Rate-Distortion Optimized Video Streaming with Rich Acknowledgments

Jacob Chakareski* and Bernd Girod

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

ABSTRACT

We consider an unconventional procedure for communicating to the server the receipt of media packets for Internet video streaming. Instead of separately acknowledging each media packet as it arrives, we periodically send to the server a single acknowledgment packet, denoted *rich acknowledgment*, that contains information about all media packets that have arrived at the client by the time the rich acknowledgment is sent. We investigate rate-distortion optimized sender-driven streaming that employs rich acknowledgments. Performance gains of up to 1.3 dB for streaming packetized video content are observed over rate-distortion optimized sender-driven systems that employ conventional acknowledgments.

1. INTRODUCTION

We consider the problem of rate-distortion optimized video streaming from a server to a client over a lossy packet network using rich feedback from the client. Packets may be lost in the network due to congestion or erasures. In addition, packets arriving late are also considered lost. Currently, in sender-driven transmission schemes employed for streaming media the client replies with an acknowledgment packet whenever a media packet arrives. The purpose of the acknowledgment packet is to inform the server that the client has received the corresponding media packet and that the server does not need to consider retransmitting that media packet again.

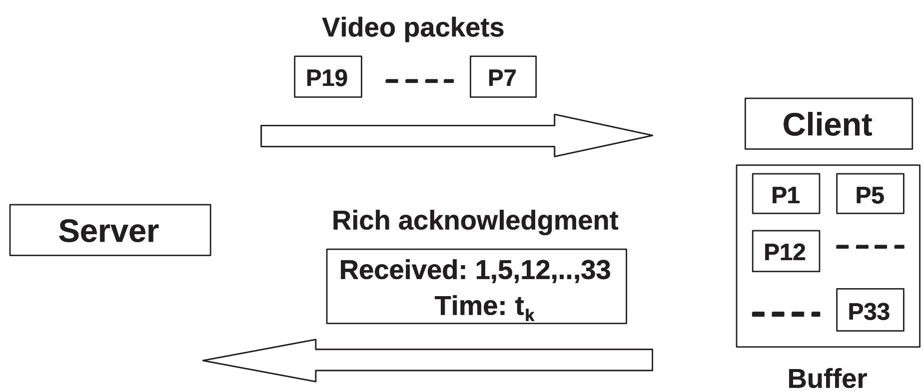

Figure 1. Streaming on demand using rich acknowledgments.

In the present work, we employ an unconventional procedure for communicating to the server the receipt of media packets. Instead of separately acknowledging each media packet as it arrives, we periodically send to the server a single acknowledgment packet, denoted henceforth *rich acknowledgment*, that contains information about all media packets that have arrived at the client by the time the rich acknowledgment is sent. This information in essence reflects the current state of the client's buffer, i.e., which packets have been received by the client thus far. The proposed scenario of streaming with rich acknowledgments is illustrated in Figure 1.

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

It should be noted that the concept of rich acknowledgments is not new and has been introduced in the networking community under the name of vector acknowledgments. However, to the best of our knowledge rich acknowledgments have not been explored yet in media streaming. The purpose of vector acknowledgments is to allow a TCP sender to perform selective retransmission of lost data packets, which is not provided by the acknowledgment scheme employed by TCP, called cumulative ack (CACK) [1]. In essence, vector acks are binary maps that describe the correctly received or missing data in the receiver's buffer and have been adopted into several proposed feedback schemes [2–4]. Another alternative for providing selective retransmission in TCP is the selective ack option (SACK) [5], which allows a receiver to communicate simultaneously the identities of several contiguous blocks of successfully received data.

The contributions of this paper can be summarized as follows. First, it introduces the concept of rich feedback, i.e., rich acknowledgments in streaming of packetized media. Second, it provides a framework for rate-distortion optimized scheduling of the packet transmissions that employs rich acknowledgments as a feedback scheme. Rate-Distortion Optimized (RaDiO) packet scheduling is one of the latest advances in media streaming. In this approach, the media server or the client is equipped with a rate-distortion optimization framework for scheduling the packet transmissions such that a constraint on the average transmission rate is met while minimizing at the same time the average end-to-end distortion. In [6–8] the authors have introduced a framework for distortion-rate optimized scheduling of the transmissions of packetized media and applied it to the scenario of sender-driven streaming. In [9] the authors have employed this framework to study the scenario of receiver-driven transmission over best-effort networks. Rate-distortion optimized streaming over lossy packet networks to wireless clients, again in a sender-driven scenario, has been studied in [10–12]. In addition, in [13, 14] a streaming system is introduced, called *RaDiO Edge*, centered around a proxy server, located at the edge of the backbone network, and equipped with a RaDiO packet scheduling procedure and a hybrid receiver/sender driven transmission. Finally, in [15] the authors present a framework for rate-distortion optimized sender-driven streaming with path diversity, while in [16] the authors consider rate-distortion optimized receiver-driven streaming with server diversity. All these works have demonstrated a substantial improvement in performance over current state of the art solutions.

Media packets are typically characterized by different deadlines, importances and interdependencies. Using this information and the framework presented in this paper, the sender is able to transmit its media packets based on the rich acknowledgments it receives 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.

We present the major ideas in our paper as follows. In Section 2, we define our abstractions of the encoding, packetization, and communication processes. In Section 3, we explain the proposed protocol or algorithm for streaming packetized media that employs rich acknowledgments as a feedback scheme. Next, in Section 4 we show how the entire media presentation can be transmitted in a rate-distortion optimized way, using as a building block an algorithm for rate-distortion optimized transmission of a single media packet. This algorithm is the subject of Section 5. In Section 6, we report our experimental results. Finally, concluding remarks are provided in Section 7.

2. SOURCE AND CHANNEL CHARACTERIZATIONS

2.1. Media Source Model

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 $t_{DTS,l}$, a set of data units $\mathcal{N}_c^{(l)}$ and an importance $\Delta d_l^{(l_1)}$. Specifically, 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. Finally, $\mathcal{N}_c^{(l)}$ 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, while $\Delta d_l^{(l_1)}$, for $l_1 \in \mathcal{N}_c^{(l)}$, is the reduction in reconstruction error (distortion) for the media presentation if data unit l is not decodable and is concealed with data unit l_1 that is received and decoded on time.

2.2. Packet Loss Probabilities

The forward and the backward channel on a network path between a server and a client are characterized as independent time-invariant packet erasure channels with random delay. Hence, they are completely specified with the probabilities of packet loss ϵ_F and ϵ_B, and the probability densities of the transmission delay p_F and p_B, respectively. This means that if the media server sends a packet on the forward channel at time t, then the packet is lost with probability ϵ_F. However, if the packet is not lost, then it arrives at the client at time t', where the forward trip time $FTT = t' - t$ is randomly drawn according to the probability density p_F. Therefore, we let $P\{FTT > \tau\} = \epsilon_F + (1-\epsilon_F)\int_\tau^\infty p_F(t)dt$ denote the probability that a packet transmitted by the server at time 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, $P\{BTT > \tau\} = \epsilon_B + (1-\epsilon_B)\int_\tau^\infty p_B(t)dt$ denotes the probability that a rich acknowledgment packet transmitted by the client at time 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.

3. MEDIA COMMUNICATION USING RICH ACKNOWLEDGMENTS

A media session starts when a client requests a presentation from the media server. Once the request packet is received, the media server starts sending packets with data units from the presentation at discrete transmission opportunities $t_i, t_{i+1}, \ldots$. Note that the transmission decisions regarding when and how often each of the data units will be sent to the client are completely determined by the server's transmission policy. The client periodically monitors the status of its buffer at every t_i and returns to the server this information via a single acknowledgment packet, i.e., a rich acknowledgment. The buffer information basically informs the server what data units have arrived at the client by the time (t_i) the rich acknowledgment was transmitted. Based on this information and the optimization framework presented in the next section, the server then dynamically decides at every transmission opportunity t_i what is at that moment the best transmission policy for every data unit in the presentation.

4. 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 [15]

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

where D_0 is the expected reconstruction error for the presentation if no data units are received. $\mathcal{A}(l_1)$ is the set of ancestors of l_1, including l_1. $\mathcal{C}(l,l_1)$ is the set of data units $j \in \mathcal{N}_c^{(l)} : j > l_1$ that are not mutual descendants, i.e., for $j, k \in \mathcal{C}(l, l_1) : j \notin \mathcal{D}(k), k \notin \mathcal{D}(j)$, where $\mathcal{D}(j)$ is the set of descendants of data unit j. "$\backslash$" denotes the operator "set difference".

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 [6]. It can be shown that the optimal individual policies at iteration n, for $n = 1, 2, \ldots$, are given by

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

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 [6], 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 $\lambda' = \lambda B_l / S_l^{(n)}$. In the next section, we show how to find π^* for the family of transmission policies Π corresponding to sender-driven streaming with rich acknowledgments.

5. COMPUTING THE OPTIMAL TRANSMISSION POLICY

For transmitting a single data unit on the forward channel, 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 server considers transmitting a packet for the data unit. The server need not transmit a packet at every transmission opportunity. The server does not transmit any packets after a rich acknowledgment is received confirming the receipt of the data unit at the client. In addition, as explained in Section 3, the client sends a rich acknowledgment packet to the server on the backward channel at every t_i notifying the server about the state of its buffer, i.e., which data units have arrived at the client by t_i. Note that at the same time this also informs the server which data units have not arrived at the client by t_i. Therefore, the information provided by a received rich acknowledgment is much richer than that provided by a received regular acknowledgment.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the server takes an action a_i, where $a_i = 1$ if the server sends a packet and $a_i = 0$ otherwise. Then, at the next transmission opportunity t_{i+1}, the server makes an observation o_i, where o_i is the set of rich acknowledgments received by the server in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{NAK(t_1), ACK(t_3)\}$ means that during the interval $(t_i, t_{i+1}]$, the rich acknowledgment sent at t_1 arrived at the server informing that the data unit has not been received by the client by t_1 ($NAK(t_1)$) and that the rich acknowledgment sent at t_3 arrived at the server informing that the data unit has been received by the client by t_3 ($ACK(t_3)$). Note that for the purposes of our algorithm it is irrelevant for the server to distinguish the transmission times of the rich acknowledgments received within $(t_i, t_{i+1}]$ confirming that the data unit has arrived at the client by their respective transmission times. As explained above, once the server receives a confirmation that the data unit has arrived at the client, it stops sending any packets afterwards, regardless of the transmission times of the rich acknowledgments that brought that confirmation. Therefore, we drop the timing notation on these rich acknowledgments and simply use ACK to denote the event that at least one rich acknowledgment has been received confirming the receipt of the data unit due to previous transmissions. In addition, we denote the event $o_i = \{NAK(t_1), ACK\}$ simply as $o_i = ACK$ since receiving any NAKs together with at least one ACK will not affect the transmission actions that the server considers afterwards. Again, as we explained earlier, the server does not transmit any packets with the data unit after an ACK has been received, regardless of any number of NAKs that have also arrived during $(t_i, t_{i+1}]$. Finally, we denote the event $o_i = \{NAK(t_1), NAK(t_3)\}$ simply as $o_i = NAK(t_3)$ since not receiving the data unit by t_3 implies that the data unit was certainly not received by t_1. In other words, we denote the events o_i, when multiple NAKs are received as an observation, using only the most recently sent NAK.

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. Therefore, a state represents uniquely this sequence of action-observation pairs. 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 2 are indicated by double circles.

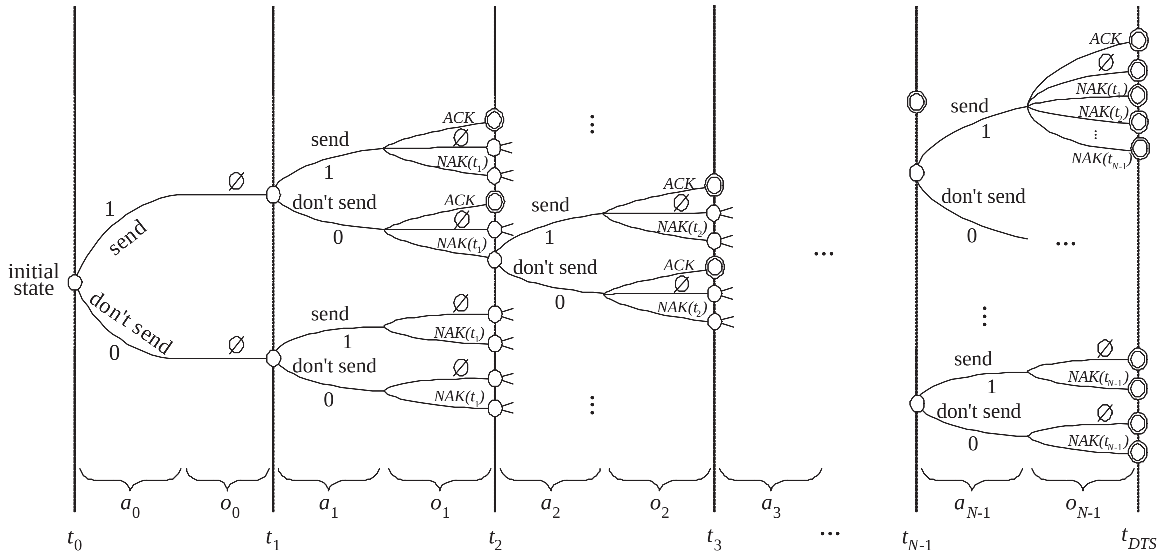

Figure 2. State space for a Markov decision process.

The action a_i taken at a non-final state q_i determines the transition probability $P(q_{i+1}|q_i, a_i)$ 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 tree with transition probabilities between states $P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i))$, 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 $\rho_\pi(q_F) = \sum_{i=0}^{F-1} a_i$, 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 packets transmitted under policy π results in successful decoding by time t_{DTS}, given q_F. For example, if q_F is the second state from the top at time t_{DTS} in Figure 2, then a packet with the data unit was sent at every transmission opportunity $t_0, t_1, \ldots, t_{N-1}$ and no rich acknowledgments were received. In that case, $\epsilon_\pi(q_F) = \prod_{i=0}^{N-1} P\{FTT > t_{DTS} - t_i\}$. Therefore, Π is a collection of all possible trees in the state space described in Figure 2.

We can now express the expected cost and error for the Markov tree induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F)\epsilon_\pi(q_F)$. As stated earlier we are interested in the policy minimizing the expected Lagrangian cost

$$J(\pi) \equiv \epsilon(\pi) + \lambda' \rho(\pi) = \sum_{q_F} P_\pi(q_F) J_\pi(q_F), \tag{3}$$

where $J_\pi(q_F) \equiv \epsilon_\pi(q_F) + \lambda' \rho_\pi(q_F)$. In a follow-up work [17], we explain in detail how the optimal transmission policy π^* is computed using a dynamic programming algorithm.

6. EXPERIMENTAL RESULTS

Here, we investigate the end-to-end distortion-rate performance for streaming packetized video content using different algorithms. The videos used in the experiments are two-layer SNR scalable representations of the image sequences *Foreman* and *Mother and Daughter*, henceforth denoted *MaD*. Using H.263+ [18] 130 frames of QCIF *Foreman* have been encoded into a base layer and an enhancement layer with corresponding rates of 32 and 64 Kbps. Similarly, 130 frames of QCIF *MaD* have been encoded into two layers with rates 32 and 69 Kbps, respectively. For both videos 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. Two RaDiO streaming systems are employed in the experiments. *Conv. ACK* is a system that performs RaDiO scheduling using conventional acknowledgments [6–8]. *Rich ACK* is the system presented in this work, which also performs RaDiO packet scheduling, but using rich acknowledgments. The Lagrange multiplier λ is fixed for the entire presentation for both systems. 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 average transmission rate (Kbps) on the forward channel. In the experiments we use $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the playback delay.

The forward and the backward channel on the network path between the server and the client are specified as follows. Packets transmitted on these channels are dropped at random, with a drop rate $\epsilon_F = \epsilon_B = \epsilon = 10$ %. Those packets that are not dropped receive a random delay, where for the forward and the backward delay densities p_F and p_B we use identical shifted Gamma distributions with parameters (n, α) and right shift κ, where $n = 2$ nodes, $1/\alpha = 25$ ms, and $\kappa = 50$ ms for a mean delay of $\kappa + n/\alpha = 100$ ms and standard deviation $\sqrt{n}/\alpha \approx 35$ ms.

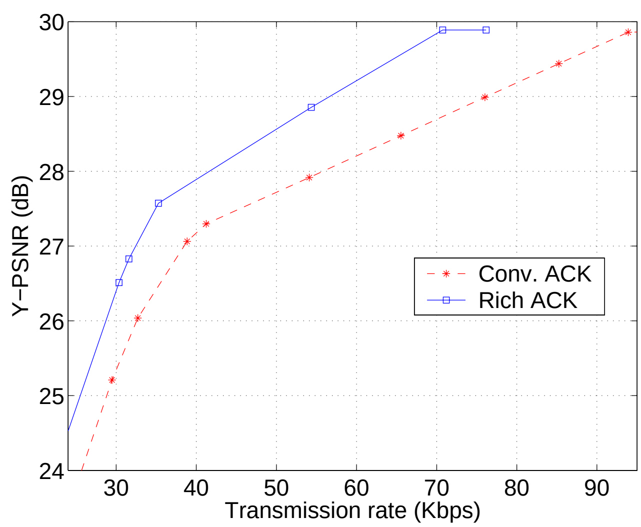

Figure 3. Rich vs. conventional acknowledgments for rate-distortion optimized streaming of QCIF *Foreman*.

It can be seen from Figure 3 that given the selected simulation parameters using rich acknowledgments can improve performance for streaming *Foreman*. *Rich ACK* performs consistently better than *Conv. ACK* over all transmission rates under consideration. The performance gains reach up to 1.3 dB for a transmission rate of 70 Kbps, or equivalently, transmission rate savings of 27 % are observed for the given PSNR of 29.9 dB. The difference in performance between the two systems is due to the fact that rich acknowledgments can make up for losses of individual acknowledgment packets in the *Conv. ACK* system. In addition, they also provide the server with a much richer knowledge of the state of the client's buffer than that provided by regular acknowledgments, as explained on the beginning of Section 5. Consequently, the server is able to exploit this information to its benefit and therefore to provide an enhanced performance over a rate-distortion optimized system that only employs regular acknowledgment packets. In essence, the transmission policies computed based on the knowledge provided by rich acknowledgments are more efficient in a rate-distortion sense than those computed based on regular acknowledgments.

We observe a similar situation for streaming *MaD* as seen from Figure 4. *Rich ACK* outperforms *Conv. ACK* over the whole range of available transmission rates. The performance gains in this case reach up to 0.9 dB for transmission rate of 70 Kbps, or equivalently, transmission rate savings of 22.2 % are observed for the given PSNR of 34.3 dB. Note, however, that the performance gains in this case are not as large as those for *Foreman*. This is due to the nature of the sequence *MaD*, which exhibits comparably less motion than *Foreman*. Therefore, the drop in quality incurred by a lost or late packet is not as significant for *MaD* as is for *Foreman* since error concealment can be performed more successfully in the case of the former.

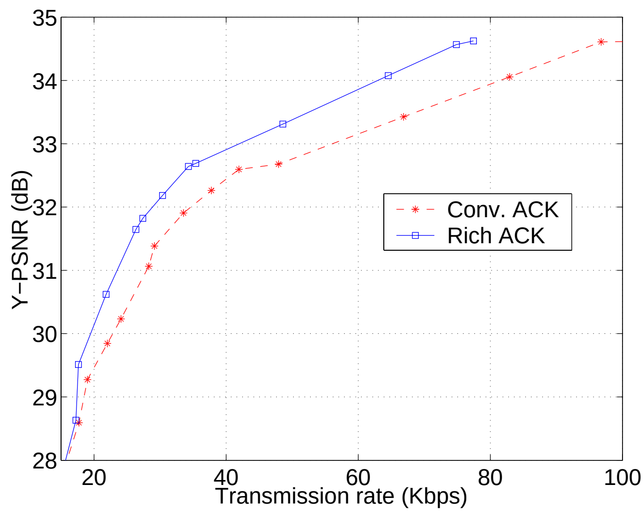

Figure 4. Rich vs. conventional acknowledgments for rate-distortion optimized streaming of QCIF *MaD*.

7. CONCLUSIONS

We have presented a system for rate-distortion optimized sender-driven streaming of packetized media with rich acknowledgments. The system consists of two major components. A feedback scheme that periodically returns to the sender a single acknowledgment packet, called rich acknowledgment, that informs the sender of the state of the receiver's buffer at the time when the rich acknowledgment was transmitted. The second component of our system is an optimization framework that enables the sender to optimize in a rate-distortion sense its transmission policies based on the knowledge provided by received rich acknowledgments and on the knowledge of the media source and the communication channels. The proposed system provides significant gains in performance over rate-distortion optimized systems that employ conventional acknowledgments as a feedback scheme. An additional advantage of streaming with rich acknowledgments that we did not explore here is that it needs a much smaller transmission bandwidth on the feedback (backward) channel. In a follow-up work [17], we explore in greater detail how the gains due to rich acknowledgments depend on the parameters that characterize the feedback channel, such as packet loss and transmission delay.

REFERENCES

1. W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Boston, MA, 1994.
2. B.T. Doshi, P.K. Johri, A.N. Netravali, and K.K. Sabnani, "Error and flow control performance of a high speed protocol," *IEEE Trans. Communications*, vol. 41, no. 5, pp. 707–720, May 1993.
3. J.C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *Proc. Conf. on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Mar. 1996, IEEE, vol. 3, pp. 1414–1424.

4. H.-S. Wilson So, Y. Xia, and J. Walrand, “A robust acknowledgement scheme for unreliable flows,” in *Proc. Conf. on Computer Communications (INFOCOM)*, New York, NY, USA, June 2002, IEEE, vol. 3, pp. 1500–1509.
5. M. Mathis, J. Madhavi, S. Floyd, and A. Romanow, “TCP selective acknowledgment options,” Tech. Rep. RFC 2018, IETF, http://www.ietf.org/rfc/rfc2018.txt, Oct. 1996, Proposed standard.
6. P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Trans. Multimedia*, 2001, submitted.
7. P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” Tech. Rep. MSR-TR-2001-35, Microsoft Research, Redmond, WA, Feb. 2001.
8. P. A. Chou and Z. Miao, “Rate-distortion optimized sender-driven streaming over best-effort networks,” in *Proc. Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 587–592, IEEE.
9. 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.
10. 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, Apr. 2002, IEEE Computer Society, pp. 53–62.
11. 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, vol. 3, pp. 2513–2516.
12. J. Chakareski and P.A. Chou, “Application layer error correction coding for rate-distortion optimized streaming to wireless clients,” Tech. Rep. MSR-TR-2002-81, Microsoft Research, Redmond, WA, Aug. 2002, Publicly available at ftp://ftp.research.microsoft.com/pub/tr/TR-2002-81.ps.
13. J. Chakareski, P.A. Chou, and B. Girod, “Rate-distortion optimized streaming from the edge of the network,” in *Proc. Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, Dec. 2002, IEEE, pp. 49–52.
14. J. Chakareski, P.A. Chou, and B. Girod, “Computing rate-distortion optimized policies for hybrid receiver/sender driven streaming of multimedia,” in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2002, IEEE, vol. 2, pp. 1310–1314.
15. 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, Mar. 2003, IEEE Computer Society, pp. 203–212.
16. J. Chakareski and B. Girod, “Server diversity in rate-distortion optimized streaming of multimedia,” in *Proc. Int’l Conf. Image Processing*, Barcelona, Spain, Sept. 2003, IEEE.
17. J. Chakareski and B. Girod, “Computing rate-distortion optimized policies for media streaming with rich acknowledgments,” in *Proc. Data Compression Conference*, Snowbird, UT, Mar. 2004, IEEE Computer Society, submitted.
18. Telecom. Standardization Sector of ITU, “Video coding for low bitrate communication,” *ITU-T Recommendation H.263 Version 2*, Feb. 1998.