Abstract*—This paper addresses the problem of streaming packetized media over a lossy packet network in a rate-distortion optimized way. We show that although the data units in a media presentation generally depend on each other according to a directed acyclic graph, the problem of rate-distortion optimized streaming of an entire presentation can be reduced to the problem of error-cost optimized transmission of an isolated data unit. We show how to solve the latter problem in a variety of scenarios, including the important common scenario of sender-driven streaming with feedback over a best-effort network, which we couch in the framework of Markov decision processes. We derive a fast practical algorithm for nearly optimal streaming in this scenario, and we derive a general purpose iterative descent algorithm for locally optimal streaming in arbitrary scenarios. Experimental results show that systems based on our algorithms have steady-state gains of 2–6 dB or more over systems that are not rate-distortion optimized. Furthermore, our systems essentially achieve the best possible performance: the operational distortion-rate function of the source at the capacity of the packet erasure channel.**

***Index Terms*—Audio coding, channel coding, error correction, Internet, Markov processes, multimedia communication, optimal control, protocols, video coding.**

I. Introduction

THIS paper addresses the problem of streaming packetized media over a lossy packet network, in a rate-distortion optimized way. In a streaming media system, a server prestores encoded media data and transmits it on demand to a client for playback in real time. The client buffers the data that it receives and begins playback after a short delay of up to several seconds. This delay is fixed and does not depend on the length of the presentation. Once the client begins playback, it is able to continue without interruption until the end of the presentation. It is this continuous playback with fixed delay that distinguishes streaming from download-and-play schemes. Furthermore streaming is distinguished from telephony and conferencing by its ability to store media data encoded offline, and by its tolerance to a longer playback delay. Streaming, telephony, and conferencing all transmit in real-time; however, for streaming the media data must be encoded without the benefit of knowing the state of the channel during transmission. For this reason, in a streaming media system, the encoding must be flexible and the server must adaptively select and transmit the correct data to the client, as a function of the state of the network as observed by the client or server. In this paper, we show, for arbitrary encodings and packetizations of multiple media, which packets to select for transmission, when to transmit them, and how to transmit them (e.g., with high or low quality of service), to minimize the expected distortion subject to constraint on the expected rate, where the expectations are taken over channel realizations. We measure rate as the total number of bytes transmitted (or more generally, as the total cost of bytes transmitted), and we measure distortion as the total end-to-end distortion of the presentation in arbitrary but incrementally additive units.

We set up a general framework for rate-distortion optimized streaming of packetized media, and within it we consider several scenarios. Throughout, we assume that the network loses (i.e., drops) or corrupts packets at random, and delivers those packets that it does not lose after a random delay. However, the network may or may not have multiple qualities of service (e.g., with different probabilities of loss, corruption, and delay) available at different costs per transmitted byte. Also, the network may or may not allow variations in transmission rate. Finally, the network may or may not provide a back channel, which can be used either for feedback (e.g., acknowledgment) from the receiver in a sender-driven mode, or for control of the sender (e.g., requests for transmission) in a receiver-driven mode. Thus our framework handles a variety of scenarios of current interest: sender-driven or receiver-driven streaming, streaming over best-effort networks such as today's Internet, streaming over multiple overlay networks, streaming over networks with integrated or differentiated services, streaming over combined wireline/wireless networks, and streaming over a network with access to multiple servers. The same framework has also been shown to handle error control for the case of receiver-driven layered multicast [1], [2].

We present the major ideas in our paper as follows. In Section II, we cover various preliminaries, including our source and channel models. Our source model consists of a labeled directed acyclic dependence graph in which each node represents a packetized data unit.

In Section III, we show that any of the aforementioned transmission scenarios can be abstracted as a set of choices for sending single packets, in which each choice π is associated with a cost per byte $\rho(\pi)$ of transmitting the packet and an error probability $\epsilon(\pi)$ of not delivering the packet by its deadline. This leads to the concept of an error-cost function $\epsilon(\rho) = \min_\pi\{\epsilon(\pi) : \rho(\pi) \leq \rho\}$, for which optimal performance (for a single packet) can be achieved by selecting the transmission option π minimizing the Lagrangian $\epsilon(\pi)+\lambda'\rho(\pi)$

Manuscript received March 21, 2001; revised May 14, 2005. The associate editor coordinating the review of this paper and approving it for publication was Prof. Tsuhan Chen.

P. A. Chou is with the Microsoft Corporation, Redmond, WA 98052-6399 USA (e-mail: pachou@microsoft.com).

Z. Miao is with the Sony Corporation, Santa Clara, CA 95054 USA (e-mail: zmiao@ieee.org).

Digital Object Identifier 10.1109/TMM.2005.864313

for some Lagrange multiplier λ'. For scenarios involving a back channel, we identify the transmission options as policies in a Markov decision process.

In Section IV, we show how to relate the error-cost functions for the packets to the distortion-rate function for the entire multimedia presentation. An optimal distortion-rate performance $D(R)$ for the entire presentation can be achieved by minimizing $D+\lambda R$ for some Lagrange multiplier λ. In turn, $D+\lambda R$ can be minimized by individually minimizing the packet Lagrangians $\epsilon(\pi)+\lambda'\rho(\pi)$ for appropriately chosen Lagrange multipliers λ'. The Lagrange multipliers λ' ultimately depend, cyclically, on the error probabilities $\epsilon(\pi)$. However, we develop an iterative descent algorithm for finding solutions that are locally optimal.

In Section V, we show how to combine our optimization algorithm with window and rate control to obtain practical streaming media systems.

In Section VI, we report experimental results focusing on sender-driven streaming over a best-effort network. Using simulations, we show that our systems gain up to 4 dB or more over systems approximating state-of-the-art commercial systems, over networks with 20% packet loss.

To our knowledge, the most closely related contemporaneous work is that by Miao and Ortega [3]–[5], which develops a low-complexity heuristic algorithm for sender-driven scheduling of packet transmissions over a best-effort network. Zhou and Li [6] also develop similar heuristics.

The most closely related rigorous work is that by Podolksy *et al.* [7], [8], which uses a Markov chain analysis to find the optimal policy for transmitting layered media at a fixed rate, including retransmissions, to minimize the end-to-end distortion. To make the analysis tractable, Podolsky *et al.* assume zero transmission delay and loss-free acknowledgment. Unfortunately they are unable to simulate an optimal system with more than a few source layers and more than a few transmission opportunities per frame, since the space of policies grows exponentially in both of these quantities. One of the main contributions of our paper is showing that this policy space can be factored so that the layers are only loosely coupled, resulting in complexity that grows roughly linearly in the number of layers.

The work of Chande *et al.* [9] was the first that we know of to formulate and solve the problem of optimal transmission over a noisy channel in the presense of feedback, using a Markov decision process framework. The work of Servetto [10] also recognized that optimal transmission over a noisy channel in the presense of feedback is a nonlinear stochastic control problem. Inspired by these works, Chou *et al.* [1], [2] used an iterative descent algorithm in a Lagrangian framework to find locally optimal transmission policies for hybrid FEC/ARQ, assuming jitter-free delay and loss-free retransmission requests, when the source layers are given by arbitrary directed acyclic graphs. Chou *et al.* applied their work to receiver-driven layered multicast of audio and video. That work became the starting point for the present paper, when we realized that the same methodology could be used to solve the problem of Podolsky *et al.* in a practical way.

There have been numerous other papers that perform some kind of rate-distortion optimization for transmission of packetized media. Many have focused on the problem of source rate control in the absence of transmission errors [11]–[14]. (Works that address the problem of source rate control in the presense of transmission errors, but are not rate-distortion optimized, include [15]–[20].) Other works have focused on the problem of error control using forward error correction (FEC). Many of these use the priority encoding transmission (PET) technique of Albanese *et al.* [21]–[30], which can be rate-distortion optimized using the algorithms of [22], [27], [31]–[35]. Others of these use a systematic rate-compatible technique [1], [2], which can be rate-distortion optimized using the algorithms of [1], [2], [36]–[38]. The present paper is an extension of these latter methods. (Still others use "signal processing FEC" for error control [39]–[43], for which rate-distortion optimization is still a research topic [4].) Some papers have investigated the problem of error control using retransmission-based protocols, e.g., [45]–[51]. However, with the exception of those works listed in the previous paragraph, to our knowledge, none are rate-distortion optimized. Finally, a few papers suggest the use of multiple qualities of service (e.g., diffserv) to support more cost-effective media transmission at a higher quality [52]–[54]. Of these, only [52] attempts to optimize the distortion subject to transmission rate constraints. Our paper substantially furthers the work in this direction.

During the time that this paper has been in review, based on a preprint of this paper appearing as a technical report [55], a number of papers have continued work along lines outlined in this paper, dealing with rate-distortion optimized streaming over a wireless last hop [56]–[59], over multiple paths [60], [61], over multiple qualities of service [62], from multiple servers [63], [64], through a caching proxy [65]–[68], with adaptive media playout [69], [70], with multiple deadlines [71], with rich acknowledgment [72], and with improved distortion-rate modeling [73]–[75], as well as streaming of light fields [76], [77] and other general advances in streaming using rate-distortion optimization [78]. Other work, not in our framework, has also begun for rate-distortion optimized streaming, such as [79]–[83], as well as streaming over diffserv [84]–[86]. The present paper is a considerably shortened version of [55].

II. Preliminaries

In this section, we cover various preliminaries, including our source and channel models, and we state the distortion-rate optimization problem that we are trying to solve. Table I summarizes our notation.

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. If the server selects a data unit for transmission, the data unit is put into a packet and sent across the network. If the packet is lost, the data unit may be sent again in another packet. In general we assume a one-to-many correspondence between data units and packets. However, each packet contains one and only one data unit.

Regardless of how many media objects (audio, video, etc.) there are in a multimedia presentation, and regardless of what algorithms are used for encoding and packetizing those media objects, the result is a set of data units for the presentation whose interdependencies can be expressed by a directed acyclic graph. Each node of the graph corresponds to a data unit, and each edge of the graph directed from data unit l' to data unit l corresponds

TABLE I
NOMENCLATURE

l, l', l''	data units
$l' \prec l$	l' is an ancestor of l
B_l	size in bytes of data unit l
Δd_l	importance of data unit l, i.e., decrease in distortion upon decoding data unit l
$t_{DTS,l}$	delivery deadline of data unit l
$N = N_l$	number of transmission opportunities for data unit l
$t_{0,l}, t_{1,l}, \ldots, t_{N-1,l}$	transmission opportunities for data unit l
t_X, s_X, r_X	time of event X on media, sender, and receiver clocks
FTT, BTT, RTT	forward, backward, and round trip times
ϵ_F, $p_F(\tau \mid \text{not lost})$ ϵ_B, $p_B(\tau \mid \text{not lost})$ ϵ_R, $p_R(\tau \mid \text{not lost})$	loss probabilities and delay densities on forward, backward, and round trip paths
κ_F, α_F, n_F κ_B, α_F, n_B κ_R, α_R, n_R	shift, memory, and order parameters of shifted Gamma distributions modeling delay densities on forward, backward, and round trip paths
μ_F, σ_F^2 μ_B, σ_B^2 μ_R, σ_R^2	mean and variance of delay distributions on forward, backward, and round trip paths
π	policy for transmitting a single data unit
$\rho = \rho_\pi = \rho(\pi)$	expected cost (e.g., redundancy) of transmitting a single data unit under policy π
$\epsilon = \epsilon_\pi = \epsilon(\pi)$	late/loss probability for transmitting a single data unit under policy π
λ', J_π	Lagrange multiplier and Lagrangian $\epsilon_\pi + \lambda' \rho_\pi$
$s_0, s_1, \ldots, s_{N-1}$	transmission opportunities (on sender clock)
$r_0, r_1, \ldots, r_{N-1}$	request opportunities (on receiver clock)
$a_0, a_1, \ldots, a_{N-1}$	actions taken at each transmission opportunity
$o_0, o_1, \ldots, o_{N-1}$	observations made after each transmission opportunity
$q_0, q_1, \ldots, q_{N-1}$	state attained before each transmission opportunity
$P_\pi(q_{i+1} \mid q_i)$	transition probability under policy π
Q_π	set of complete paths under policy π
$\boldsymbol{q} = (q_0, q_1, \ldots, q_F)$	path from initial to final state
$P_\pi(\boldsymbol{q})$	probability of path $\boldsymbol{q}$ under policy π
$\rho_\pi(\boldsymbol{q})$, $\epsilon_\pi(\boldsymbol{q})$, $J_\pi(\boldsymbol{q})$	cost, expected error, and expected Lagrangian of path $\boldsymbol{q}$ under policy π
L	number of data units in a given group
$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$	vector of transmission policies for all data units in a group
$R = R(\boldsymbol{\pi})$	expected cost (e.g., number of bytes) to stream entire presentation under $\boldsymbol{\pi}$
$D = D(\boldsymbol{\pi})$	expected distortion (e.g., squared error) over entire presentation under $\boldsymbol{\pi}$
λ, $J(\boldsymbol{\pi})$	Lagrange multiplier and Lagrangian $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$
d_0	distortion over entire presentation if no data units are decoded
D_0, ΔD_l	expected value of d_0, Δd_l
S_l	sensitivity to losing data unit l
ISA algorithm	Iterative Sensitivity Adjustment algorithm
X1 algorithm	transmit-one algorithm: $\arg\min_\pi J_\pi$
$[t_{lag}(s), t_{lead}(s)]$	window of delivery deadlines of data units eligible for transmission
δ, ν	playback delay and speed
T	interval between transmission opportunities
$W = NT$	duration of transmission opportunities
GOF	group of frames
ACK, NAK	acknowledgement, negative acknowledgement
FEC, ARQ	forward error correction, automatic repeat request
RaDiO	rate-distortion optimization

to a dependence of data unit l on data unit l'. That is to say, in order for data unit l to be decoded, data unit l' must also be decoded. This induces a partial order between data units, for which we write $l' \prec l$ if l' is an ancestor of l (or equivalently if l is a descendant of l'). Thus, if a set of data units is received by the client, only those data units whose ancestors have all been also received can be decoded.

Typically, the graph of dependencies between *all* of the data units in a presentation is a collection of connected components, where each connected component is itself a directed acyclic graph representing the dependencies between all of the data units of an independently encoded and packetized group of frames (GOF) of one media type. Some such directed acyclic dependence graphs are illustrated in Fig. 1. Fig. 1(a) shows a dependence graph typical of an embedded encoding of a group of frames, which is packetized sequentially. Fig. 1(b) shows a dependence graph typical of an encoding by a standard video coder of a group of (IBBPBBPBBP) frames, which is packetized as one data unit per frame. And Fig. 1(c) shows a dependence graph typical of an encoding of a group of frames by MPEG-4's fine grain scalability (FGS) mode [87], [88].

The dependence graph is computed offline at the time that the media data are encoded and packetized, and the data unit dependencies are stored along with the data units in the file on the media server. Also stored in the file and labeling each data unit l in the dependence graph are three constant quantities: its data unit size B_l in bytes, its importance Δd_l in units of distortion, and its timestamp $t_{\mathrm{DTS},l}$. We now discuss each of these in turn.

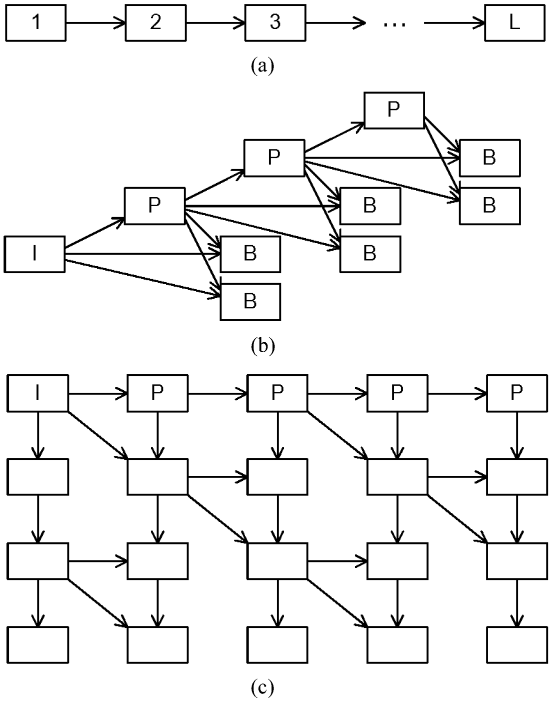


Fig. 1. Directed acyclic dependence graphs. (a) Sequential dependencies typical of embedded codes. (b) Dependencies between IBBPBBPBBP video frames. (c) Typical dependencies for MPEG-4 progressive fine grain scalability mode.

The data unit size B_l is the number of source bytes in the data unit.

The importance Δd_l is the amount by which the distortion at the receiver will *decrease* if the data unit is *decoded* (on time) at the receiver. Recall that a data unit can be decoded only if all of the data units on which it depends can also be decoded. For example, in Fig. 1(a), Δd_l for the third data unit in the sequence is the decrease in distortion if three data units are decoded instead of only two. Similarly, in Fig. 1(b), Δd_l for a B frame is the decrease in distortion if the B frame is decoded, compared to the distortion if the B frame is not decoded. In this way, the overall distortion can be computed as the initial distortion d_0 (i.e., the distortion if no data units are decoded) less the sum of the decreases Δd_l over all data units l that have been decoded on time. We say the distortion is *incrementally additive* with respect to the partial order given by the dependence graph. An important limitation of this incrementally additive model is that the amount by which the distortion decreases when a data unit is decoded does not depend on whether its sibling or cousin data units are decoded. So for example, in this model, the decrease in distortion when a B frame is decoded does not depend on whether or not any other B frame is decoded. Strictly speaking, this rules out exact modeling of a number of error concealment techniques. Fortunately, the model can still provide good approximations to the most likely variations in distortion under arbitrary error concealment techniques, with properly chosen parameters.[1] Extensions to the model that cover arbitrary error concealment techniques without approximation are treated in [73]–[75], [89].[2]

The timestamp $t_{\mathrm{DTS},l}$ is the time by which the data unit must be decoded to be useful (i.e., for the distortion to decrease by Δd_l). This corresponds to the decoder time-stamp (DTS) in MPEG terminology, and represents the time at which the decoder extracts the data from its input buffer prior to presentation (which in turn occurs at the presentation timestamp, PTS). Thus, in the context of the server/client model for streaming, $t_{\mathrm{DTS},l}$ is the *delivery deadline* by which data unit l must arrive at the client, or be too late to be usefully decoded. Packets containing a data unit that arrive after the data unit's delivery deadline are discarded.

Each data unit l is also labeled by the server on the fly with a set of $N = N_l$ transmission opportunities $t_{0,l}, t_{1,l}, \ldots, t_{N-1,l}$ prior to $t_{\mathrm{DTS},l}$ at which the data unit may be put into a packet and transmitted. Often this set of transmission opportunities is a single time $t_{0,l}$ (such as a "send time") prior to the delivery deadline, but in general we assume it is a finite set of times $t_{0,l}, t_{1,l}, \ldots, t_{N-1,l}$ (such as the set of times at $T = 50$ ms intervals within a window $[t_{\mathrm{lag}}, t_{\mathrm{lead}}]$) prior to the delivery deadline. Determination of this set is addressed in Section V.

It pays to be careful about the temporal coordinate systems (or clocks) in which time is expressed. In this paper, we deal with three different temporal coordinate systems: the media (or encoder) temporal coordinate system t, the sender (or server) temporal coordinate system s, and the receiver (or client) temporal coordinate system r. Each of these is related to the other by an affine coordinate transformation. For details, see [55]. We use the notation t_X, s_X, and r_X to denote the time of a single event X in each of the three temporal coordinate systems.

We model the network as an independent time-invariant packet erasure channel with random delays. That means that if the sender inserts a packet into the network at sender time s, then the packet is lost with some probability, say ϵ_F, independent of s. However, if the packet is not lost, then it arrives at the receiver at sender time s', where the forward trip time $\mathrm{FTT} = s' - s$ is randomly drawn according to probability density $p_F(\tau|\text{not lost})$. Each packet is lost or delayed independently of the other packets. This independence and time-invariance is reasonable over short time scales (such as a few seconds), provided the sender's packets are not self-congesting.[3] More sophisticated modeling with hidden Markov models (to accommodate "good" and "bad" network states, for example) is also possible in our framework [60]. However, in practice it is sufficient simply to estimate ϵ_F and $p_F(\tau|\text{not lost})$ given the recent past. This allows the distributions to change

[1] In general, Δd_l should be set to the *expected* decrease in distortion if data unit l is decoded, where the expectation is taken over all possible combinations of losses of data units that are not ancestors of data unit l. For example, in Fig. 1(b), Δd_l for a B frame should be set to the weighted average of the decrease in distortion if it is decoded with and without the benefit of its neighboring B frame.

[2] With some error concealment techniques, it may be difficult to determine the Δds in real time. However, it is always possible to determine the Δds offline, e.g., during creation of the media file that separates real-time encoding from real-time streaming.

[3] Over short time scales, the underlying state of the network path, including the location and load of the bottleneck queue, remains relatively constant. Assuming the sender's packets are not self-congesting, each packet from the sender leaves the bottleneck queue before the next packet from the sender arrives in the queue, in steady state. (Otherwise, the sender's packets would build up in the queue and would cause congestion.) Therefore the set of packets already in the queue is different for each of the sender's packets, and hence the number of such packets (the primary determiner of loss and delay for the sender's packets) is approximately independent and identically distributed given the network's underlying state.

slowly over time to reflect changes in the underlying network state, which has the accuracy of hidden Markov modeling, yet the tractability of independent modeling.

For convenience, we combine the packet loss probability and the packet delay density into a single probability measure, by assigning $\mathrm{FTT} = \infty$ in the event that the packet is lost. This places mass ϵ_F at infinity, and weights the density $p_F(\tau|\text{not lost})$ by the factor $(1-\epsilon_F)$. Thus

$$P\{\mathrm{FTT} > \tau\} = \epsilon_F + (1-\epsilon_F)\int_\tau^\infty p_F(t|\text{not lost})dt$$

which is the probability that a packet sent at time s is not received by time $s+\tau$, whether lost or simply delayed. In principle it is not ever possible to determine by waiting for a packet whether it is lost, or just delayed for a very long time.

We assume that the back channel, if available, can be similarly characterized. If the client sends a packet to the server, then the packet is lost with probability ϵ_B, otherwise it is delayed according to density $p_B(\tau|\text{not lost})$. The probability that the backward trip time is greater than τ is

$$P\{\mathrm{BTT} > \tau\} = \epsilon_B + (1-\epsilon_B)\int_\tau^\infty p_B(t|\text{not lost})dt.$$

The round trip time $\mathrm{RTT} = \mathrm{FTT} + \mathrm{BTT}$ is by definition the sum of forward and backward trip times. The probability that the round trip time is greater than τ is therefore

$$\begin{aligned} P\{\mathrm{RTT} > \tau\} = {} & \epsilon_F + (1-\epsilon_F)\epsilon_B \\ & + (1-\epsilon_F)(1-\epsilon_B)\int_\tau^\infty p_R(t|\text{not lost})dt, \end{aligned}$$

where p_R is the convolution of p_F and p_B.

We do not assume any particular form for the densities p_F, p_B, or p_R. However, for concreteness in the next section and in Section VI (Experimental Results) we do model these distributions parametrically. As in [90], we model packet delay as having a shifted Gamma distribution with rightward shift κ and parameters n and α, e.g.,

$$p_F(\tau|\text{not lost}) = \frac{\alpha_F}{\Gamma(n_F)}(\alpha(\tau-\kappa_F))^{n_F-1}e^{-\alpha_F(\tau-\kappa_F)} \quad (1)$$

for $\tau \geq \kappa_F$. This is the distribution of a random variable that is equal to a constant κ_F plus the sum of n_F independent identically distributed exponential random variables each with parameter α_F [91]. One way to interpret this is that the forward trip time FTT is the result of a packet going through n_F routers, each of which requires a constant processing time κ_F/n_F plus waiting time in a steady state M/M/1 queue [92]. Since an exponential random variable with parameter α has mean $1/\alpha$ and variance $1/\alpha^2$, the forward trip time as modeled by (1) has mean $\mu_F = \kappa_F + n_F/\alpha_F$ and variance $\sigma_F^2 = n_F/\alpha_F^2$. The accuracy of this model has recently been verified in [93].

We end this section with a discussion of our objective: rate-distortion optimized streaming of any given *presentation*, or finite-duration set of packetized media. By *rate* we mean the expected cost R of streaming the entire presentation. Cost may be measured as the number of bytes transmitted. However it can also be measured more generically. As we mentioned in the Introduction, each data unit is sent with some transmission option π, which has cost per source byte $\rho(\pi)$ and hence a data unit cost $B\rho(\pi)$, where B is the size of the data unit in bytes. The cost of streaming the entire presentation is the sum of the data unit costs for all of the data units transmitted. The rate R is the expected value of this total cost, averaged over all possible realizations of the random channel, for the given presentation.

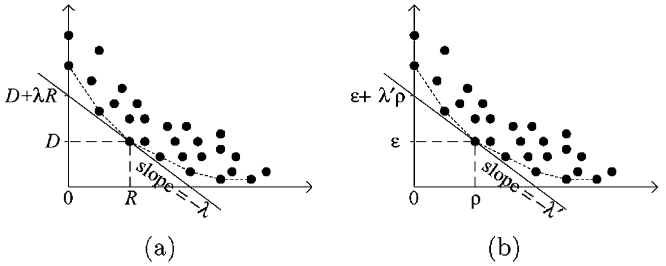


Fig. 2. (a) Set of achievable distortion-rate pairs, its lower convex hull (dotted), and an achievable pair (R, D) minimizing the Lagrangian $D + \lambda R$. Each dot is the (R, D) performance of some algorithm. (b) Likewise, the set of achievable error-cost pairs, its lower convex hull, and an achievable pair (ρ, ϵ) minimizing the Lagrangian $\epsilon + \lambda'\rho$.

By *distortion* we mean the expected distortion D incurred for the entire presentation. As we mentioned earlier in this section, whenever a data unit is decoded on time at the receiver, the distortion decreases (from some initial large distortion d_0) by the importance Δd of the data unit. The distortion incurred for the entire presentation is therefore some initial large distortion d_0 less the sum of the importances for all the data units decoded on time. The distortion D is the expected value of this total distortion, averaged over all possible realizations of the random channel, for the given presentation.

We seek a streaming algorithm that, given any presentation θ, has the minimum possible expected distortion $D = D_\theta(R)$ for an expected rate R. By restricting ourselves to algorithms whose rate-distortion performance (R, D) for the given presentation lies on the lower convex hull of the set of all rate-distortion performances achievable (for the presentation) by some algorithm, as illustrated in Fig. 2(a), we can find an optimal algorithm with expected distortion D and expected rate R by minimizing the Lagrangian $D + \lambda R$ for some positive Lagrange multiplier λ. Finding such an algorithm, under various communication scenarios, is the objective of this paper.

III. Transmitting a Single Data Unit

In this section, we study the problem of optimally transmitting a single data unit. Knowing whether, when, and how to best transmit each data unit in isolation will lead (in Section IV) to optimal transmission of the entire presentation.

When considering transmission of only a single data unit, the distortion-rate measures can be normalized. Rather than measuring distortion in terms of, say, squared error, expected distortion can be measured as the "error probability" or more precisely the late/loss probability, that is, the probability that the data unit does not arrive at its destination on time. Rather than measuring rate in terms of, say, bytes per second, rate can be measured as the expected number of times the data unit is transmitted, or more generally, as the expected number of bytes transmitted per source byte, or more generally still, as the expected cost per source byte to transmit the data unit. We refer to these normalized distortion-rate measures as "error-cost" measures.

An algorithm for transmitting a single data unit, here called a policy, has an expected error ϵ and an expected cost ρ, where the expectations are taken over channel realizations, e.g., packet losses and delays. An optimal policy is one that has the minimum possible error ϵ for its expected cost ρ. By restricting ourselves to policies whose error-cost performance (ρ, ϵ) lies on the lower convex hull of the set of all error-cost performances achievable by some policy, as illustrated in Fig. 2(b), we can find an optimal policy with expected error ϵ and expected cost ρ by minimizing the Lagrangian $\epsilon + \lambda' \rho$ for some positive Lagrange multiplier λ'. In the next section we show that we can indeed restrict attention to such policies when looking for a streaming algorithm that minimizes $D + \lambda R$. We now explore the error-cost performances that can be achieved in various scenarios.

Scenario A: Single QoS, no feedback. This is the simplest scenario. A data unit with delivery deadline s_{DTS} can either be transmitted at time $s_0 < s_{\text{DTS}}$, or not, over a network with a single quality of service (QoS) e.g., a best-effort network. In this scenario, there are only two error-cost possibilities. If the data unit is not transmitted, then the error probability is one, while the cost per source byte (expected number of packet transmissions) is zero. On the other hand, if the data unit is transmitted, then the error probability is $\epsilon \equiv P\{\text{FTT} > s_{\text{DTS}} - s_0\}$, while the cost per source byte is one. The operational error-cost function for this scenario and its convex hull are illustrated in Fig. 3(a) for $\epsilon = 20\%$.

Scenario B: Multiple QoS, no feedback. This is a more interesting scenario. Suppose that there are multiple qualities of service available on the network (or equivalently suppose there are multiple networks) each with its own forward trip time distribution and its own marginal cost per transmitted byte. For example, best-effort service with forward trip time $\text{FTT}^{(1)}$ could cost $\rho^{(1)} = 1$ microcent per transmitted byte, a low-loss service with forward trip time $\text{FTT}^{(2)}$ could cost $\rho^{(2)} = 4$ microcents per transmitted byte, and a low-delay, low-loss service with forward trip time $\text{FTT}^{(3)}$ could cost $\rho^{(3)} = 8$ microcents per transmitted byte. Then a data unit with delivery deadline s_{DTS} transmitted (or not) at time s_0 has one of four distortion-rate possibilities: $\epsilon^{(0)} = 1$ and $\rho^{(0)} = 0$; $\epsilon^{(1)} = P\{\text{FTT}^{(1)} > s_{\text{DTS}} - s_0\}$ and $\rho^{(1)} = 1$; $\epsilon^{(2)} = P\{\text{FTT}^{(2)} > s_{\text{DTS}} - s_0\}$ and $\rho^{(2)} = 4$; and $\epsilon^{(3)} = P\{\text{FTT}^{(3)} > s_{\text{DTS}} - s_0\}$ and $\rho^{(3)} = 8$. The operational error-cost function for this scenario and its convex hull are illustrated in Fig. 3(b).

Scenario C: FEC, no feedback. In this scenario only a single best-effort network is available, but the application can emulate different qualities of service over this network using forward error correction schemes of different strengths. For instance, if $k \geq 1$ and $n \geq k$, the application can emulate a higher quality of service for a data unit transmitted at time s_0 by 1) grouping it together with $k - 1$ other data units also to be transmitted at time s_0, 2) applying an (n, k) systematic Reed-Solomon code to produce $n - k$ parity units, and 3) transmitting the data packets plus their parity packets at time s_0. The original data unit cannot be recovered at the receiver by time s_{DTS} only if it is late or lost (which happens with probability $\epsilon = P\{\text{FTT} > s_{\text{DTS}} - s_0\}$) *and* at least $n - k$ of the other $n - 1$ packets are also late or lost (which happens with probability $f_{n,k} = \sum_{i=n-k}^{n-1} \binom{n-1}{i} \epsilon^i (1 - \epsilon)^{n-1-i}$). Thus the loss/late probability is reduced by a factor $f_{n,k}$ over best-effort, at a cost of n/k transmitted bytes per source byte. The error-cost performances for various values of (n, k) can be plotted (e.g., for $k = 1, \ldots, 8$ and $n = k, \ldots, k + 8$) to produce an operational error-cost function, as illustrated in Fig. 3(c).

Scenario D: Retransmission, no feedback. This is a similar scenario, in which quality of service can be emulated using retransmissions. Let $s_0, s_1, \ldots, s_{N-1}$ be N discrete transmission opportunities and let s_{DTS} be the delivery deadline. Repeatedly transmitting the data unit at all N opportunities results in a small loss/late probability (equal to $\prod_i P\{\text{FTT} > s_{\text{DTS}} - s_i\}$) but a large cost (equal to N). On the other hand transmitting the data unit at none of the N opportunities results in a large loss/late probability (equal to 1) but a small cost (equal to 0). Intermediate loss/late probabilities and costs can also be achieved and easily computed for any fixed transmission pattern. For example, suppose $a_0, a_1, \ldots, a_{N-1}$ represents a transmission pattern where $a_i = 1$ if a data packet is transmitted at time s_i and $a_i = 0$ otherwise. Then the loss/late probability is equal to $\prod_{i:a_i=1} P\{\text{FTT} > s_{\text{DTS}} - s_i\}$ while the cost is equal to $\sum_{i:a_i=1} 1$ transmitted bytes per source byte. The error-cost performances for all 2^N transmission patterns can be plotted to produce an operational error-cost function, as illustrated in Fig. 3(d).

Scenario E: Sender-driven retransmission, with feedback. The previous scenario becomes more realistic when combined with feedback. Suppose the receiver sends an acknowledgment packet back to the sender the instant that it receives a data packet, and that the sender truncates its transmission pattern upon receipt of the acknowledgment packet. Then although the loss/late probability remains the same, the expected number of data packet transmissions is reduced to $\sum_{i:a_i=1} (\prod_{j<i:a_j=1} P\{\text{RTT} > s_i - s_j\})$.[4] This scenario, which we refer to as sender-driven transmission over a single-QoS network using retransmissions with feedback, is the principal scenario considered in this paper. The operational error-cost function for this scenario is illustrated in Fig. 3(e).

Scenario F: Receiver-driven retransmission, with feedback. This is a receiver-driven version of the above scenario. The receiver initiates transmission by sending a request packet to the sender; the sender responds by sending a data packet to the receiver. Let $r_0, r_1, \ldots, r_{N-1}$ be N discrete request opportunities at which the receiver can transmit a request packet, and let r_{DTS} be the deadline for delivery of the data unit to the receiver. Suppose $a_0, a_1, \ldots, a_{N-1}$ represents a request pattern where $a_i = 1$ if a request packet is transmitted at time r_i and $a_i = 0$ otherwise. Suppose the sender transmits the data packet to the receiver the instant that it receives a request, and that the receiver truncates its request pattern upon receipt of the data unit. Then it is not too hard to show that the loss/late proba-

[4] To see this, consider that the quantity in parentheses is the expected value of the indicator function of the event that a data packet is transmitted at time s_i, which in turn is the probability that none of the previously transmitted packets are acknowledged by time s_i.

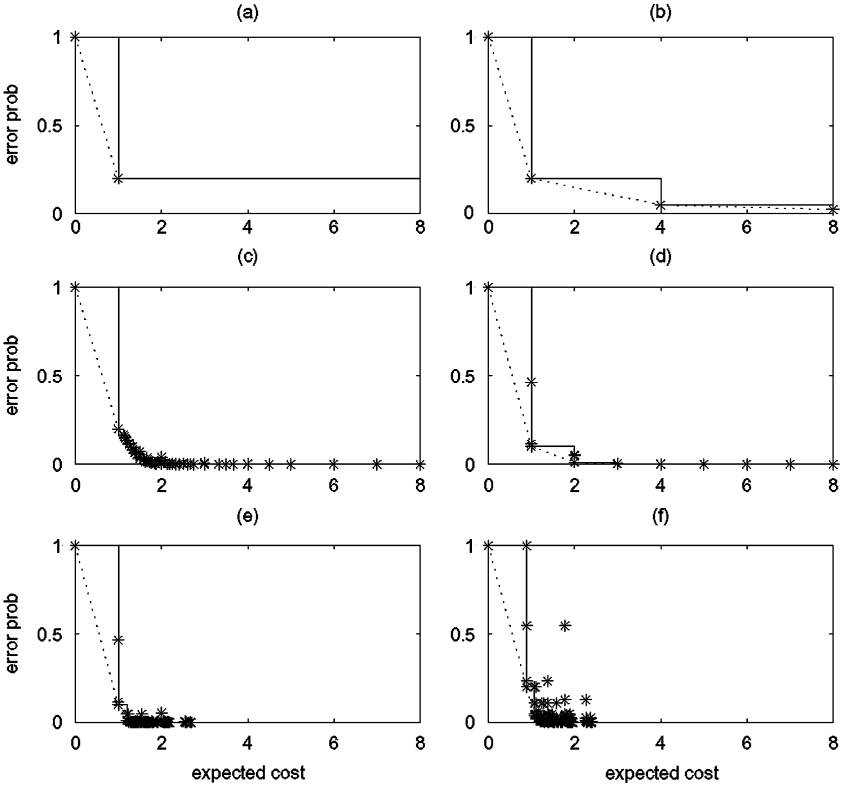


Fig. 3. Error-cost functions. (a) Single QoS, no feedback. (b) Multiple QoS, no feedback. (c) FEC, no feedback. (d) Retransmission, no feedback. (e) Sender-driven retransmission, with feedback. (f) Receiver-driven retransmission, with feedback.

bility is equal to $\prod_{i:a_i=1} P\{\mathrm{RTT} > r_{\mathrm{DTS}} - r_i\}$ (which is larger than in the sender-driven case) while the expected data packet transmission rate is equal to $\sum_{i:a_i=1}(\prod_{j<i:a_j=1} P\{\mathrm{RTT} > r_i - r_j\})P\{\mathrm{BTT} < \infty\}$ (which is smaller than in the sender-driven case). The operational error-cost function is illustrated in Fig. 3(f).

Hybrids of any of the above scenarios are also possible. For example, receiver-driven transmission over a multiple-QoS network using retransmissions with feedback can be handled by letting $a_i = Q_i \in \{1, 2, 3\}$ indicate a request by the receiver at time r_i for a data unit to be transmitted with quality of service Q_i (and $a_i = 0$ otherwise). In this case, the loss/late probability is equal to $\prod_{i:a_i \neq 0} P\{\mathrm{RTT}^{(Q_i)} > r_{\mathrm{DTS}} - r_i\}$, and the cost is equal to $\sum_{i:a_i \neq 0} \rho^{(Q_i)}(\prod_{j<i:a_j \neq 0} P\{\mathrm{RTT}^{(Q_j)} > r_i - r_j\})P\{\mathrm{BTT} < \infty\}$, where $\mathrm{RTT}^{(Q)} = \mathrm{BTT} + \mathrm{FTT}^{(Q)}$ is the round trip time over the single backward channel and the Q'th forward channel. This last "intserv/diffserv" scenario is mathematically equivalent to a number of other scenarios of interest, such as the "overlay" scenario in which the receiver is connected to the sender by multiple physical networks, each offering a different quality of service, and the "multiple server" scenario in which the receiver has access to multiple senders over different network paths, each offering a different quality of service.

In the remainder of this section, we study in detail, using a Markov decision process (MDP) framework, Scenario E: sender-driven transmission over a single-QoS network using retransmissions with feedback. Although the MDP framework can be used for efficient computation, its primary importance is conceptual. Indeed, the MDP framework can be used to generalize rate-distortion optimized streaming to other scenarios, which are beyond the scope of this paper, such as wireless [56]–[59], multipath [60], [61], multi-QoS [62], multiserver [63], [64], and caching proxy [65]–[68] scenarios. The reader is referred to the references for the applicability of the MDP framework to these scenarios.

A Markov decision process with finite horizon N is an N-step stochastic process through a state space in which an action can be taken at each state in a corresponding trellis of length N to influence the outgoing transition probabilities and thereby maximize an expected reward or minimize an expected cost along the transitions. The assignment of actions to trellis states is called a *policy* (denoted by π) and the optimal policy, in our context, is the one that minimizes the expected cost $\epsilon_\pi + \lambda' \rho_\pi$ of traversing the trellis in N steps starting from a known initial state.

Fig. 4 shows the trellis for the Markov decision process associated with Scenario E. The process begins in the initial state at time s_0. In this state, the sender can choose either to send the data unit, taking action $a_0 = 1$, or not to send the data unit, taking action $a_0 = 0$. If the sender chooses to send the data unit, then just prior to time s_1, the sender can observe either that some packet containing the data unit has been acknowledged, in which case $o_0 = 1$, or that no packet has been acknowledged, in which case $o_0 = 0$. If a packet containing the data unit has been acknowledged by time s_1, then the process enters a final state at time s_1. Otherwise the process enters a nonfinal state at time s_1, and the sender can once again choose either to send the data unit, or not, repeating the process up to a total of N times.

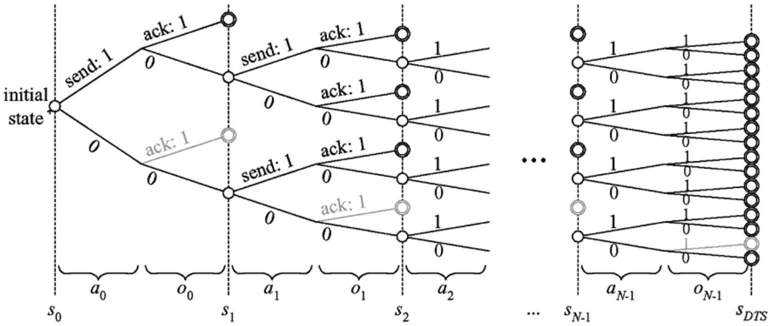


Fig. 4. Trellis for a Markov decision process. Final states are indicated with double circles.

Each state in the trellis (circles in Fig. 4) captures the action-observation history leading up to that state from the initial state. That is, a state q_i at time s_i represents a sequence of i action-observation pairs, $(a_0, o_0) \circ (a_1, o_1) \circ \ldots \circ (a_{i-1}, o_{i-1})$.

The action taken at a state determines the transition probabilities to the next state. Indeed, if the next state $q_{i+1} = q_i \circ (a_i, o_i)$ is the current state q_i followed by the action-observation pair (a_i, o_i), then (in this scenario)

$$P(q_{i+1}|q_i, a_i) = \begin{cases} p(a_0, \ldots, a_i) & \text{if } o_i = 0 \text{ (not ACK'd)} \\ 1 - p(a_0, \ldots, a_i) & \text{if } o_i = 1 \text{ (ACK'd)}, \end{cases}$$

where $p(a_0, \ldots, a_i) = \prod_{j \le i : a_j = 1} P\{\text{RTT} > s_{i+1} - s_j | \text{RTT} > s_i - s_j\}$. That is, the probability that no acknowledgment arrives by time s_{i+1} given that no acknowledgment arrived by time s_i is the product of the probabilities for each data packet sent at time s_j that no acknowledgment arrives by time s_{i+1} (i.e., $\text{RTT} > s_{i+1} - s_j$) given that no acknowledgment arrived by time s_i (i.e., $\text{RTT} > s_i - s_j$).

Thus any policy $\pi : q \mapsto a$ assigning actions to states induces a Markov chain with transition probabilities

$$P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i)).$$

Let $\mathcal{Q}_\pi$ be the set of all complete paths through this Markov chain, and let

$$\boldsymbol{q} = (q_0, q_1, \ldots, q_F) \in \mathcal{Q}_\pi. \tag{2}$$

That is, let $\boldsymbol{q}$ satisfy $q_{i+1} = q_i \circ (a_i, o_i)$ where $a_i = \pi(q_i)$ and $o_i = 0$ for $i < F - 1$. Then $\boldsymbol{q}$ has probability

$$P_\pi(\boldsymbol{q}) = \prod_{i=0}^{F-1} P_\pi(q_{i+1}|q_i) \tag{3}$$

transmission cost

$$\rho_\pi(\boldsymbol{q}) = \sum_{i=0}^{F-1} a_i \tag{4}$$

and error (loss/late probability)

$$\epsilon_\pi(\boldsymbol{q}) = \begin{cases} 0, & \text{if } o_{F-1} = 1 \text{ (ACK'd)} \\ q(a_0, \ldots, a_{F-1}), & \text{if } o_{F-1} = 0 \text{ (not ACK'd)}, \end{cases} \tag{5}$$

where $q(a_0, \ldots, a_{F-1}) = \prod_{j : a_j = 1} P\{\text{FTT} > s_{\text{DTS}} - s_j | \text{RTT} > s_{\text{DTS}} - s_j\}$. The latter expression follows from the facts that if the path leads to an acknowledgment, then the probability that the data packet is lost or late is zero, while if the path leads to no acknowledgment by time s_{DTS}, then the probability that the data packet is lost or late is the product of the probabilities that each data packet transmitted is lost or late $(\text{FTT} > s_{\text{DTS}} - s_j)$ given that no acknowledgment is received for that packet $(\text{RTT} > s_{\text{DTS}} - s_j)$.

Armed with definitions of probability, transmission cost, and error for each path, one can now express the expected cost and error for the Markov chain induced by policy π:

$$\rho_\pi \equiv E_\pi \rho_\pi(\boldsymbol{q}) = \sum_{\boldsymbol{q} \in \mathcal{Q}_\pi} P_\pi(\boldsymbol{q}) \rho_\pi(\boldsymbol{q}) \tag{6}$$

$$\epsilon_\pi \equiv E_\pi \epsilon_\pi(\boldsymbol{q}) = \sum_{\boldsymbol{q} \in \mathcal{Q}_\pi} P_\pi(\boldsymbol{q}) \epsilon_\pi(\boldsymbol{q}). \tag{7}$$

In principle, it is possible to enumerate all possible policies π, plot the error-cost performances $\{(\rho_\pi, \epsilon_\pi)\}$ in the error-cost plane, and produce an operational error-cost function for this scenario with the same results as in Fig. 3(e). However, if one is only interested in finding a point on the convex hull of the operational error-cost function, it is simpler matter to find the policy minimizing the expected Lagrangian

$$J_\pi \equiv \epsilon_\pi + \lambda' \rho_\pi = \sum_{\boldsymbol{q} \in \mathcal{Q}_\pi} P_\pi(\boldsymbol{q}) J_\pi(\boldsymbol{q}) \tag{8}$$

where $J_\pi(\boldsymbol{q}) \equiv \epsilon_\pi(\boldsymbol{q}) + \lambda' \rho_\pi(\boldsymbol{q})$. This can be accomplished with dynamic programming as described in [55], or branch and bound algorithms as described in [94]. In this paper, whatever algorithm is used to compute the optimal choice π minimizing $\epsilon_\pi + \lambda' \rho_\pi$, for whatever scenario, we will call the transmit-one (X1) algorithm.

As an example, Fig. 5 shows the error-cost performances of different optimal policies $\pi^*_{\lambda'}$ computed by the X1 algorithm described above for different values of λ'. The expected error is shown on a logarithmic scale. Each optimal policy is shown as the sequence of actions $[a_0, a_1, \ldots, a_{N-1}]$ taken along the longest path in the Markov chain defined by the policy, that is, the sequence of actions taken by the sender at each transmission opportunity until it receives an acknowledgment. The all-zeros policy (which never transmits) is shown at the upper left with expected error equal to 1 and expected cost equal to 0. The all-ones policy (which always transmits) is shown at the lower right with

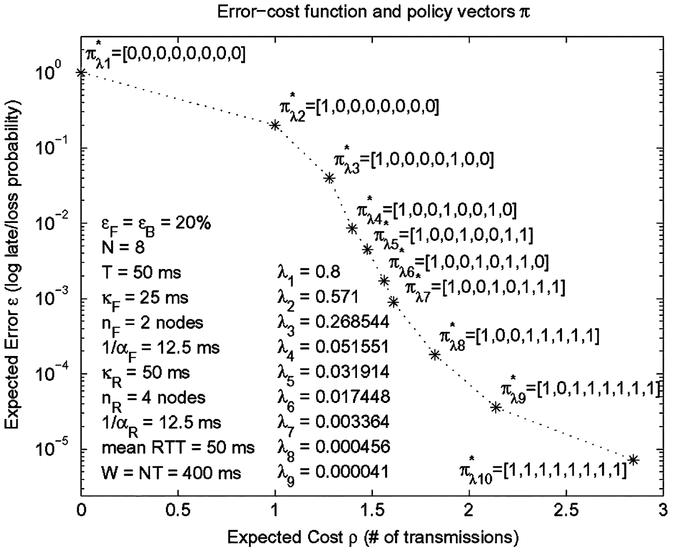


Fig. 5. Optimal policies and their error-cost performances. The optimal policies for different values of λ' are shown as sequences of actions $[a_0, a_1, \ldots, a_{N-1}]$.

expected error equal to 7×10^{-6} and expected cost equal to 2.8. Intermediate policies are shown in between. As λ' decreases, the optimal policy decreases in error but increases in cost. In this example, there are $N = 8$ transmission opportunities every $T = 50$ ms. The mean forward trip time is $\kappa_F + n_F/\alpha_F = T$ and the mean round trip time is $\kappa_R + n_R/\alpha_R = 2T$, using the parametric models discussed in Section II.

IV. Transmitting a Group of Data Units

In this section, we study how a whole group of interdependent data units can be transmitted in a distortion-rate optimized way, using as a building block the scenario-appropriate method for transmitting a single data unit.

Suppose we wish to transmit a group of L data units whose dependencies are specified by an arbitrary directed acyclic graph. These L data units could be all the data units in a session, all the data units in a group of frames, or only those data units whose delivery deadlines lie in a limited time window.

Let π_l be the transmission policy for data unit $l \in \{1, \ldots, L\}$ and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ be the vector of transmission policies for all L data units in the group. Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion and an expected transmission cost for the group. The expected transmission cost is the sum of the expected transmission costs for each data unit in the group. In turn, the expected transmission cost for each data unit $l \in \{1, \ldots, L\}$ is the expected cost per byte of transmitting the data unit, $\rho(\pi_l)$, times its size in bytes, B_l. That is, we have for the expected transmission cost

$$R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l). \tag{9}$$

In particular, if $\rho(\pi_l)$ is the expected number of times that data unit l is transmitted, then $R(\boldsymbol{\pi})$ is the expected number of bytes transmitted for all the data units in the group. When divided by the duration of the group, this yields the average bandwidth consumed.

The expected distortion for the group is somewhat more complicated to express. Let I_l be the indicator random variable that is 1 if data unit l arrives at the receiver on time, and is 0 otherwise. Then $\prod_{l' \preceq l} I_{l'}$ is 1 if data unit l is *decodable* by the receiver on time, and is 0 otherwise. If data unit l is decodable by the receiver on time, then the reconstruction error is reduced by the quantity Δd_l; otherwise the reconstruction error is not reduced. Hence the total reduction in reconstruction error for the group is $\sum_l \Delta d_l \prod_{l' \preceq l} I_{l'}$. Subtracting this quantity from the reconstruction error for the group if no data units are received, and taking expectations, we have for the expected distortion

$$D(\boldsymbol{\pi}) = D_0 - \sum_l \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) \tag{10}$$

where D_0 is the expected reconstruction error for the group if no data units are received, ΔD_l is the expected reduction in reconstruction error if data unit l is decoded on time, and $\epsilon(\pi_l)$ is the probability that data unit l does not arrive at the receiver on time (as computed in the previous section). Here we have used the assumption that the data packet transmission processes are independent in order to factor the expectation in (10).

With (9) and (10) for the expected transmission cost and expected distortion for any given policy vector now in hand, we are able to optimize the policy vector to minimize the expected distortion subject to a constraint on the expected transmission cost. By restricting ourselves to solutions on the lower convex hull of the set of rate-distortion pairs $\{(R(\boldsymbol{\pi}), D(\boldsymbol{\pi}))\}$, we can solve the problem by finding the policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian

$$\begin{aligned} J(\boldsymbol{\pi}) &= D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi}) \\ &= D_0 + \sum_l \left[\Delta D_l \left(-\prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) \right) + \lambda B_l \rho(\pi_l) \right]. \end{aligned} \tag{11}$$

The solution to this problem is completely characterized by the dependence graph, the set of distortion increments ΔD_l, and packet sizes B_l (which are determined by the source, source code, and packetization) and the error-cost functions $\epsilon(\pi)$ and $\rho(\pi)$ (which are determined by the transmission scenario and channel characteristics). This simplifies the problem of determining the quantities needed for the optimization. However, the minimization itself is complicated by the fact that the expression for the expected distortion cannot be split into a sum of terms each depending on only a single π_l, as is usually the case in rate allocation problems. Hence, we solve the problem using an iterative descent algorithm.

Our iterative approach is based on the method of alternating variables for multivariate minimization [95]. The objective function $J(\pi_1, \ldots, \pi_L)$ in (11) is minimized one variable at a time, keeping the other variables constant, until convergence. To be precise, let $\boldsymbol{\pi}^{(0)}$ be any initial policy vector and let $\boldsymbol{\pi}^{(n)} = (\pi_1^{(n)}, \ldots, \pi_L^{(n)})$ be determined for $n = 1, 2, \ldots$, as follows. Select one component $l_n \in \{1, \ldots, L\}$ to optimize at step n. This can be done round-robin style, e.g.,

```
Given $\lambda$, $\{(B_l, \Delta D_l)\}_{l=1}^{L}$,
0. Initialize:
    $\epsilon_1 = \cdots = \epsilon_L = \min_\pi \epsilon(\pi)$, $\rho_1 = \cdots = \rho_L = \max_\pi \rho(\pi)$,
    $D = D_0 - \sum_l \Delta D_l \prod_{l' \preceq l}(1 - \epsilon_{l'})$, $R = \sum_l B_l \rho_l$,
    $J^{(0)} = D + \lambda R$, and
    $n = 1$.
1. $l = l_n = ((n-1) \bmod L) + 1$
2. $S_l = \sum_{l' \succeq l} \Delta D_{l'} \prod_{l'' \preceq l' : l'' \neq l}(1 - \epsilon_{l''})$
3. $\lambda'_l = \lambda B_l / S_l$
4. $\pi_l^* = \arg\min_\pi \epsilon(\pi) + \lambda'_l \rho(\pi)$ [Algorithm X1]
5. $\epsilon_l = \epsilon(\pi_l^*)$,    $\rho_l = \rho(\pi_l^*)$
6. $D = D_0 - \sum_l \Delta D_l \prod_{l' \preceq l}(1 - \epsilon_{l'})$,    $R = \sum_l B_l \rho_l$
7. $J^{(n)} = D + \lambda R$
8. If $J^{(n)} = J^{(n-1)}$ stop; else $n = n + 1$ and go to Step 1.
Return $\pi_1^*, \ldots, \pi_L^*$.
```

Fig. 6. Iterative Sensitivity Adjustment (ISA) algorithm.

$l_n = ((n-1) \bmod L) + 1$. Then for $l \neq l_n$, let $\pi_l^{(n)} = \pi_l^{(n-1)}$, while for $l = l_n$, let

$$\begin{aligned}\pi_l^{(n)} &= \arg\min_{\pi_l} J\left(\pi_1^{(n)}, \ldots, \pi_{l-1}^{(n)}, \pi_l, \pi_{l+1}^{(n)}, \ldots, \pi_L^{(n)}\right) \\ &= \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l) \qquad (12)\end{aligned}$$

where (12) follows from (11) with

$$S_l^{(n)} = \sum_{l' \succeq l} \Delta D_{l'} \prod_{\substack{l'' \preceq l' \\ l'' \neq l}} \left(1 - \epsilon\left(\pi_{l''}^{(n)}\right)\right). \qquad (13)$$

The factor S_l can be regarded as the sensitivity to losing data unit l, i.e., the amount by which the expected distortion will increase if data unit l cannot be recovered at the receiver, given the current transmission policies for the other data units. Another interpretation of S_l is as the partial derivative of (10) with respect to $\epsilon_l = \epsilon(\pi_l)$, evaluated at $\boldsymbol{\pi}^{(n)}$. See [1], [2], [73], [74].

Now, the solution to (12) is simple. This is the problem of transmitting a single data unit, which can be solved with the X1 algorithm as described in the previous section: find the transmission policy π_l minimizing $\epsilon(\pi_l) + \lambda' \rho(\pi_l)$, where $\lambda' = \lambda B_l / S_l$. Thus, using the X1 algorithm, the policy vector $\boldsymbol{\pi}^{(n)}$ can be determined and the process can be repeated until $J(\boldsymbol{\pi}^{(n)})$ converges. Convergence is guaranteed because $J(\boldsymbol{\pi}^{(n)})$ is nonincreasing and bounded below. The overall algorithm, which we call the Iterative Sensitivity Adjustment (ISA) algorithm, is summarized in Fig. 6.

Along with the ISA algorithm, we have established the following.

Proposition (Sufficiency of the X1 algorithm). A necessary condition for $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ to minimize $J(\boldsymbol{\pi})$ is that each π_l minimizes $\epsilon(\pi_l) + \lambda'_l \rho(\pi_l)$ for some $\lambda'_l > 0$.

In summary, using the ISA algorithm in conjunction with the X1 algorithm, we are able to find transmission policies $\pi_1^*, \ldots, \pi_L^*$ for each of the L data units in a group such that after independently following the policies, the expected distortion $D(\boldsymbol{\pi}^*)$ and the expected transmission cost $R(\boldsymbol{\pi}^*)$ are minimal (or at least locally minimal) for each other, since they lie on the convex hull of all operational rate-distortion pairs $(R(\boldsymbol{\pi}), D(\boldsymbol{\pi}))$, for the given set of transmission opportunities.

It is important to note that this "optimal" performance can be improved upon, in principle, if the transmission scenario involves feedback. Although independence of the transmission processes allows us to factor the expectation across the product in (10), this independence may be too constraining when feedback is available since the information fed back for one data unit may benefit the transmission of another data unit. For example, the knowledge that data unit l has arrived at the receiver boosts the sensitivity of the data units l' that depend on l as well as the sensitivity of the data units l'' on which l depends.

Thus, when feedback is available, we attempt to improve performance with the heuristic of stepwise rate-distortion optimization. Specifically, we rerun the ISA algorithm at every transmission opportunity, taking into account the most recent information fed back for any of the data units. To be more specific, at every transmission opportunity s_k (assuming for simplicity that every data unit has the same set of transmission opportunities $s_0, s_1, \ldots, s_{N-1}$), for every data unit l, given that the transmission process has arrived at some state $q_{k,l}$, we condition the quantities in (2)–(8) on $q_{k,l}$, namely, the path $\boldsymbol{q} \in Q_\pi(q_{k,l})$, the path probabilities $P_\pi(\boldsymbol{q}|q_{k,l})$, the error probabilities $\epsilon_\pi(\boldsymbol{q}|q_{k,l})$, and ultimately, the expected cost, error, and Lagrangian $\rho_\pi(q_{k,l})$, $\epsilon_\pi(q_{k,l})$, and $J_\pi(q_{k,l})$. Then, we run the ISA algorithm on these conditional quantities to determine stepwise-optimal policies for each data unit, which we follow for one step.

This stepwise-optimal procedure can be likened to a procedure common among human agents who are assigned separate tasks toward a common goal, where achievement of the goal depends to varying degrees on achievement of the individual tasks. Suppose the agents call each other at the end of each day to report their state of progress. Since it is infeasible for an agent to follow from the outset an optimal strategy involving contingency plans for every possible daily state of every other agent, each agent instead optimizes his own long-term strategy assuming an expected level of success for each other agent. The agent is able to modify his long-term strategy daily, as the expected level of success of the other agents changes according to their status reports.

Although this stepwise-optimal procedure does not necessarily produce the optimal performance when feedback is available, it produces near-optimal performance, as our experimental results in Section VI show. Moreover, it is tractable, because it factors the full state space into a product of state spaces, one for each data unit (or agent). Although these state spaces are coupled, the coupling is loose. Hence it is possible to separately solve for the optimal policy for each data unit, and then run the ISA algorithm to couple these solutions. In contrast, the truly optimal solution involves a state space that grows exponentially in the number of data units as well as the number of transmission opportunities. Podolsky *et al.* studied such a solution in [7] and [8], and even with a simplified channel model, concluded that the problem is intractable when there are more than two data units and more than two transmission opportunities. Our

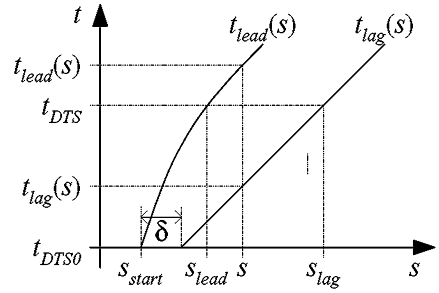


Fig. 7. Window control. The horizontal interval between s_{lead} and s_{lag} is the window of transmission opportunity for data units with delivery deadline t_{DTS}. The vertical interval between $t_{\text{lag}}(s)$ and $t_{\text{lead}}(s)$ is the set of delivery deadlines t_{DTS} whose data units are eligible for transmission at time s.

factorization of the problem into loosely coupled problems of transmitting only a single data unit is one of the main contributions of our work.

V. Practical Streaming Using Window and Rate Control

In the previous two sections, we showed how a hypothetical system can achieve locally optimal distortion-rate performance in nonfeedback scenarios, and stepwise-optimal distortion-rate performance in feedback scenarios. However, because we measure distortion-rate performance in an *average* sense, it is possible for such a hypothetical distortion-rate optimized system to transmit most of the data units in each group in a single burst, resulting in a large instantaneous rate despite a low average rate. When the group of data units is large, e.g., the entire session, this is untenable. However, window control can be used to spread out the transmissions over the duration of the session.

In window control, different data units are given different windows of transmission opportunities, based on their delivery deadlines. To be specific, at any given transmission time s, only those data units l whose delivery deadlines $t_{\text{DTS},l}$ fall within the window $[t_{\text{lag}}(s), t_{\text{lead}}(s)]$ are given the opportunity to transmit. The window boundaries $t_{\text{lag}}(s)$ and $t_{\text{lead}}(s)$ advance monotonically with s.

Fig. 7 graphs typical values of $t_{\text{lag}}(s)$ and $t_{\text{lead}}(s)$ as a function of the transmission time $s, s \geq s_{\text{start}}$. For any given transmission time s, the vertical interval $\{t_{\text{DTS}} : t_{\text{lag}}(s) \leq t_{\text{DTS}} \leq t_{\text{lead}}(s)\}$ is the set of delivery deadlines whose data units are eligible for transmission at time s. Conversely, for any given delivery deadline t_{DTS}, the horizontal interval $\{s : t_{\text{lag}}(s) \leq t_{\text{DTS}} \leq t_{\text{lead}}(s)\}$ is the set of times at which data units with delivery deadline t_{DTS} are eligible for transmission. It is during this horizontal interval that data units with delivery deadline t_{DTS} can be transmitted. In our experiments we choose

$$\begin{aligned} t_{\text{lag}}(s) &= t_{\text{DTS0}} + \nu[s - (s_{\text{start}} + \delta)] \\ t_{\text{lead}}(s) &= t_{\text{DTS0}} + \nu(s - s_{\text{start}}) \\ &\quad + (b/a)\ln(a(s - s_{\text{start}}) + 1) \end{aligned}$$

where the playback delay $\delta = 1$ s, the playback speed $\nu = 1$, and $a = b = 1$. The reasons for this choice are given in [55].

Now let us consider the transmission dynamics. Beginning at time s_{start}, at each transmission opportunity s (say, every $T =$ 50 ms), the ISA algorithm runs on the group of data units eligible for transmission at time s. We call this group of data units, $\{l : t_{\text{lag}}(s) \leq t_{\text{DTS},l} \leq t_{\text{lead}}(s)\}$, the *transmission buffer* at time s. For each data unit l in the transmission buffer, the ISA algorithm iteratively computes the sensitivity S_l [according to (13)], and produces the optimal policy π_l [minimizing $S_l\epsilon(\pi_l) + \lambda B_l \rho(\pi_l)$ according to (12)]. Upon convergence, each data unit l in the transmission buffer is sent (or not) according to the first action in policy π_l. Thus at each transmission opportunity, some of the data units in the transmission buffer are sent, and others are not. This process is repeated at each transmission opportunity. This system is our basic rate-distortion optimized (RaDiO) system, which we evaluate in the next section.

Even with window control, however, the transmission rate can be bursty. Rate control can be used in conjunction with window control to smooth the instantaneous transmission rate still further. The rate control mechanism we propose is similar, in a broad sense, to the rate control mechanisms found in standard video encoders. In standard video encoders, the rate control mechanism typically adjusts a quantization stepsize Q [96] or possibly a Lagrange multiplier λ [97], [98] to affect the instantaneous bit rate out of the encoder into an encoder buffer. If the encoder buffer is close to empty, then Q or λ is decreased to keep the buffer from underflowing, while if the encoder buffer is close to full, then Q or λ is increased to keep the buffer from overflowing. In this paper we propose a roughly similar mechanism for controlling the instantaneous rate of data packet transmissions out of the ISA algorithm. The Lagrange multiplier λ can be increased or decreased to adjust the number of data units selected for transmission at each transmission opportunity. Of particular interest is the special case in which λ is adjusted so that exactly one data unit is selected for transmission at each transmission opportunity. That is, starting from zero, λ is increased until there remains only one data unit l in the transmission buffer for which the optimal policy π_l (minimizing $S_l\epsilon(\pi_l) + \lambda B_l \rho(\pi_l)$) has "send" as its first action. This one data unit is relatively simple to find by creating a list of Lagrange multipliers $\Lambda = \{\lambda_l\}$, where for each data unit l, λ_l is the threshold for λ above which the data unit is not transmitted, and below which the data unit is transmitted, at the current transmission opportunity. Once this list is created, the data unit l to transmit is the data unit with the largest λ_l on the list. The problem is to compute λ_l for each data unit. Through a series of approximations documented in [55], λ_l can be estimated as 0 for any data unit l that 1) is within a forward trip time (e.g., $\mu_F + 3\sigma_F$) of its deadline, 2) has already been transmitted and has been acknowledged, or 3) has already been transmitted within a round trip time (e.g., $\mu_R + 3\sigma_R$) of the current time. For the other data units, λ_l can be estimated as S_l/B_l multiplied by a factor of ϵ_F/ϵ_R for each unacknowledged transmission. Hence, approximate rate-distortion optimal streaming can be achieved at very low computational complexity. Indeed using these approximations we have prototyped a rate-distortion optimized streaming media server running in Java that requires only about one percent of the CPU on a 700 MHz Pentium III, for a 40 Kbps audio stream.

This system is our rate-controlled RaDiO system, which we evaluate in the next section.

VI. Experimental Results

In this section, we report our experimental results only for the scenario of sender-driven transmission over a single-QoS network with retransmissions. First we examine in detail error-cost optimized transmission of a single data unit, and later examine rate-distortion optimized streaming of an entire audio presentation. Throughout the section we assume that each data unit has transmission opportunities every T seconds, at sender times $s_0, s_1, \ldots, s_{N-1}$ with delivery deadline $s_{\mathrm{DTS}} = s_N$ where $s_n = s_0 + nT$ for $n = 1, \ldots, N$. Thus $W = \mathrm{NT}$ is the size of the window of transmission opportunity for each data unit.

The error-cost function for transmission of a single data unit was shown in Figs. 3(e) and 5 of Section III for the parameters shown in Fig. 5. Fig. 5 shows the error-cost function on a log-linear scale, with each vertex of its convex hull labeled by the sequence of actions $[a_0, a_1, \ldots, a_7]$ for the optimal policy π_λ^* corresponding to the Lagrange multiplier λ for that vertex. By following policy $\pi_{\lambda_4}^*$, which transmits up to three times at intervals of $3T$ within a window of duration $8T$, it is possible to reduce the late/loss probability to less than one percent at an expected cost of only about 1.4 transmitted packets per data unit. The transmission interval $3T$ is equal to a slightly aggressive timeout interval: the mean RTT ($\kappa_R + n_R/\alpha_R = 2T$) plus two times the standard deviation ($2\sqrt{n_R}/\alpha_R = T$).

It is natural to ask whether extending the window size larger than $8T$, which is four times the mean RTT, can allow arbitrary further reductions in the late/loss probability, or whether the reductions saturate. It is also natural to ask, for a *fixed* window size W, whether the late/loss probability can be improved by increasing the density of transmission opportunities. In [55] we show that improvement saturates for window sizes larger than four times the mean RTT, and for transmission opportunity densities higher than twice per mean RTT. Hence, in subsequent experiments we set T to be half the mean RTT and we set $N = 8$.

Now we consider the overall distortion-rate performances of various systems when streaming one minute of packetized audio content. The audio content, the first minute of Sarah McLachlan's *Building a Mystery*, is compressed using a scalable version of the Windows Media Audio codec. The codec performs perceptual weighting on lapped orthogonal transform coefficients, followed by bitplane coding to produce an embedded bit string for each group of frames (GOF) of duration about 0.75 s. The bit string for each GOF is partitioned into segments of length 500 bytes, and packetized into data units. Twelve 500-byte data units are kept for each GOF, for a maximum bit rate of $12 \times 500 \times 8/0.75 = 64$ Kbps. The 12 data units per GOF are sequentially dependent, as shown in Fig. 1(a). Each data unit l is labeled by the decrease Δd_l in the perceptually weighted squared error if the data unit is decoded on time and all of its predecessors in the same GOF are decoded on time. All 12 data units in the Mth GOF receive the same decoding timestamp, equal to $0.75M$.

We compare several streaming systems. All of the systems use the same playback delay $\delta = 1$ s and the same transmission buffer size, which ramps from 0 to 5 s during the clip according to the functions $t_{\mathrm{lag}}(s)$ and $t_{\mathrm{lead}}(s)$ as specified in Section V. The systems also use the same channel parameters, which are shown in Fig. 8. Transmitted packets are dropped at random, and those not dropped receive a random delay, using a pseudo-random number generator. The pseudo-random number generator is initialized to the same seed for each of the systems compared. For each system at each transmission rate, performance is averaged over ten runs to smooth out the effects of particular channel realizations. Fig. 8 shows, for each of the systems compared, the signal-to-noise ratio in dB of the end-to-end perceptual distortion as a function of the transmission rate (in Kbps) averaged over the one minute audio clip. We now describe the compared systems in detail.

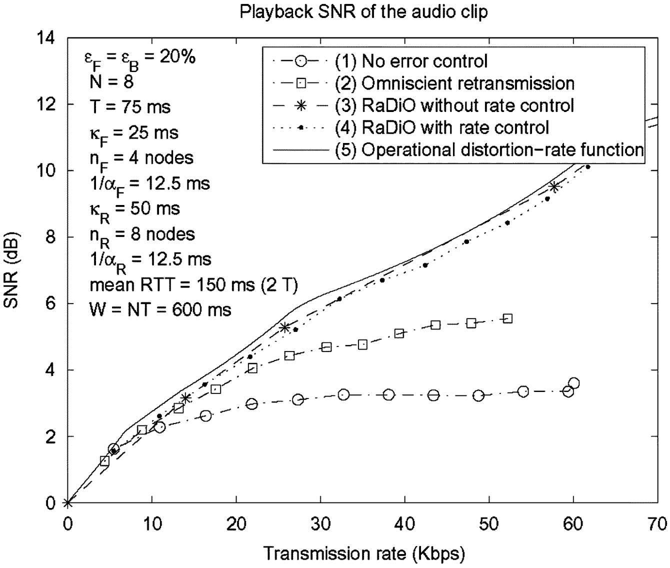


Fig. 8. Layered audio transmission.

System 1: No error control. In this system, there is no error control. Data units are transmitted at most once, in GOF order. A data unit is transmitted only if all of its predecessors in the same GOF are also transmitted. The number of data units transmitted in each GOF is proportional to the transmission rate. As shown in Fig. 8, the performance of this system saturates as the transmission rate increases. This is because in the absense of error control, base layer packets are being lost 20% of the time, limiting overall performance, regardless of the transmission rate. This shows the need for some sort of error control.

System 2: Omniscient retransmission. In this system, error control is provided by retransmissions, which may occupy up to 20% of the channel bandwidth (equal to the packet loss probability). Data units for which the server receives negative acknowledgment (NAKs) from the client are queued, and are retransmitted from the queue on a space-available basis. However, data units that are still in the retransmission queue past their delivery deadlines are removed from the queue and are not retransmitted. The remaining 80% or more of the channel bandwidth is used for first-time transmission of data units in the same manner as in System 1. *Omniscient* refers to the manner in which the client sends NAKs. Commonly, a client will send a NAK whenever a packet sequence number is detected missing for more than some timeout interval, but more sophisticated strategies are possible and are implemented in commercial streaming media systems. Here, we provide an upper bound on the performance of any such system by simulating an omniscient, though unrealizable, strategy in which the client sends precisely one NAK for each packet that is lost, at precisely the moment that the packet would have arrived at the client if it had not been lost. As Fig. 8 shows, such a system with omniscient retransmission can perform about two dB better than a system without error control.

System 3: Rate-distortion optimization without rate control. In this system, rate-distortion optimization using the ISA algorithm is applied without rate control to the scheduling of packet transmissions at the sender. Unlike System 2, no NAKs are available; only ACKs are sent back to the server upon receipt of a packet by the client. The server uses its history of previous transmissions as well as its history of acknowledgments to determine which packets to transmit (or retransmit) at each transmission opportunity. The Lagrange multiplier λ is fixed for the entire presentation. Hence, the number of data units selected at each transmission opportunity may vary, resulting in a variable transmission rate during the presentation. However, the *average* transmission rate during the presentation is well behaved and monotonically increases as λ decreases. As Fig. 8 shows, rate-distortion optimization without rate control outperforms the system with omniscient retransmissions by up to four or more dB, and outperforms the system without any error control by up to an additional two dB, for a total gain up to six or more dB.

System 4: Rate-distortion optimization with rate control. In this system, the approximations described in Section V are used to estimate for each data unit l the Lagrange multiplier λ_l above which the data unit is not selected for transmission. At each transmission opportunity, the data unit with the highest such λ_l is selected for transmission. The time until the next transmission opportunity is the size of the selected data unit divided by the desired transmission rate. As Fig. 8 shows, there is very little penalty (a fraction of a dB) for using this computationally efficient, constant-rate algorithm.

System 5: Rate-distortion bound. The lowest possible distortion at the receiver if it receives at most R_0 bits per second on average during a clip is given by the operational distortion-rate function $\hat{D}(R_0)$, which can be computed from the source sequence using the optimal pruning algorithm of [99]–[101]. Since with high probability $R = R_0/(1 - \epsilon_F)$ bits per second must be transmitted by the sender for the receiver to receive R_0 bits per second ($(1 - \epsilon_F)$ being the capacity of an erasure channel with loss probability ϵ_F), $\hat{D}((1 - \epsilon_F)R)$ is the the lowest possible distortion if the sender transmits R bits per second on average during the clip. We plot the corresponding signal-to-noise ratio in Fig. 8, which shows that no streaming system can achieve substantially better than the systems presented in this paper.

VII. Conclusion

This paper develops a framework for rate-distortion optimized streaming, and within the framework develops practical systems that essentially achieve the operational distortion-rate function of the source at the capacity of the channel. Thus, there do not exist systems that can perform significantly better.

One of the main lessons in the paper is that rate-distortion optimized streaming of an entire presentation can be solved by focusing on error-cost optimized transmission of a single data unit in isolation. The set of choices for transmitting a single data unit is the only difference between widely different transmission scenarios, and hence rate-distortion optimized streaming can be easily extended to different transmission scenarios, including sender-driven or receiver-driven transmission over best-effort networks, multiple overlay networks, integrated or differentiated services networks, combined wireline/wireless networks, and multiple access networks.

References

[1] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," in *Proc. Data Compression Conf.*. Snowbird, UT, Mar. 2000, pp. 440–449.

[2] ——, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 108–122, Mar. 2001.

[3] Z. Miao and A. Ortega, "Optimal scheduling for streaming of scalable media," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Pacific Grove, CA, Nov. 2000, pp. 1357–1362.

[4] Z. Miao, "Algorithms for streaming, caching and storage of digital media," Ph.D. dissertation, Univ. Southern California, Los Angeles, May 2002.

[5] Z. Miao and A. Ortega, "Expected run-time distortion based scheduling for delivery of scalable media," in *Proc. Int. Packet Video Workshop*, Pittsburgh, PA, Apr. 2002.

[6] J. Zhou and J. Li, "Scalable audio streaming over the Internet with network-aware rate-distortion optimization," in *Proc. IEEE Int. Conf. Multimedia Exh. (ICME)*, Tokyo, Japan, Aug. 2001.

[7] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," Univ. California, Comput. Sci. Div., Berkeley, Tech. Rep. UCB/CSD-98-1024, Nov. 1998.

[8] ——, "Soft ARQ for layered streaming media," *J. VLSI Signal Process. Syst. Signal, Image Video Technol., Special Issue on Multimedia Signal Processing*, vol. 27, no. 1–2, pp. 81–97, Feb. 2001.

[9] V. Chande, H. Jafarkhani, and N. Farvardin, "Joint source-channel coding of images for channels with feedback," in *Proc. IEEE Information Theory Workshop*, San Diego, CA, Feb. 1998.

[10] S. D. Servetto, "Compression and reliable transmission of digital image and video signals," Ph.D. dissertation, Univ. Illinois, Urbana-Champaign, 1999.

[11] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[12] C.-Y. Hsu, A. Ortega, and A. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 5, pp. 1016–1028, Aug. 1997.

[13] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding of video signals over ATM networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1002–1015, Aug. 1997.

[14] D. A. Turner and K. W. Ross, "Optimal streaming of layer-encoded multimedia presentations," in *Proc. IEEE ICME*, New York, NY, Jul. 2000.

[15] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTF," in *First int. Workshop on High Speed Networks and Open Distributed Platforms*, St. Petersburg, Russia, Jun. 1995.

[16] H. Song, J. Kim, and C.-C. J. Kuo, "Real-time encoding frame rate control for H.263+ video over the Internet," *Signal Process.: Image Commun.*, vol. 15, no. 1–2, pp. 127–148, Sep. 1999.

[17] K. Chawla, Z. Jiang, X. Qiu, and A. Reibman, "Transmission of streaming video over an EGPRS wireless network," in *Proc. IEEE ICME*, New York, Jul. 2000.

[18] A. Reibman, Y. Wang, X. Qiu, Z. Jiang, and K. Chawla, "Transmission of multiple description and layered video over an EGPRS wireless network," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 2, Vancouver, BC, Canada, Oct. 2000, pp. 136–139.

[19] T. Tian, A. H. Li, J. Wen, and J. D. Villasenor, "Prority dropping in network transmission of scalable video," in *Proc. ICIP*, vol. 3, Vancouver, BC, Canada, Oct. 2000, pp. 400–403.

[20] P.-C. Hu, Z.-L. Zhang, and M. Kaveh, "Channel condition ARQ rate control for real-time wireless video under buffer constraints," in *Proc. ICIP*, vol. 2, Vancouver, BC, Canada, Oct. 2000, pp. 124–127.

[21] 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.

[22] 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 Process.*. Denver, CO, Aug. 1996.

[23] J. M. Boyce, "Packet loss resilient transmission of MPEG video over the Internet," *Signal Process.: Image Commun.*, vol. 15, no. 1–2, pp. 7–24, Sep. 1999.

[24] U. Horn, K. Stuhlmiiller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Process.: Image Commun.*, vol. 15, no. 1–2, pp. 77–94, Sep. 1999.

[25] 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, pp. 92–101.

[26] ——, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–829, Jun. 2000.

[27] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. Asilomar Conf Signals, Systems, and Computers*, vol. 1. Asilomar, CA, Oct. 1999, pp. 342–346.

[28] W. Zhu, Q. Zhang, and Y.-Q. Zhang, "Network-adaptive rate control with unequal loss protection for scalable video over Internet," in *Proc. IEEEInt. Symp. Circuits and Systems*, Sydney, NSW, May 2001.

[29] Q. Zhang, G. Wang, W. Zhu, and Y.-Q. Zhang, "Robust scalable video streaming over Internet with network-adaptive congestion control and unequal loss protection," in *Proc. EURASIP/IEEEInt. Packet Video Workshop*, Kyongju, Korea, Apr. 2001.

[30] P. A. Chou and K. Ramchandran, "Clustering source/channel rate allocations for receiver-driven multicast with error control under a limited number of streams," in *Proc. ICME*, vol. 3. New York, Jul. 2000, pp. 1221–1224.

[31] A. E. Mohr, R. E. Ladner, and E. A. Riskin, "Approximately optimal assignment for unequal loss protection," in *Proc. ICIP*. Vancouver, BC, Canada, Sep. 2000.

[32] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," in *Proc. SPIE Visual Commun. and Image Process.*, vol. 3974, San Jose, CA, Jan. 2000, pp. 300–311.

[33] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," in *Proc. EURASIP/IEEEInt. Packet Video Workshop*, Kyongju, Korea, Apr. 2001.

[34] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," in *Proc. ICIP*, Rochester, NY, Sep. 2002.

[35] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 230–239, Apr. 2004.

[36] J. Lu, A. Nosratinia, and B. Aazhang, "Progressive source-channel coding of images over bursty error channels," in *Proc. ICIP*, Chicago, IL, Oct. 1998.

[37] M. J. Ruf and J. W. Modestino, "Operational rate-distortion performance for joint source and channel coding of images," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 305–320, Mar. 1999.

[38] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 850–860, Jun. 2000.

[39] J.-C. Bolot and A. Vega-Garcia. The case for FEC-based error control for packet audio in the Internet. [Online]. Available: http://www-sop.inria.fr/rodeo/personnel/bolot/papers.html

[40] M. Podolsky, C. Romer, and S. McCanne, "Simulation of FEC-based error control for packet audio on the internet," in *Proc. IEEE Infocom*, San Francsico, CA, Mar. 1998.

[41] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for interactive audio on the Internet," in *Proc. IEEE Infocom*, New York, Mar. 1999.

[42] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," in *Proc. SPIE Visual Commun. and Image Process.*, San Jose, CA, Jan. 1999.

[43] A. C. Miguel, A. E. Mohr, and E. A. Riskin, "SPIHT for generalized multiple description coding," in *Proc. ICIP*, Kobe, Japan, Oct. 1999.

[44] S. Mehrotra, "Multiple description coding using overcomplete linear expansions," Ph.D. dissertation, Stanford Univ., Stanford, CA, Jun. 2000.

[45] C. Papadopoulos and G. M. Parulkar, "Retransmission-based error control for continuous media applications," in *Proc. Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Zushi, Japan, Jul. 1996.

[46] G. Carle and E. W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," *IEEE Network Mag.*, vol. 11, no. 6, pp. 24–36, Nov. 1997.

[47] X. Li, S. Paul, P. Pancha, and M. H. Ammar, "Layered video multicast with retransmissions (LVMR)," in *Proc. Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, St. Louis, MO, May 1997.

[48] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable Internet video using MPEG-4," *Signal Process.: Image Commun.*, vol. 15, no. 1–2, pp. 95–126, Sep. 1999.

[49] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 923–941, Sep. 2000.

[50] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet" challenges and approaches," in *Proc. IEEE*, vol. 88, Dec. 2000, pp. 1855–1875.

[51] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "An efficient transport scheme for multimedia over wireless Internet," in *Proc. Int. Conf. Third Generation Wireless and Beyond*, San Francisco, CA, May 2001.

[52] S. D. Servetto, K. Ramchandran, K. Nahrstedt, and A. Ortega, "Optimal segmentation of a VBR source for its parallel transmission over multiple ATM connections," in *Proc. ICIP*, Santa Barbara, CA, Oct. 1997.

[53] V. Padmanabhan, "Using differentiated services mechanisms to improve network protocol and application performance," in *SPIE RTAS Workshop on QoS Support for Real-Time Internet Applications*, Vancouver, BC, Canada, Jul. 1999.

[54] J. Shin, J. Kim, and C.-C. J. Kuo, "Relative priority based QoS interaction between video applications and differentiated service networks," in *Proc. ICIP*, vol. 3, Vancouver, BC, Canada, Oct. 2000, pp. 536–539.

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

[56] J. Chakareski and P. A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," in *Proc. Int. Conf. Acoustics Speech, and Signal Process.*, vol. 3, Orlando, FL, May 2002, pp. 2513–2516.

[57] J. Chakareski, P. A. Chou, and B. Aazhang, "Computing rate-distortion optimized policies for streaming media to wireless clients," in *Proc. Data Compression Conf.*. Snowbird, UT, Apr. 2002, pp. 53–62.

[58] J. Chakareski and P. A. Chou, "Application Layer Error Correction Coding for Rate-Distortion Optimized Streaming to Wireless Clients," Microsoft Research, Redmond, WA, ftp://ftp.research.microsoft.com/pub/tr/TR-2002-81.ps, Tech. Rep. MSR-TR-2002-81, Aug. 2002. Publicly available at.

[59] ——, (2002, Aug.) Application layer error correction coding for rate-distortion optimized streaming to wireless clients. *IEEE Trans. Commun.*. [Online], vol (8), pp. 1675–1687

[60] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conf.*. Snowbird, UT, Mar. 2003, pp. 203–212.

[61] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proc. 11th ACM Int. Conf. on Multimedia*, Berkeley, CA, Nov. 2003, pp. 422–431.

[62] A. Sehgal and P. A. Chou, "Cost-distortion optimized streaming media over DiffServ networks," in *Proc. IEEE ICME*, vol. 1. Lausanne, Switzerland, Aug. 2002, pp. 857–860.

[63] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized streaming of multimedia," in *Proc. ICIP*, Barcelona, Catalonia, Spain, Sep. 2003.

[64] A. C. Begen, Y. Altunbasak, and M. A. Begen, "Rate-distortion optimized on-demand media streaming with server diversity," in *Proc. ICIP*, Barcelona, Catalonia, Spain, Sep. 2003.

[65] A. Sehgal and P. A. Chou, "Cost-distortion optimized caching of streaming media," in *Proc. Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 2, Orlando, FL, May 2002, pp. 1973–1976.

[66] J. Chakareski, P. A. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *Proc. Workshop on Multimedia Signal Process.*, S. Thomas, Ed., Dec. 2002, pp. 49–52.

[67] ——, "Computing rate-distortion optimized policies for hybrid receiver/sender driven streaming of multimedia," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Pacific Grove, CA, Nov. 2002, pp. 1310–1314.

[68] ——, RaDiO Edge: Rate-distortion optimized proxy-driven streaming from the network edge, in IEEE/ACM Trans. Networking. Submitted.

[69] M. Kalman, E. Steinbach, and B. Girod, "R-D optimized media streaming enhanced with adaptive media playout," in *Proc. IEEE ICME*, vol. 1, Rochester, NY, Aug. 2002, pp. 869–872.

[70] ——, "Rate-distortion optimized video streaming with adaptive playout," in *Proc. ICIP*, vol. 3, Rochester, NY, Sep. 2002, pp. 189–192.

[71] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," in *Proc. ICIP*. Barcelona, Catalonia, Spain, Sep. 2003.

[72] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming with rich acknowledgments," in *Proc. SPIE Visual Commun. and Image Process.*, San Jose, CA, Jan. 2004.

[73] R. Zhang, S. L. Regunathan, and K. Rose, "End-to-end distortion estimation for RD-based robust delivery of pre-compressed video," in *Conf. Rec. 35th Asilomar Conf. Signals, Systems and Computers*, Asilomar, CA, Nov. 2001, pp. 210–214.

[74] ——, "Optimized video streaming over lossy networks with real-time estimation of end-to-end distortion," in *Proc. IEEE ICME*, vol. 1, Lausanne, Switzerland, Aug. 2002, pp. 861–864.

[75] G. Cheung and W.-T. Tan, "Directed acyclic graph based source modeling for data unit selection of streaming media over QoS networks," in *Proc. IEEE ICME*, vol. 2. Lausanne, Switzerland, Aug. 2002, pp. 81–84.

[76] P. Ramanathan, M. Kalman, and B. Girod, "Rate-distortion optimized streaming of compressed light fields," in *Proc. ICIP*, Barcelona, Catalonia, Spain, Sep. 2003.

[77] C.-L. Chang and B. Girod, "Rate-distortion optimized interactive streaming for scalable bitstreams of light fields," in *Proc. SPIE Visual Commun. and Image Process.*. San Jose, CA, Jan. 2004.

[78] B. Girod, M. Kalman, Y. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," *Wireless Commun. Mobile Comput.*, vol. 2, no. 6, pp. 549–552, Sep. 2002.

[79] D. Quaglia and J. C. de Martin, "Delivery of MPEG video streams with constant perceptual quality of service," in *Proc. ICME*, vol. 2, Lausanne, Switzerland, Aug. 2002, pp. 85–88.

[80] F. Zhai, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "A rate-distortion optimized error control scheme for scalable video streaming over the Internet," in *Proc. ICME*, Baltimore, MD, Jul. 2003.

[81] F. Zhai, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "A novel cost-distortion optimization framework for video streaming over differentiated services networks," in *Proc. ICIP*, Barcelona, Catalonia, Spain, Sep. 2003.

[82] ——, "Joint source coding and packet classification for real-time video transmission over differentiated services networks," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 716–726, Aug. 2004.

[83] T. Stockhammer, H. Jenkac, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 268–277, Apr. 2004.

[84] J. C. de Martin, "Source-driven packet marking for speech transmission over differentiated-services networks," in *Proc. ICASSP*, Salt Lake City, UT, May 2001.

[85] J. C. de Martin and D. Quaglia, "Distortion-based packet marking for MPEG video transmission over DiffServ networks," in *Proc. ICME*, Tokyo, Japan, Aug. 2001.

[86] F. De Vito, L. Farinetti, and J. C. De Martin, "Perceptual classification of MPEG video for differentiated-services communications," in *Proc. ICME*, vol. 1. Lausanne, Switzerland, Aug. 2002, pp. 141–144.

[87] W. Li and Y. Chen, *Experiment Result on Fine Granularity Scalability* Seoul, Korea, Mar. 1999. Contribution M4473, ISO/IEC JTC1/SC29/WG11.

[88] S. Li, F. Wu, and Y.-Q. Zhang, "Experimental Results with Progressive Fine Granularity Scalable (PFGS) Coding,", Noordwijkerhout, NL, Tech. Rep. m5742, Mar., 2000. ISO/IEC JTC1/SC29/WG11.

[89] H. Wang and A. Ortega, "Robust video communication by combining scalability and multiple description coding techniques," in *Proc. SPIE Symp. Electronic Imaging*, San Jose, CA, Jan. 2003.

[90] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," *Internetworking: Res. Exp.* , vol. 5, pp. 163–205, Dec. 1994.

[91] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*, 3rd ed. New York: McGraw-Hill, 1974.

[92] S. M. Ross, *Stochastic Processes*: Wiley, 1974.

[93] D. Gunarwardena, P. Key, and L. Massoulie, "Network characteristics: modeling, measurements and admission control," in *Proc. EEE/IFIP/ACM Int. Workshop on Quality of Service (IWQoS)*. Monterey, CA, Jun. 2003.

[94] M. Röder, J. Cardinal, and R. Hamzaoui, "On the complexity of rate-distortion optimal streaming of packetized media," in *Proc. Data Compression Conf.*. Snowbird, UT, Mar. 2004.

[95] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1987.

[96] "Video codec test model number 10 (TMN-10) (H.263+)," *ITU-T SG16/Q15 Doc. Q15-D-65*, Apr. 1998. ITU-T SG16/Q15.

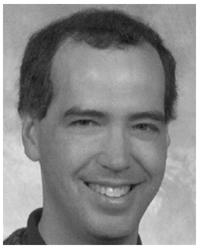
[97] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled lagrange multiplier method," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 546–588, Sep. 1994.

[98] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[99] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 1, pp. 1445–1453, Sep. 1988.

[100] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, no. 2, pp. 299–315, Mar. 1989.

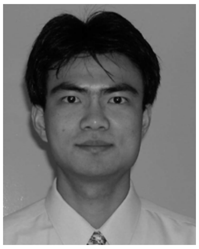
[101] E. A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 400–402, Mar. 1991.



Philip A. Chou (S'82–M'87–SM'00–F'04) received the B.S.E. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1980, the M.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 1983, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1988.

From 1988 to 1990, he was Member of Technical Staff with AT&T Bell Laboratories, Murray Hill, NJ. From 1990 to 1996, he was a Member of Research Staff with the Xerox Palo Alto Research Center, Palo Alto, CA. In 1997, he was Manager of the Compression Group, VXtreme, Mountain View, CA, before it was acquired by Microsoft in 1997. Since 1998 to the present, he has been a Senior Researcher with Microsoft Research, Redmond, WA. He also served as a Consulting Associate Professor at Stanford University, Stanford, CA, from 1994 to 1995 and has been an Affiliate Professor with the University of Washington, Seattle, since 1998. His research interests include data compression, information theory, communications, and pattern recognition, with applications to video, images, audio, speech, and documents.

Dr. Chou was the recipient of the 2002 ICME Best Paper Award with A. Seghal and the 1993 Signal Processing Society Paper Award with T. Lookabaugh. He served as an Associate Editor in source coding for the IEEE Transactions on Information Theory from 1998 to 2001 and served as a Guest Associate Editor for special issues in the IEEE Transactions on Image Processing and the IEEE Transactions on Multimedia in 1996 and 2004, respectively. From 1998 to 2004, he was a member of the IEEE Signal Processing Society's Image and Multidimensional Signal Processing Technical Committee (IMDSP TC). He is a member of Phi Beta Kappa, Tau Beta Pi, Sigma Xi, and the IEEE Computer, Information Theory, Signal Processing, and Communications Societies and was an active member of the MPEG Committee.



Zhourong Miao (M'02) was born in Shanghai, China, in 1974. He received the B.S. degree in electrical engineering from Shanghai Jiao Tong University in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1999 and 2002, respectively.

He was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, in summer 1999 and the Microsoft Research Center, Redmond, WA, in summer 2000 as a Research Intern. He is currently with the Sony Research Lab, Santa Clara, CA. His research interests include video compression and communication, proxy caching for streaming video and multimedia networking.