RATE-DISTORTION OPTIMIZED VIDEO STREAMING OVER INTERNET PACKET TRACES

*Jacob Chakareski** *and Bernd Girod*

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

ABSTRACT

In this paper, we study the performance of rate-distortion optimized video streaming over traces of packet delays and packet losses collected in the Internet. The study provides us with an understanding how important rate-distortion optimization may be for Internet streaming today and when it actually pays off to perform it. We propose a simple technique for channel estimation that can be incorporated within rate-distortion optimized streaming to remove the assumption of known channel statistics. In the experiments, performance of rate-distortion optimized streaming is compared to that of simpler transmission techniques such as ARQ.

1. INTRODUCTION

One of the most important recent advances in streaming technology is the emergence of Rate-Distortion Optimized (RaDiO) packet scheduling techniques [1–3] that take into account the different importance of individual packets and knowledge about the channel in a Lagrangian rate-distortion cost function $J = D + \lambda R$. This is in contrast to the early streaming systems which simply transmitted media packets without regard for the importance of individual packets or the prevalent channel conditions, except, maybe, the available maximum transmission rate. Using a RaDiO technique packet transmission schedules are computed such that a constraint on the average transmission rate is met while minimizing at the same time the average end-to-end distortion. The performance improvements of the RaDiO techniques reported to date relative to non-Lagrangian heuristics are very encouraging.

A framework for RaDiO sender-driven streaming of packetized media has been proposed in [3]. The flexibility of the framework has allowed its application to a number of streaming scenarios such as [4–6]. Important limitations of the original framework for video streaming were overcome by an advanced framework for RaDiO video streaming proposed in [7]. Using this framework, various streaming scenarios such as streaming over multiple network paths [7], distributed streaming from multiple servers [8], streaming with an intermediate network proxy [9], and most recently streaming with rich acknowledgements [10] and rich requests [11] have been addressed.

In all the studies of RaDiO streaming reported to date, except for the initial work in [12], performance is evaluated with respect to statistical models of Internet behaviour in terms of packet delay and packet loss. In this paper, we undertake a comprehensive study of RaDiO streaming over traces of packet delays and packet losses collected in the Internet. The study provides us with an understanding how important rate-distortion optimization may be for Internet streaming today and when it is actually beneficial to perform it. Furthermore, we have designed a technique for online channel estimation that is incorporated within the framework for RaDiO streaming employed in our study.

The rest of the paper is structured as follows. In the next section, we provide a brief background on the framework for RaDiO streaming used in the paper. We examine the performance of the RaDiO framework for streaming over network traces in Section 4. In the experiments, we compare the performance of RaDiO streaming with that of simpler transmission techniques such as ARQ. Finally, we end the paper with concluding remarks provided in Section 5.

2. RATE-DISTORTION OPTIMIZATION

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 depending on the transmission scenario.

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 refer the reader to [7] for expressions for the expected distortion and rate.

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

$$\pi_l^{(n)} = \arg\min_{\pi_l} S_l^{(n)} \epsilon(\pi_l) + \lambda B_l \rho(\pi_l), \tag{1}$$

where $S_l^{(n)}$ 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. Furthermore, $\epsilon(\pi_l)$ is the probability that data unit l does not arrive at the receiver on time (under policy π_l), called the *expected error*, and $\rho(\pi_l)$ is the expected number of transmitted bytes per source byte (under policy π_l), called the *expected cost*.

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

The minimization (1) 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 $\lambda' = \lambda B_l / S_l^{(n)}$. We refer the reader to [7] for more details on the optimization framework.

3. CHANNEL ESTIMATION

The optimization framework still employs statistical models of the forward and the round-trip communication channels on the network path between the sender and the receiver in order to perform packet scheduling as described in [7]. However, since Internet behaviour in this paper is reproduced based on collected channel traces, we assume that a priori the statistics of the packet delay and loss experienced in each direction are not known. Therefore, a RaDiO system based on the framework estimates online the parameters of the statistical models. The estimation procedure works as follows.

At each transmission opportunity, at least one packet is sent to the receiver. If there are no outgoing data packets at present, then a small probe packet, comprising only a packet header and no payload, is sent instead. The receiver responds to every incoming packet on the forward channel by sending immediately an acknowledgement packet on the backward channel. The returning acknowledgement packet contains the time of the arrival of the incoming packet on the forward channel as well as its sequence number. The sender then collects the information provided by returning acknowledgement packets in terms of packet delay experienced in the forward and round-trip directions. This information is used by the sender to estimate the parameters of the statistical models characterizing the delay in each direction. Furthermore, missing sequence numbers of packets are used to estimate the packet loss rate in each direction.

The framework assumes that the packet delays are statistically characterized according to a shifted Gamma distribution with parameters (n, α) and right shift κ [13]. For the first parameter, that signifies the number of routers or queues on the network path between the sender and the receiver, we can safely assume that $n_F = n_R = 1$. That is because in the Internet the statistics of the packet delay and packet loss experienced on a network path are typically governed by the behaviour of the most congested queue along the path. The other two parameters of the shifted Gamma distribution are estimated as follows. Let FTT and RTT be respectively the measured forward-trip and round-trip times associated with the last received acknowledgement packet. Then, the running estimates of the right shift and the mean, $\tilde{\kappa}_F$ and $1/\tilde{\alpha}_F$, in the case of the forward channel are updated as

$$\begin{aligned} \tilde{\kappa}_F &= \min(\tilde{\kappa}_F, FTT), \\ 1/\tilde{\alpha}_F &= (1-\tau)(1/\tilde{\alpha}_F) + \tau FTT, \end{aligned} \tag{2}$$

where the factor $\tau \in [0, 1]$ helps to filter-out measurement noise present in FTT. The same expressions from above applied to the round-trip time measurement are employed to update $\tilde{\kappa}_R$ and $1/\tilde{\alpha}_R$ for the round-trip channel.

4. EXPERIMENTAL RESULTS

In this section, we examine the end-to-end distortion-rate performance of the optimization framework from Section 2 for streaming packetized video content. The video content is the standard CIF test video sequence Foreman that has been compressed using the 8.0 version of the state-of-the-art video codec H.264 [14] at a frame rate of 30 fps and at three different bit-rates ranging around 320, 160 and 80 Kbps, respectively. For compression efficiency, in an encoding only the first frame of the sequence is intra-encoded (I-frame), while the rest of the frames are encoded predictively as P- and B-frames. There are three B-frames between consecutive P-frames in an encoding.

In the streaming experiments, all three encodings of Foreman are employed. Switching between different encodings during streaming depending on the available network bandwidth is performed using SP/SI frames. In particular, in an encoding every 16-th frame is encoded as a primary SP frame which enables stream switching roughly every 1/2 second. Then, when switching between two encodings needs to be made, a secondary SP frame or an SI frame is sent to the receiver. For more details on stream switching via SP/SI frames, the reader is referred to [15].

Two sender-driven streaming systems are examined in the experiments in this section. *ARQ* is a system that does not take into consideration the distortion information of individual packets when making transmission decisions. It employs time-out driven retransmission for packets that have not been acknowledged by the receiver. Specifically, at each transmission opportunity data units are (re)transmitted by *ARQ* in a GOP order based on their delivery deadline. A data unit is considered for retransmission only if its arrival at the receiver has not been acknowledged within RTO (retransmission time-out interval) time after its last transmission, where RTO is set to $\tilde{\mu}_R + 3\,\tilde{\sigma}_R$. The procedure described in Section 3 is employed by *ARQ* to compute $\tilde{\mu}_R$ and $\tilde{\sigma}_R$. The second streaming system under comparison here is denoted *RaDiO* as it employs the optimization framework from Section 2 to perform scheduling decisions. The Lagrange multiplier λ is fixed for the entire presentation for the *RaDiO* system.

Representatives of two types of network traces are employed in the experiments: (1) network traces that do not exhibit packet losses, and (2) network traces that exhibit packet losses uniformly over the whole duration of the trace collection procedure. We continue this section by examining first experimental results involving a network trace representing the first category.

4.1. Traces without Packet Losses

The Internet trace that we utilize here has been collected during a course of 5 hours between a machine at Stanford University and a residence in Mountain View, California which connected to the Internet via a cable modem. The collection process started around 5:00pm on a working day. There were 14 Internet hops between the two machines. The downlink bandwidth to the cable modem from the Stanford host, as estimated by an ftp file transfer, was in excess of 1.5 Mbits per second. The uplink bandwidth was approximately 230 Kbits per second. The trace collection process sent 20 packets per second in each direction (downlink and uplink) of size 32 bytes (IP header, UDP header, and sequence number), for a total of 6.4 Kbits per sec. The probe packets thus utilized less than 2.8% of the available upstream bandwidth, and 0.42% of the available downstream bandwidth. Therefore, it is reasonable to assume that they did not affect the loss and delays experienced by themselves in either direction, i.e., they did not have a self-congestion effect. For more details on the methodology employed to collect the trace and other related issues, the reader is referred to [16].

In Figure 1 we examine the Y-PSNR performance in dB of the two systems under comparison for streaming the *Foreman* sequence as a function of the transmission rate in Kbps on the forward (downlink) channel. For both systems the time interval between transmission opportunities T is set to 50 ms, which is the time spacing at which network measurements were collected via packet probes in the trace collection procedure, as described earlier. Furthermore, the play-out delay Δ for the video clip is 600 ms. It can be seen from the figure that *ARQ* and *RaDiO* provide

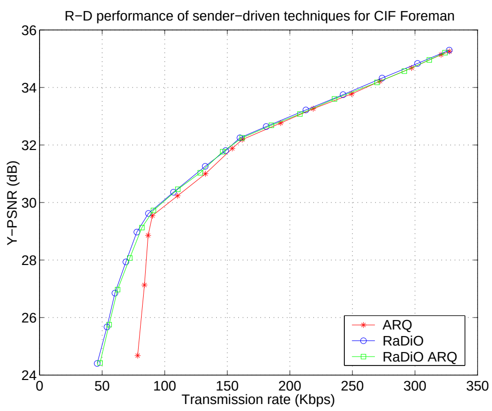

Fig. 1. R-D performance for streaming *Foreman* over a network trace with no packet losses.

almost the same performance in the two upper thirds of the range of transmission rates under consideration. Specifically, for transmission rates greater than 90 Kbps the two systems perform quite alike with *RaDiO* outperforming *ARQ* with only an insignificant margin. That is because of two reasons. First, no retransmissions of packets occur in either system, since packet losses are very rare and packet latencies are low. Retransmissions would certainly have a stronger impact on the performance of *ARQ*, as this system does not distinguish between packets based on their distortion importance so every packet unacknowledged within the RTO is retransmitted equally likely.

The second reason is the fact that over the range of transmission rates where the performances of the two systems are quite similar, no packet dropping, i.e., omission[1], at the sender occurs in *ARQ* in order to account for bandwidth variability. Specifically, in this range of rates *ARQ* employs stream switching via SP/SI frames, rather than packet dropping at the sender, in order to reduce its transmission rate. As in the case of retransmissions, packet dropping would certainly have a stronger effect on the performance of *ARQ* due to the same reason explained earlier, i.e., the fact that *ARQ* is oblivious to packet distortion information. This is in fact obvious from the performances of *ARQ* and *RaDiO* that correspond to the lower third of transmission rates under consideration in Figure 1. For transmission rates below 90 Kbps no stream switching can occur, so *ARQ* is forced to work only with a single encoding and therefore it has to employ packet dropping in order to reduce its transmission rate further. It can be seen from Figure 1 that in this range of rates *RaDiO* provides a much more graceful degradation in performance as the transmission rate is reduced toward its low end. For example, for transmission rate of 77 Kbps there is a performance difference between the two systems that reaches almost 4.5 dB, which is quite significant.

[1]Also known as source pruning.

This apparent deficiency of *ARQ* can be overcome by enhancing this system as follows. Instead of allowing *ARQ* to stream from a full source encoding, we preprocess the encoding ahead of time, i.e., before streaming. Specifically, by dropping certain packets from the encoding we can reduce the source rate of the encoding in a rate-distortion optimal way according to the available transmission rate. For that purpose we could use an optimal pruning algorithm such as the *RaDiO* system itself. The original *ARQ* system is then used to stream from the processed encoding. We denote such an enhanced (rate-distortion optimized) ARQ system as *RaDiO ARQ*. Its performance is shown in Figure 1 along with those of the original *ARQ* and *RaDiO*. It can be seen from the figure that now the performances of the ARQ system with preprocessing and of *RaDiO* are aligned over the whole range of transmission rates under consideration.

The last results presented in this section answer the question how the performances of the two systems are affected by late loss. In particular, a late loss of a packet occurs when the packet arrives at the receiver after its delivery deadline. We noticed that the packet delays experienced on the downlink during the last 1/5 of the trace duration exhibit larger values on the average relative to the rest of the trace. Moreover, there is a larger amount of variation of packet delay values for this segment of the trace. This increased activity on the downlink is because the last 1/5 of the trace roughly corresponds to the period of the day when people are back from work and use their Internet connections from home in the evening to surf the web and download (or stream) data. Therefore, we repeated our streaming experiments from before but only over this last section of the trace. To make the late loss impact the performance, we employed two smaller values for the play-out delay Δ in the experiments.[2] Specifically, we recorded the performances of *ARQ* and *RaDiO* for Δ = 400 ms and for Δ = 200 ms. These results are shown in the following figure.

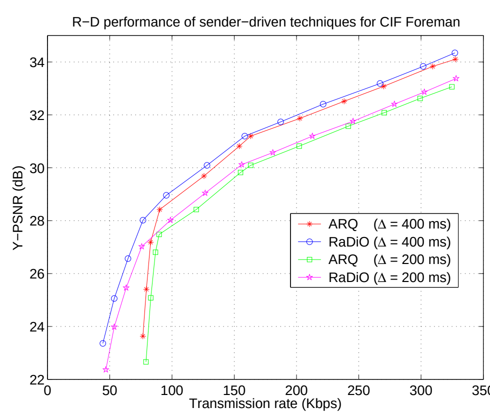

Fig. 2. R-D performance for streaming *Foreman* over a network trace without packet losses and Δ = 200 ms and 400 ms.

It can be seen from Figure 2 that late loss has equal impact on the performance of both *ARQ* and *RaDiO*. This is expected since late loss does not incur retransmissions of packets in this scenario. Therefore, the packet distortion information that *RaDiO* can exploit to its advantage is not helpful here. For example, it can be seen from the figure that for the case of Δ = 400 ms, the perfor-

[2]Even with the increased activity during this period, packet delays on the downlink quite rarely exceed 500 ms.

mances of *ARQ* and *RaDiO* have equally degraded to below 34 dB at a transmission rate of 300 Kbps, relative to the corresponding performances shown in Figure 1. When the play-out delay is additionally reduced to 200 ms, the performances of the two streaming systems degrade uniformly even further for roughly 1 dB. In the next section, we show how packet losses can have a different impact on the performances of *ARQ* and *RaDiO*.

4.2. Traces with Packet Losses

The Internet trace that we utilize here has been collected during a course of a few hours between a machine at Stanford University and another computer at Erlangen University in Germany. The path Stanford→Erlangen is the downlink channel, while the return path Erlangen→Stanford serves as the uplink channel. The trace collection process sent a probe packet every 20 ms in each direction (downlink and uplink) of size 32 bytes (IP header, UDP header, and sequence number). The trace has been provided to us courtesy of Y. Liang [17]. The recorded delay in each direction on this trace exhibits very little variation. In particular, most of its values range in the neighbourhood of 320 ms for the forward (downlink) channel and in the neighbourhood of 70 ms for the backward (uplink) channel. Furthermore, on the downlink packet losses are observed at a rate of 9.59%. The packet loss rate appears to be time-invariant. On the other hand, no lost packets are recorded on the uplink.

In Figure 3 we examine the Y-PSNR performance of *ARQ* and *RaDiO* for streaming *Foreman* over this network trace. The time interval T between successive transmission opportunities is set to 40 ms for both streaming systems. Furthermore, the play-out delay Δ for the video clip is 600 ms. It can be seen from Figure 3 that now there is a significant difference in performance between the two systems over the whole range of transmission rates under consideration. For example, in the two upper thirds of the transmission rate range, *RaDiO* outperforms *ARQ* with a margin that ranges between 1.2 - 1.7 dB. In essence, the performance gains of *RaDiO* in this case are due to the fact that *RaDiO*, contrary to *ARQ*, transmits more important packets earlier and frequently on multiple occasions, without waiting (long enough) first for a returning acknowledgement packet that may potentially arrive due to previous transmissions. This increases the robustness of the more important packets to packet losses during transmission and therefore ensures their timely delivery to the client.

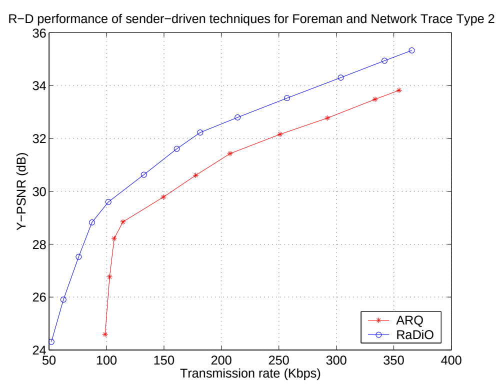

Fig. 3. R-D performance for streaming *Foreman* over a network trace with uniformly distributed packet losses.

5. CONCLUSIONS

We have examined the performance of a framework for rate-distortion optimized packet scheduling in the context of streaming over Internet traces featuring collected packet delays and losses experienced over network paths. We use a simple channel estimation technique that is incorporated within the optimization framework thereby relieving the framework from the assumption of known channel statistics. It has been shown that depending on the quality of the network path in terms of packet loss and delay, rate-distortion optimized packet scheduling may or may not provide performance gains over simpler transmission techniques, such as time-out driven (re)transmission (ARQ), where no distortion information is taken into consideration when making transmission decisions. In addition, it was shown how ARQ can be enhanced for bandwidth adaptation via the optimization framework.

6. REFERENCES

[1] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," Tech. Rep. UCB/CSD-98-1024, University of California, Computer Science Division, Berkeley, CA, Nov. 1998.

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

[3] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Tech. Rep. MSR-TR-2001-35, Feb. 2001.

[4] 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.

[5] 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.

[6] M. Kalman, E. Steinbach, and B. Girod, "R-d optimized media streaming enhanced with adaptive media playout," in *Proc. Int'l Conf. Multimedia and Exhibition*, Lausanne, Switzerland, Aug. 2002, IEEE, vol. 1, pp. 869–872.

[7] 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.

[8] 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, vol. 3, pp. 645–648.

[9] J. Chakareski, P.A. Chou, and B. Girod, "RaDiO Edge: Rate-distortion optimized proxy-driven streaming from the network edge," *IEEE/ACM Trans. Networking*, Nov. 2003, Submitted.

[10] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming with rich acknowledgments," in *Proc. Conf. on Visual Communications and Image Processing*, San Jose, CA, Jan. 2004, SPIE, vol. 5308, pp. 661–668.

[11] J. Chakareski and B. Girod, "Rate-distortion optimized media streaming with rich requests," in *Proc. Int'l Packet Video Workshop*, Irvine, CA, USA, Dec. 2004.

[12] M. Kalman and B. Girod, "Rate-distortion optimized video streaming using conditional packet delay distributions," in *Proc. Workshop on Multimedia Signal Processing*, Siena, Italy, Sept. 2004.

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

[14] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.264*, Mar. 2003.

[15] M. Karczewisz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, July 2003, special issue on the H.264/AVC video coding standard.

[16] M. Kalman and B. Girod, "Modeling the delays of successively-transmitted Internet packets," in *Proc. Int'l Conf. Multimedia and Exhibition*, Taipei, Taiwan, June 2004.

[17] Y. J. Liang, E.G. Steinbach, and B. Girod, "Real-time voice communication over the Internet using packet path diversity," in *Proc. Int'l Conf. Multimedia*, Ottawa, Canada, Oct. 2001, ACM, pp. 431–440.