SERVER DIVERSITY IN RATE-DISTORTION OPTIMIZED MEDIA STREAMING

Jacob Chakareski[*] and Bernd Girod

Information Systems Laboratory, Department of Electrical Engineering, Stanford University
{cakarz,bgirod}@stanford.edu

ABSTRACT

We consider diversity for media streaming in a receiver-driven rate-distortion optimization framework. Diversity is achieved by requesting media packets from multiple servers. A framework is proposed that enables the receiver to decide at every instant which packets, if any, to request for transmission and from which servers in order to meet a rate constraint while minimizing the end-to-end distortion. Experimental results demonstrate the benefit of exploiting server diversity in rate-distortion optimized receiver-driven streaming of packetized media.

1. INTRODUCTION

This paper addresses the problem of streaming packetized media over a lossy packet network from multiple servers to a single client, in a rate-distortion optimized way. It is assumed that the servers are located at different locations in the network and communicate with the client over independent network paths. Packets may be lost in any of the paths due to congestion or erasures. The problem under consideration is illustrated in Figure 1.

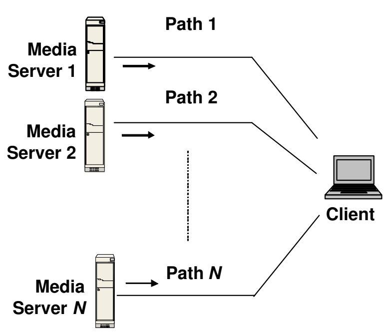

Fig. 1. Distributed streaming from multiple servers.

We setup a general framework for receiver-driven rate-distortion optimized streaming of packetized media from multiple servers to a single client. Media packets are typically characterized by different deadlines, importances and interdependencies. Using this information and the proposed framework, the client is able to request transmission of media packets from the multiple servers, based on the feedback it receives from the servers, in a rate-distortion optimized way, that is, minimizing the expected end-to-end distortion subject to a constraint on the expected overall transmission rate from the servers to the client. Such a rate-distortion optimized transmission algorithm, or transmission policy, results in unequal error protection provided to different portions of the media stream. The core step of the optimization framework involves trading off the expected redundancy (the cost used to communicate a media packet) for the probability that the packet will be communicated in error.

To our knowledge, the most closely related contemporaneous works are the following. In [1] a receiver-driven framework, based on Forward Error Correction (FEC) codes, is proposed for distributed streaming of video from multiple senders to a single receiver. Similarly in [2] the authors study a receiver-driven distributed streaming scenario based on Multiple Description (MD) codes. In addition, [3] presents a framework for rate-distortion optimized receiver-driven streaming over a single network path, modeled as an independent time-invariant packet erasure channel. Finally, in [4] the authors present a framework for rate-distortion optimized packet scheduling and routing for sender-driven media streaming with path diversity.

2. PACKET LOSS PROBABILITIES

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. All of the data units in the presentation have interdependencies, which can be expressed by a directed acyclic graph. Associated with each data unit l is a size B_l, a decoding time $t_{DTS,l}$, and an importance Δd_l. The size B_l is the size of the data unit in bytes. $t_{DTS,l}$ is the *delivery deadline* by which data unit l must arrive at the client, or be too late to be usefully decoded. Packets containing data units that arrive after the data units' delivery deadlines are discarded. The importance Δd_l is the amount by which the distortion at the client will *decrease* if the data unit arrives on time at the client and is decoded.

We model the network path between a server and a client, as a burst loss channel using a K-state discrete-time Markov model. The forward and the backward channel make state transitions independently of each other every T seconds, where the transitions are described by probability matrices $\mathcal{P}_{(F)}$ and $\mathcal{P}_{(B)}$, respectively. In each state the forward and the backward channel are characterized as an independent time-invariant packet erasure channel with random delay. Hence, they are completely specified with the probability of packet loss $\epsilon^k_{F/B}$ and the probability density of the transmission delay $p^k_{F/B}$, for $k = 1, \ldots, K$. This means that if the server sends, as a response to a request, a packet with the requested data unit on the forward channel at time t, given that the forward channel is in state k at t, then the packet is lost with probability ϵ^k_F. However, if the packet is not lost, then it arrives at the client at time t', where the forward trip time $FTT^k = t' - t$ is randomly drawn according to the probability density p^k_F. Therefore, we let $P\{FTT^k > \tau\} = \epsilon^k_F + (1-\epsilon^k_F)\int_\tau^\infty p^k_F(t)dt$ denote

[*]On leave from Rice University, Houston, TX 77005.

the probability that a data unit transmitted by the server at time t, given that the forward channel is in state k at t, does not arrive at the client application by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ. Then similarly, $P\{BTT^k > \tau\} = \epsilon_B^k + (1-\epsilon_B^k)\int_\tau^\infty p_B^k(t)dt$ denotes the probability that a request packet transmitted by the client at time t, given that the backward channel is in state k at t, does not arrive at the server by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ. Finally, we are interested in $P\{RTT^{kj} > \tau\}$, which is the probability that the client does not receive the requested data unit by time $t+\tau$ for a request packet transmitted at time t, given that the forward and the backward channel are respectively in states j and k, at t.

To derive $P\{RTT^{kj} > \tau\}$ assume first that the transmission on the backward channel occured immediately after the channel made a state transition. If $BTT^k \leq T$, the request is received by the server before the forward channel makes the next state transition. Then $P\{RTT^{kj} > \tau | BTT^k \leq T\} = P\{BTT^k + FTT^j > \tau | BTT^k \leq T\}$ as the server sends the requested data unit while the forward channel is still in the current state j. The probability of this event is $P\{BTT^k \leq T\}$. However, if $lT < BTT^k \leq (l+1)T$, for $l \geq 1$, then the state of the forward channel makes l transitions before the request actually arrives at the server. The probability of this event is $P\{lT < BTT^k \leq (l+1)T\}$. Here the state on the forward channel when the acknowledgement is sent can be any of the K possible values. Hence we compute the desired quantity as the expected value over all of them, i.e., $P\{RTT^{kj} > \tau | lT < BTT^k \leq (l+1)T\} = \sum_{p=1}^K \mathcal{P}_{jp(F)}^{(l)} P\{BTT^k + FTT^p > \tau | lT < BTT^k \leq (l+1)T\}$. Note that $\mathcal{P}_{jp(F)}^{(l)}$ is the probability of making a transition from state j to state p in l transition intervals. These probabilities are obtained using matrix power, i.e., $\mathcal{P}_{(F)}^{(l)} = \mathcal{P}_{(F)}^{l}$. Finally, we obtain $P\{RTT^{kj} > \tau\}$ by averaging over all possible outcomes for BTT^k.

3. RATE-DISTORTION OPTIMIZED POLICY SELECTION

Suppose there are L data units in the media presentation. Let $\pi_l \in \Pi$ 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. Π is a family of policies defined precisely in the next section.

Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion $D(\boldsymbol{\pi})$ and an expected transmission rate $R(\boldsymbol{\pi})$ for the media presentation. We seek the policy vector $\boldsymbol{\pi}$ that minimizes $D(\boldsymbol{\pi})$ subject to a constraint on $R(\boldsymbol{\pi})$. This can be achieved by minimizing the Lagrangian $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$ for some Lagrange multiplier $\lambda > 0$, thus achieving a point on the lower convex hull of the set of all achievable distortion-rate pairs.

We now compute expressions for $R(\boldsymbol{\pi})$ and $D(\boldsymbol{\pi})$. The expected transmission rate $R(\boldsymbol{\pi})$ is the sum of the expected number of bytes transmitted for each data unit $l \in \{1,\ldots,L\}$, $R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l)$, where B_l is the number of bytes in data unit l and $\rho(\pi_l)$ is the expected number of transmitted bytes per source byte (under policy π_l), called the *expected cost*. The expected distortion $D(\boldsymbol{\pi})$ can be expressed in terms of the probability $\epsilon(\pi_l)$ that data unit l does not arrive at the receiver on time (under policy π_l), called the *expected error*. We borrow the expression for $D(\boldsymbol{\pi})$ from [4]

$$D(\boldsymbol{\pi}) = D_0 - \sum_l \sum_{l_1 \in N_c^{(l)}} \Delta d_l^{(l_1)} \prod_{j \in A(l_1)} (1-\epsilon(\pi_j)) \times \prod_{l_2 \in C(l,l_1)} \left(1 - \prod_{l_3 \in A(l_2)\setminus A(l_1)} (1-\epsilon(\pi_{l_3}))\right) \quad (1)$$

where $N_c^{(l)} = \{1,\ldots,l\}$ is the set of data units that the receiver considers for error concealment in case data unit l is not decodable by the receiver on time. $\Delta d_l^{(l_1)}$, for $l_1 \in N_c^{(l)}$, is the reduction in distortion if data unit l is not decodable and is concealed with a previous data unit l_1 that is received and decoded on time. $A(l_1)$ is the set of ancestors of l_1, including l_1. $C(l,l_1)$ is the set of data units $j \in N_c^{(l)} : j > l_1$ that are not mutual descendants, i.e., for $j,k \in C(l,l_1) : j \notin D(k), k \notin D(j)$, where $D(j)$ is the set of descendants of data unit j. "\" denotes the operator "set difference". In deriving (1), we assume statistical independence of the losses affecting separate data units for tractability. For a further discussion, see [4].

Finding a policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian $J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$, for $\lambda > 0$, is difficult since the terms involving the individual policies π_l in $J(\boldsymbol{\pi})$ are not independent. Therefore, we employ an iterative descent algorithm, called Iterative Sensitivity Adjustment (ISA), in which we minimize the objective function $J(\pi_1,\ldots,\pi_L)$ one variable at a time while keeping the other variables constant, until convergence [5]. It can be shown that the optimal individual policies at iteration n, for $n = 1,2,\ldots$, are given by

$$\pi_l^{(n)} = \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l), \quad (2)$$

where $S_l^{(n)} = \sum_{l_1 \,:\, l \in N_c^{(l_1)}} S_{l,l_1}^{+(n)} - S_{l,l_1}^{-(n)} = S_l^{+(n)} - S_l^{-(n)}$ 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 client, given the current transmission policies for the other data units. Note that differently from [5], the sensitivity here consists of two nonnegative terms $S_l^{+(n)}$ and $S_l^{-(n)}$. The first term increases the sensitivity associated with data unit l in case l is in the ancestor set of data unit l_2 used for concealment of a data unit l_1. On the other hand, the second term reduces the sensitivity associated with l in case l is not in the ancestor set of l_2. This result is intuitive and allows us to better model the situations where data unit l is irrelevant for concealment of another data unit. Expressions for $S_{l,l_1}^{+(n)}$ and $S_{l,l_1}^{-(n)}$ are easily obtained from (1) by grouping terms.

The minimization (2) is now simple, since each data unit l can be considered in isolation. Indeed the optimal transmission policy $\pi_l \in \Pi$ for data unit l minimizes the "per data unit" Lagrangian $\epsilon(\pi_l) + \lambda' \rho(\pi_l)$, where $\lambda' = \lambda B_l / S_l^{(n)}$. Thus to minimize (2) for any l and λ', it suffices to know the lower convex hull $\epsilon(\rho) = \min_{\pi \in \Pi}\{\epsilon(\pi) : \rho(\pi) \leq \rho\}$ of the function, which we call the expected *error-cost* function. In the next section we show how to compute the expected error-cost function for the family of transmission policies corresponding to receiver-driven streaming from multiple servers.

4. COMPUTING THE EXPECTED ERROR-COST FUNCTION

Assume that there are M media servers at different locations in the network with which a client communicates over M independent network paths. Furthemore, assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to a data unit's delivery deadline t_{DTS} at which the client is allowed to transmit a request packet for the data unit on the backward channel of any $m \leq M$ paths. The client need not transmit a request at every transmission opportunity. The client does not transmit any further requests after a packet with the data unit arrives on the forward channel of any of the paths.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the client takes an action $a_i = [a_{i1}, \ldots, a_{iM}]$, where $a_{im} = 1$ means that a request is sent on the backward channel of path m and $a_{im} = 0$ means that no request is sent on the backward channel of path m. Then, at the next transmission opportunity t_{i+1}, the client makes an observation o_i, where o_i is the set of packets received by the client in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{DAT_{j_1}^{m_1}, DAT_{j_2}^{m_2}\}$ means that during the interval $(t_i, t_{i+1}]$, packets with the data unit arrived on the forward channels as a response to the requests sent at time t_{j_1} and t_{j_2} on the backward channels of paths m_1 and m_2, respectively. The history, or the sequence of action-observation pairs $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ leading up to time t_{i+1}, determines the state q_{i+1} at time t_{i+1}, as illustrated in Figure 2. If the final observation o_i includes a DAT, then q_{i+1} is a final state. In addition, any state at time $t_N = t_{DTS}$ is a final state. Final states in Figure 2 are indicated by double circles.

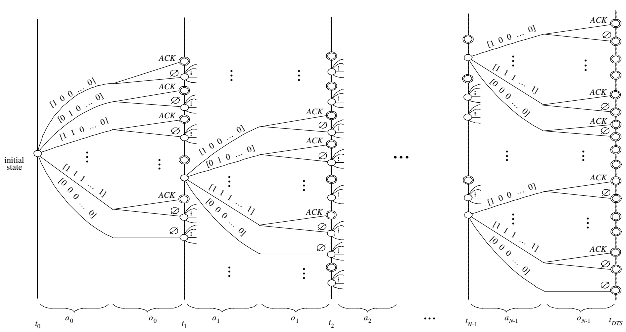

Fig. 2. Markov decision tree for a data unit with server diversity.

The action a_i taken at a non-final state q_i determines the transition probabilities $P(q_{i+1}|q_i, a_i)$ to the next state q_{i+1}. Formally, a policy π is a mapping $q \mapsto a$ from non-final states to actions. Thus any policy π induces a Markov chain with transition probabilities $P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i))$, and consequently also induces a probability distribution on final states. Let q_F be a final state with history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{F-1}, o_{F-1})$, and let $q_{i+1} = q_i \circ (a_i, o_i)$, $i = 1, \ldots, F-1$, be the sequence of states leading up to q_F. Then q_F has probability $P_\pi(q_F) = \prod_{i=0}^{F-1} P_\pi(q_{i+1}|q_i)$, transmission cost $\rho_\pi(q_F) = \sum_{i=0}^{F-1} \sum_{m=1}^{M} Cost(a_{im})$, and error $\epsilon_\pi(q_F) = 0$ if o_{F-1} contains a DAT and otherwise $\epsilon_\pi(q_F)$ is equal to the probability that none of the transmitted requests results in the data unit arriving at the client on time, given q_F. $Cost(a_{im})$ is the expected cost of transmitting the data unit from server m, as a response to the prospective request a_{im}, and is obtained using the Markov model for the backward channel of path m. Finally, we can express the expected cost and error for the Markov chain induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\epsilon_\pi(q_F)$.

We wish to find the policy π^* that minimizes $\epsilon(\pi) + \lambda' \rho(\pi)$, as discussed in the previous section. We do that by enumerating all possible policies π, plotting the error-cost performances $\{(\rho(\pi), \epsilon(\pi))\}$ in the error-cost plane, and producing an operational error-cost function for our scenario. At every transmission opportunity t_i we find π^*, where $\{(\rho(\pi), \epsilon(\pi)) : \pi \in \Pi\}$ is calculated conditioned on q_i and all the policies under consideration are consistent with the history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{i-1}, o_{i-1})$ leading up to state q_i at time t_i. Then, the client sets a_i to the first action $\pi^*(q_i)$ of π^*, and the procedure is repeated at each successive transmission opportunity until a final state is reached. In the following we describe briefly how the error-cost performances are computed. As explained earlier, $\epsilon(\pi)$ is simply the probability that all the transmitted requests from π as well as those from the transmission history do not result in the data unit arriving at the client on time. Furthermore, upon receipt of the data unit, the client truncates its transmission pattern and does not consider sending any request packets afterwards. Therefore, the cost for each transmission of a request $a_{jp} = 1 : j \in \{i, \ldots, N-1\}, p = 1, \ldots, M$ is equal to the probability that none of the previous requests results in the data unit arriving at the client by t_j times $Cost(a_{jp})$. Hence, $\rho(\pi)$ is simply the sum of the individual costs over all transmission opportunities and servers(paths). Due to space considerations, we omit here the expressions for $\epsilon(\pi)$ and $\rho(\pi)$.

5. EXPERIMENTAL RESULTS

Here we investigate the end-to-end distortion-rate performance of the proposed framework for streaming of packetized video content. The video content is a two layer SNR scalable representation of the sequence *Foreman*. Using H.263+ the first 130 frames of QCIF *Foreman* have been encoded into a base and enhancement layer with corresponding rates of 32 and 64 Kbps. The frame rate is 10 fps and the size of the Group of Pictures (GOP) is 10 frames, consisting of an I frame followed by 9 consecutive P frames. Performance is measured in terms of the luminance peak signal-to-noise ratio (Y-PSNR) in dB of the end-to-end perceptual distortion, averaged over the duration of the video clip, as a function of the overall bit rate available on the forward channel(s) of the network path(s) between the server(s) and the client.

In the experiments we use $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the playback delay. Furthermore, we employ a $K = 2$ state Markov model for each path. The model parameters are kept same over all paths and are specified in Table 1. In particular, in Table 1a we specify the delay and loss characteristics for a channel state. We keep the same characteristics for the forward and the backward channel. The delay density is modeled using a shifted Gamma distribution specified with three parameters: shift κ, mean μ and standard deviation σ. Finally, the state transitions are modeled using two parameters: the stationary probability of being in State 2, π_2, and the expected duration of stay in State 2, τ_2, once a transition is made to this state. We employ four sets of values for these parameters denoted Model 0 - 3 in Table 1b. Due to the selected values Models 0 - 3 cover a range of possibilities in terms of the loss and delay characteristics exhibited on a network path.

We study first the performance of the proposed framework as

	Loss	Delay		
	ϵ (%)	κ (ms)	μ (ms)	σ (ms)
State 1	3	25	75	50
State 2	15	25	275	250

(a) Loss and delay parameters.

	π_2	τ_2 (ms)
Model 0	0	0
Model 1	0.2	200
Model 2	0.5	1000
Model 3	0.8	2000

(b) State transitions.

Table 1. Network path characterization.

a function of the number of servers available. The state transitions are generated using Model 2 in these experiments. It can be

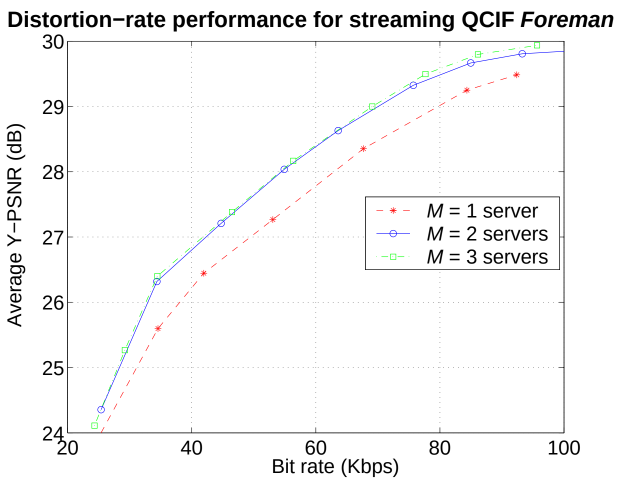

Fig. 3. R-D performance for $M = 1, 2$ and 3 servers.

seen from Figure 3 that streaming *Foreman* from two servers can improve performance compared to the case of streaming from a single server. An improvement is observed over the whole range of available rates. The gains in performance are most significant for the range of rates 30 - 90 Kbps and reach up to 0.6 dB. The difference in performance decreases as we move towards very low or very high transmission rates. The improved performance is due to the fact that having an alternative server for streaming reduces dramatically the probability of having to communicate with a server over a network path that features degraded quality (State 2) at transmission. This ultimately contributes to a higher likelihood of delivering the data units on time. Furthermore, it can be seen from Figure 3 that using further servers for streaming does not provide additional gains in performance, since the performances of $M = 3$ and of $M = 2$ are almost identical. As the likelihood of not delivering the data units on time to the client is already quite reduced for $M = 2$, streaming from yet one more server does not provide further benefits, given the selected path model.

Next, we study the performance of the framework as a function of the quality of the network paths between the client and the servers. In Figure 4 we show the performance for streaming *Foreman* from $M = 1$ and from $M = 2$ media servers in case of Models 0, 1 and 3. It can be seen that streaming from two servers does not offer any advantages in case of Model 0. This is expected,

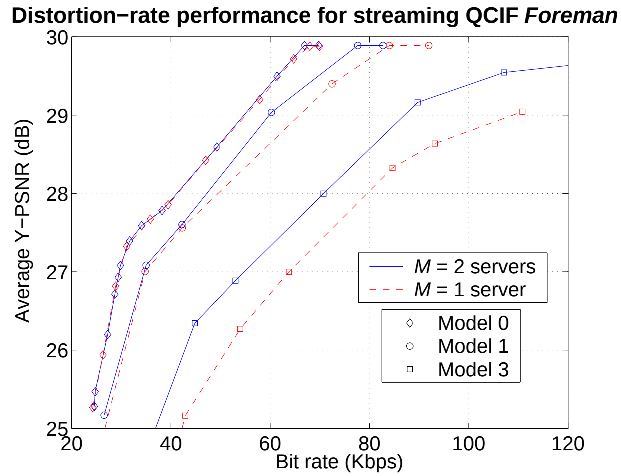

Fig. 4. R-D performance for $M = 1, 2$ and Models 0,1,3.

as the network paths here do not switch between states and hence there is no need for streaming from another server. However, as we move from Model 0 towards Model 3 the need for streaming from an alternative server, in order to avoid communicating with a server over a bad quality network path, steadily increases. Hence, the performance difference between $M = 2$ and $M = 1$ is largest when the state transitions on a path are governed by Model 3.

6. CONCLUSIONS

A framework has been presented that incorporates server diversity in a rate-distortion optimized receiver-driven streaming of packetized media. Using our framework a client can request media packets from multiple servers in order to obtain an improved performance over the case when only a single server is used. Experimental results for streaming video content demonstrate the benefit of using the proposed framework. The gains in performance are dependent on the quality of the network paths in terms of loss and delay.

7. REFERENCES

[1] T. Nguyen and A. Zakhor, “Distributed video streaming with forward error correction,” in *Proc. Int’l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[2] A. Majumdar, R. Puri, and K. Ramchandran, “Distributed multimedia transmission from multiple servers,” in *Proc. Int’l Conf. Image Processing*, vol. 3. Rochester, NY, USA: IEEE, Sept. 2002, pp. 177–180.

[3] P. A. Chou and A. Sehgal, “Rate-distortion optimized receiver-driven streaming over best-effort networks,” in *Proc. Int’l Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[4] J. Chakareski and B. Girod, “Rate-distortion optimized packet scheduling and routing for media streaming with path diversity,” in *Proc. Data Compression Conference*. Snowbird, UT: IEEE Computer Society, Apr. 2003, to appear.

[5] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Trans. Multimedia*, 2001, submitted.