Computing rate-distortion optimized policies for hybrid receiver/sender driven streaming of multimedia

Jacob Chakareski[*‡], Philip A. Chou[†] and Bernd Girod[‡]

[*]Rice University, Houston, TX 77005 USA

[†]Microsoft Corporation, Redmond, WA 98052-6399 USA

[‡]Information Systems Laboratory, Stanford University, Stanford, CA 94305 USA

Email: cakarz@stanford.edu, pachou@microsoft.com, bgirod@stanford.edu

Invited paper

ABSTRACT

We consider the problem of streaming packetized media over the Internet through an intermediate proxy server to a client, in a rate-distortion optimized way. The proxy employs a hybrid receiver/sender driven transmission scheme to communicate with both the media server and the client. Computing the optimal transmission policy for the proxy involves estimation of the probability that a single packet will be communicated to the client in error as a function of the expected redundancy (or cost) used to communicate the packet. In this paper, we show how to compute this error-cost function, and thereby optimize the proxy's transmission policy.

I. INTRODUCTION

We consider the problem of streaming packetized media over a lossy backbone packet network through a proxy server to a client. The proxy is located at the junction of the backbone network and the last hop to the client. Packets may be lost in the backbone network due to congestion, or in the last hop due to erasures.

The proxy employs a hybrid receiver/sender driven transmission scheme to communicate with the media server and with the client. The communication with the media server is receiver-driven, while the communication with the client is sender-driven. All packets sent in either direction are subject to random loss, delay, and corruption. Therefore the proxy can never be completely aware of the state of both the client and the media server. However, the proxy is aware of the different deadlines, importances, and dependencies of the various media packets to be transmitted. Furthermore, at every instance the proxy is aware of the different packets going in both directions that have reached so far the edge of the backbone network. Using this information the proxy is able to stream the media packets to the client, in a rate-distortion optimized way, that is, minimizing the expected end-to-end distortion subject to a constraint on the expected transmission rate. Such a rate-distortion optimized transmission algorithm, or transmission policy, results in unequal error protection provided to different portions of the media stream. To compute the rate-distortion optimized transmission policy, we use the Iterative Sensitivity Adjustment (ISA) algorithm introduced in [1]. The core step of this algorithm involves estimation of the probability that a single media packet will be communicated in error as a function of the expected redundancy, or cost, used to communicate the packet. The lower convex hull of the set of all expected error-cost pairs is called the error-cost function. How to compute this function, for the scenario of hybrid receiver/sender driven streaming, is the focus of this paper.

To our knowledge, the most closely related contemporaneous works are [1–3] where the authors have studied distortion-rate optimized streaming over lossy packet networks to wireline and to wireless clients, in both sender-driven and receiver-driven scenarios, and [4, 5] which considers proxy caching in a cost-distortion optimization framework. Another related work is [6] where the media server exploits feedback from a proxy to better adapt the media content sent to the client.

II. PRELIMINARIES

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. Associated with each data unit is a decoding time t_{DTS}. The decoding time t_{DTS} is the time at which the decoder is scheduled to extract the data unit from its input buffer and decode it. That is, t_{DTS} is the *delivery deadline* by which the data unit must arrive at the client in order to be usefully decoded. Packets containing data units that arrive after the data units' delivery deadlines are discarded.

We consider a system in which the media server communicates to the client indirectly, through a proxy. We refer to the server→proxy path as Channel 1 and to the proxy→client path as Channel 2. Each channel has a forward and backward direction. We model each direction of each channel as a time-invariant packet erasure channel with random delays. For the forward direction of Channel 1, this means that if the media server inserts a data packet into the network at time t, then the packet is lost with some probability, say ϵ_{F_1}, independently of t. However, if the packet is not lost, then it arrives at the proxy at some later time t', where the forward trip time $FTT_1 = t' - t$ is randomly drawn according to a probability density p_{F_1}. The backward direction of Channel 1 is similarly characterized by the probability of packet loss ϵ_{B_1} and delay density p_{B_1}. Successive losses and delays, as well as losses and delay in forward and backward channels are assumed to be statistically independent. Then, these induce the probability $\epsilon_{R_1} = 1 - (1 - \epsilon_{F_1})(1 - \epsilon_{B_1})$ of losing a packet in either the forward or backward direction, and the round trip time distribution $P\{RTT_1 > \tau\} = \epsilon_{R_1} + (1 - \epsilon_{R_1}) \int_\tau^\infty p_{R_1}(t)dt$, where $p_{R_1} = p_{F_1} * p_{B_1}$ is the convolution of p_{F_1} and p_{B_1}. Note that $P\{RTT_1 > \tau\}$ is the probability that a data packet requested from the media server by the proxy at time t does not arrive at the proxy by time $t + \tau$.

Each direction of Channel 2 is also modeled as an independent time-invariant packet erasure channel with random delays. Hence the forward and the backward directions are characterized by random loss and delay densities ϵ_{F_2}, p_{F_2} and ϵ_{B_2}, p_{B_2}, respectively. We let $P\{FTT_2 > \tau\} = \epsilon_{F_2} + (1 - \epsilon_{F_2}) \int_\tau^\infty p_{F_2}(t)dt$ denote the probability that a packet transmitted in the forward direction of Channel 2 at time t does not arrive at the client by time $t + \tau$, whether it is lost in the network, or simply delayed by more than τ. Finally, these in turn induce the probability ϵ_{R_2} of losing a packet in either the forward or backward directions of Channel 2 and the round trip time density p_{R_2}. Note that $P\{RTT_2 > \tau\}$ is the probability that the proxy does not receive an acknowledgement packet by time $t + \tau$ for a data packet sent to the client by the proxy at time t.

Next we define an *effective* channel, denoted *Channel 12*, which in essence corresponds to the network path from the proxy to the media server to the client (and back). Having such a channel facilitates the mathematical framework presented later on. The associated forward and round trip times for Channel 12 are $FTT_{12} = RTT_1 + FTT_2$ and $RTT_{12} = RTT_1 + RTT_2$, while the associated packet erasure rates are $\epsilon_{F_{12}} = \epsilon_{R_1} + (1 - \epsilon_{R_1})\epsilon_{F_2}$ and $\epsilon_{R_{12}} = \epsilon_{R_1} + (1 - \epsilon_{R_1})\epsilon_{R_2}$. Finally, in case of no packet loss, the forward trip and the round trip delays on Channel 12 are distributed according to the probability densities $f_{F_{12}} = f_{R_1} * f_{F_2}$ and $f_{R_{12}} = f_{R_1} * f_{R_2}$.

III. Hybrid Receiver/Sender Driven Transmission

A multimedia session starts when a client requests a presentation from the media server. The request packet is received by the proxy server and is not forwarded further. The proxy then sends a request to the media server for a rate-distortion preamble for the desired presentation. The preamble contains in effect the size, importance, and deadline information for each data unit.

After it has the preamble, the proxy starts communicating with both the media server and the client using a hybrid receiver/sender driven transmission scheme. The communication with the media server is receiver-driven, where the proxy sends request packets to the media server, requesting a particular data unit to be transmitted. The media server responds to a request by sending the requested data unit in a data packet to the proxy. The proxy may repeat requests for a particular data unit periodically if necessary, until the data unit finally arrives at the proxy, or until the proxy gives up. Upon seeing an arriving data packet, the proxy stores a copy of the data unit in its buffer for later retransmissions and immediately forwards it to the client. The proxy then stops requesting that data unit from the media server and can begin any necessary error recovery of the data unit using sender-driven communication with the client. The proxy performs error recovery for Channel 2 if necessary by retransmitting the buffered data unit periodically, until it receives an acknowledgement packet from the client for that data unit, or until the proxy gives up transmitting the data unit.

IV. R-D Optimization using the ISA Algorithm

The ISA algorithm has been explained elsewhere [1] and due to space constraints is not explained here. Its core step is finding a point on the lower convex hull of a set of points in the so-called "error-cost" plane, $\{(\rho(\pi), \epsilon(\pi)) : \pi \in \Pi\}$, where π is a *transmission policy* or protocol or algorithm for communicating a single data unit, Π is a family of transmission policies corresponding to the scenario at hand, $\epsilon(\pi)$ is the *expected error* or the probability that a data unit does not arrive at the client on time under policy π, and $\rho(\pi)$ is the *expected cost* or the expected number of transmitted bytes per source byte on Channel 2 under policy π. The point of interest corresponds to the policy π^* minimizing $\epsilon(\pi) + \lambda\rho(\pi)$, for a given Lagrange multiplier $\lambda > 0$. In Section VI we employ the above lower convex hull, called the *error-cost function*, to analyze the performance of different streaming systems. In the next section we show how to find π^*, for the family of transmission policies Π corresponding to hybrid receiver/sender driven transmission.

V. Finding the Optimal Policy

For requesting/transmitting a data unit, we assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to the data unit's delivery deadline t_{DTS} at which the proxy server is allowed either to request the data unit from the media server or to retransmit it to the client. The proxy need not transmit a packet at every transmission opportunity.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the proxy takes an action a_i, where $a_i = 1$ if the proxy sends a packet and $a_i = 0$ otherwise. Then, at the next transmission opportunity t_{i+1}, the proxy makes an observation o_i, where o_i is in fact the information collected by the proxy in the interval $(t_i, t_{i+1}]$ about previous transmissions. For example, $o_i = \{DAT_{j_1}, ACK_{j_2}\}$ means that during the interval $(t_i, t_{i+1}]$, a data packet arrived at the proxy for the request sent to the media server at time t_{j_1}, and an ACK packet arrived at the proxy for the packet sent at time t_{j_2}. Note that the acknowledgement can be due to a request or a data packet transmitted at t_{j_2}. 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 1. If the final observation o_i includes an ACK, 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 1 are indicated by double circles.

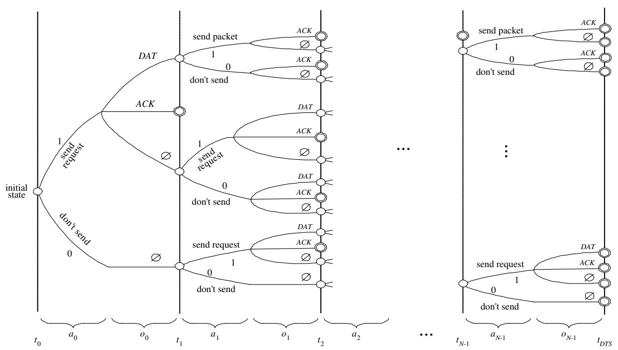

Fig. 1. Markov decision tree for one media unit in hybrid receiver/sender driven packet schedulling.

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}. For example, in Figure 1, if the action taken at the initial state q_0 is $a_0 = 1$ (send request), then the transition probabilities to the four states at time t_1 are respectively $P\{RTT_{12} > t_1 - t_0 \& RTT_1 \leq t_1 - t_0\}$, $P\{RTT_{12} \leq t_1 - t_0\}$, $P\{RTT_1 > t_1 - t_0\}$, and 0. (Note that $RTT_1 > t_1 - t_0$ implies that $RTT_{12} > t_1 - t_0$, since these are the events that the proxy server does not observe a response, respectively from the media server and from the client, by time t_1 for a request packet transmitted at time t_0.) On the other hand, if the action taken is $a_0 = 0$ (don't send), then the transition probabilities are respectively 0, 0, 0, and 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 $q_{i+1} = q_i \circ (a_i, o_i), i = 0, \ldots, F-1$, 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 $\rho_\pi(q_F) = \sum_{i=0}^{F-1} a_i Cost(a_i)$, and error $\epsilon_\pi(q_F) = 0$ if o_{F-1} contains an ACK and otherwise $\epsilon_\pi(q_F)$ is equal to the probability that none of the transmitted packets results in the data unit arriving at the client by time t_{DTS}, given q_F. Note that the quantity $Cost(a_i)$ is the expected cost of transmitting a packet from the server or retransmitting a packet from the proxy under the proxy action a_i, i.e., $Cost(a_i) = 1$ if $a_i = 1$ indicates transmission of a data packet from the proxy to the client, $Cost(a_i) = P\{RTT_1 < \infty\}$ if $a_i = 1$ indicates transmission of a request packet from the proxy to the server, and $Cost(a_i) = 0$ if $a_i = 0$. For example, if q_F is the second state from the top at time t_{DTS} in Figure 1, then a request packet was sent at t_0 and a data packet was sent afterwards at each transmission opportunity $t_i, i > 0$ and no acknowledgements were received. In that case, $\epsilon_\pi(q_F) = P\{FTT_{12} > t_{DTS} - t_0 | RTT_1 \leq t_1 - t_0, RTT_{12} > t_{DTS} - t_0\} \prod_{i=1}^{N-1} P\{FTT_2 > t_{DTS} - t_i | RTT_2 > t_{DTS} - t_i\}$.

Armed with definitions of probability, transmission cost, and error for each final state, we can now express the expected cost and error for the Markov chain induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\epsilon_\pi(q_F)$. At every transmission opportunity t_i the proxy finds the policy 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, a_i is set 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. Note that it would be sufficient to determine π^* only once at time t_0, except for the fact that λ may be adjusted (by the ISA algorithm [1]) at each iteration to take into account feedback from the transmission of other data units.

Two cases of transmission history can be distinguished: (1) no previous DATs, i.e., $o_0 = o_1 = \cdots = o_{i-1} = \emptyset$, and (2) with previous DATs. In the following we show how π^* can be determined for each of them.

A. No previous DATs

As explained earlier, depending on the information collected about previous transmissions, the proxy may transmit a request or a data packet at a transmission opportunity t_i, if the action taken at state q_i is $a_i = 1$. Therefore, the policies for this case of transmission history are necessarily subtrees of the Markov decision tree shown in Figure 1. Tree structured transmission policies in a rate-distortion optimization framework have been studied for the first time in [3, 7]. As stated earlier we are interested in the policy minimizing the expected Lagrangian

$$J(\pi) \equiv \epsilon(\pi) + \lambda\rho(\pi) = \sum_{q_F} P_\pi(q_F) J_\pi(q_F), \qquad (1)$$

where $J_\pi(q_F) \equiv \epsilon_\pi(q_F) + \lambda\rho_\pi(q_F)$. Let

$$J_\pi(q_i) = \begin{cases} \epsilon_\pi(q_F) + \lambda\rho_\pi(q_F) \\ \qquad \text{if } q_i \text{ is final } (i = F), \\ \sum_{q_{i+1}} P(q_{i+1}|q_i, \pi(q_i)) J_\pi(q_{i+1}) \\ \qquad \text{otherwise} \end{cases}$$

be the expected Lagrangian of all paths through q_i (under π). Then let

$$J^*(q_i) = \begin{cases} \epsilon_\pi(q_F) + \lambda\rho_\pi(q_F) \\ \qquad \text{if } q_i \text{ is final } (i = F), \\ \min_a \sum_{q_{i+1}} P(q_{i+1}|q_i, a) J^*(q_{i+1}) \\ \qquad \text{otherwise.} \end{cases} \qquad (2)$$

By induction, $J^*(q_i) \leq J_\pi(q_i)$ for all q_i and all π, with equality if $\pi = \pi^*$, where

$$\pi^*(q_i) = \arg\min_a \sum_{q_{i+1}} P(q_{i+1}|q_i, a) J^*(q_{i+1}) \qquad (3)$$

for all non-final states q_i. Thus the optimal policy (minimizing (1)) can be computed efficiently using (2) and (3).

B. With previous DATs

As the data unit has already arrived at the proxy, the proxy can only transmit a data packet at a transmission opportunity t_i, if the action taken at state q_i is $a_i = 1$. Hence, the policies for this case of transmission history are simply paths through the Markov decision tree in Figure 1. Therefore, there are 2^{N-i} prospective policies, or binary transmission patterns. Finding the policy minimizing the expected Lagrangian can be done either using the dynamic programming algorithm from Section V-A or slightly less efficiently with an exhaustive search through all 2^{N-i} policies. Below we explain briefly this second approach.

As explained earlier, the expected error $\epsilon(\pi_n)$ for a policy $\pi_n = [a_i, \ldots, a_{N-1}] : n = 1, \ldots, 2^{N-i}$ is simply the probability that all the transmissions from π_n as well as those from the transmission history do not result in the data unit being delivered on time to the client. Furthermore, upon receipt of an acknowledgement packet, the proxy truncates π_n and does not consider sending any further packets. Therefore, the cost for each transmission $a_j = 1 : j \in i, \ldots, N-1$ is equal to the probability that none of the previous transmissions results in an acknowledgement packet received by the proxy by t_j. Hence, the expected cost $\rho(\pi_n)$ is simply the sum of the individual costs. Then, we obtain the minimizing policy π^* as

$$\pi^* = \arg\min_{\pi_n} \epsilon(\pi_n) + \lambda\rho(\pi_n) \qquad (4)$$

Due to space constraints we deliberately omit here the expressions for $\epsilon(\pi_n)$ and $\rho(\pi_n)$, and leave their derivation as an exercise to the reader. The fact that there are two types of packets in the transmission history must be recognized in the derivation. Finally, as this case of transmission history corresponds to a sender-driven transmission, the reader is referred to [1] for computing the expected error-cost in such a scenario.

VI. Experimental Results

Here we investigate the end-to-end distortion-rate performance for streaming packetized video content using different algorithms. The video content is a two layer SNR scalable representation of the sequence *Foreman*. Using H.263+ [8] the first 130 frames of QCIF *Foreman* have been encoded into a base and enhancement layer with corresponding rates of 32 and 64 Kbps. Two streaming systems are employed in the experiments. *Sender-driven* is a system that performs R-D optimized scheduling of the packet transmissions at the media server [1]. *Proxy-driven* also performs R-D optimized scheduling of the packet transmissions, but at the proxy, using the algorithms from Sections IV and V. 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 available bit rate (Kbps) on Channel 2.

Channel 1 is specified as follows. Packets transmitted on this channel are dropped at random, with a drop rate $\epsilon_{F_1} = \epsilon_{B_1} = \epsilon_1 = 10$ %. Those packets that are not dropped receive a random delay, where for the forward and backward delay densities p_{F_1} and p_{B_1} we use identical shifted Gamma distributions with parameters (n_1, α_1) and right shift κ_1, where $n_1 = 2$ nodes, $1/\alpha_1 = 25$ ms, and $\kappa_1 = 50$ ms. Channel 2 is similarly specified with $\epsilon_{F_2} = \epsilon_{B_2} = \epsilon_2$, $n_2 = 1$ node, $1/\alpha_2 = 5$ ms, and $\kappa_2 = 5$ ms.

First, we compare the performance of the two systems as a function of the quality of Channel 2, where the loss rate ϵ_2 is increased from 1% to 15%. It can be seen from Figure 2 that *Proxy-driven* performs better over the whole range of available rates on Channel 2, for each of the loss rates under consideration. The difference in performance increases as the loss rate on Channel 2 is increased, which is expected. As the quality of Channel 2 degrades, more packets need to be retransmitted from the server in *Sender-driven* and from the proxy in *Proxy-driven*. Due to the random nature of the loss and delay in the backbone, the retransmissions from the server are more likely to be lost or received late than those from the proxy. We observed similar results for streaming another QCIF video *Mother and Daughter*. However, these are not included here due to space constraints.

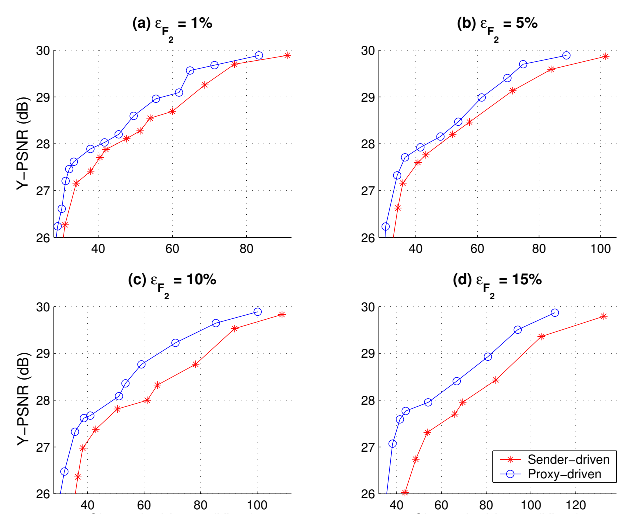

Fig. 2. R-D performance of *Sender-driven* and *Proxy-driven* for *Foreman*, ϵ_{F_2} = (a) 1%, (b) 5%, (c) 10% and (d) 15%.

Next, we examine the error-cost function of the two streaming systems for the four loss rates under consideration for Channel 2. As explained earlier, this function provides an insight into how efficiently a system trades off redundancy and error on the average for transmission of a single data unit. This ultimately affects the efficiency of the streaming system in terms of the provided trade-off between the amount of transmitted data and the resulting distortion at the client for the whole media presentation. In Figure 3 we show the error-cost functions for the two systems under each of the four loss rates. The functions are computed for the case of no history of previous transmissions and number of transmission opportunities $N = 3$. It can be seen that the error-cost function of *Proxy-driven* outperforms that of *Sender-driven* for cost values ≥ 0.9. This is the error-cost region where a transmission is actually made at the current slot t_i as the policies with $a_i = 1$ typically lie on this section of the error-cost function. Therefore, on the average *Proxy-driven* transmits less bytes to achieve the same probability of delivering a data unit to the client on time.

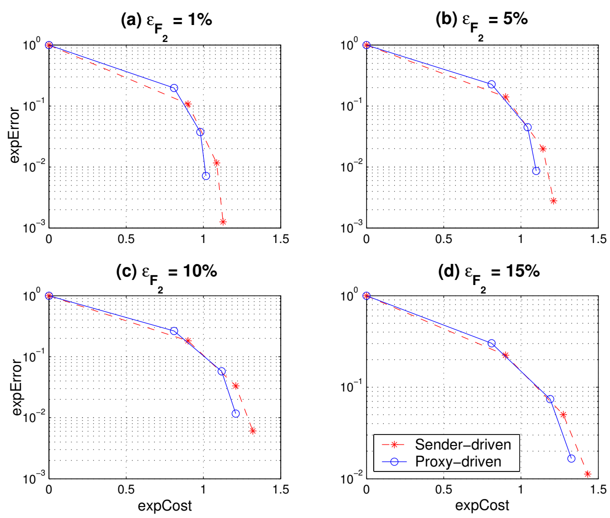

Fig. 3. Error-cost function for *Sender-driven* and *Proxy-driven*, ϵ_{F_2} = (a) 1%, (b) 5%, (c) 10% and (d) 15%.

Finally, we examine the distortion-rate performance of the two systems as a function of the quality of Channel 1. The parameters of Channel 2 are kept fixed and the loss rate is $\epsilon_2 = 5\%$. As we already examined earlier the situation where Channel 1 exhibits random packet loss and delay, here we model Channel 1 as a lossless link, characterized with a fixed transmission delay only. A delay of 5 ms and 50 ms is respectively used to obtain the results shown in Figures 4a and 4b. Note that we ignore here any additional existing delays on the side of the backbone network, such as the processing delay at the media server. The *Foreman* video is used in the experiments. As can be seen from Figure 4, the performance difference between *Proxy-driven* and *Sender-driven* becomes insignificant as Channel 1 becomes a lossless link with a considerable transmission delay. The difference in performance diminishes even further, as the transmission delay is decreased to a very small quantity.

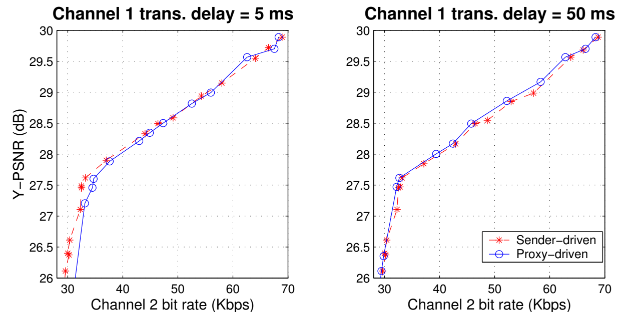

Fig. 4. R-D performance of *Sender-driven* and *Proxy-driven* for QCIF Foreman, trans. delay = (a) 5 ms, and (b) 50 ms.

VII. Conclusions

A methodology has been presented for computing rate-distortion optimized transmission policies for streaming packetized media in a hybrid receiver/sender driven scenario. This is a natural result of a larger R-D optimization framework. The computation of the optimal policies is done using a Markov decision tree with finite horizon N, associated with the transmission scenario under consideration. The R-D optimization framework is employed for streaming of video sequences and its performance is compared with a sender-driven R-D optimized system. The results demonstrate that the proposed framework performs favourably over a large range of scenarios considered for the quality of the last hop. The performance gains increase as the quality of the last hop degrades. Finally, in situations where the backbone network behaves as a lossless link with a small fixed transmission delay, the proxy-driven system does not offer any advantages over sender-driven systems.

References

[1] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *IEEE Trans. Multimedia*, 2001. Submitted.

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

[3] J. Chakareski and P. A. Chou. Application layer error correction coding for rate-distortion optimized streaming to wireless clients. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002. IEEE.

[4] A. Sehgal and P. A. Chou. Cost-distortion optimized caching of streaming media. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002. IEEE.

[5] Z. Miao and A. Ortega. Scalable proxy caching of video under storage constraints. *IEEE J. Selected Areas in Communications*, 20(7):1315–1327, September 2002. Special issue on Internet Proxy Services.

[6] G. Cheung, W.-T. Tan, and T. Yoshimura. Rate-distortion optimized application-level retransmission using streaming agent for video streaming over 3g wireless network. In *Proc. Int'l Conf. Image Processing*, volume 1, pages 529–532, Rochester, NY, USA, September 2002.

[7] J. Chakareski, P. A. Chou, and B. Aazhang. Computing rate-distortion optimized policies for streaming media to wireless clients. In *Proc. Data Compression Conference*, Snowbird, UT, April 2002. IEEE.

[8] ITU Telecom. Standardization Sector of ITU. Video coding for low bitrate communication. *Draft ITU-T Recommendation H.263 Version 2*, September 1997.