Techniques for Improved Rate-Distortion Optimized Video Streaming

Mark Kalman and Bernd Girod

Information Systems Laboratory
Stanford University

We present techniques recently developed by our research group at Stanford University to improve the performance of existing algorithms for rate-distortion optimized video streaming. In rate-distortion optimized streaming, media packet transmissions are controlled in a way that gives an optimal tradeoff between the expected bit-rate that will be required for the transmissions and the expected media reconstruction distortion that will result.
In one of the techniques that we present here, performance is improved by incorporating richer, more reliable feedback information.
In another, multiple arrival deadlines are considered for media packets.
A third technique utilizes an improved statistical model characterizing packet delay and loss. Another seeks to minimize congestion when media is streamed over abottleneck link.

1. INTRODUCTION

Technologies like local area wireless networks, home networks, and cellular phone systems are continually increasing the ability of consumers to cost-effectively communicate at high data rates from any location. As these technologies become more and more ubiquitous, media streaming, which has enjoyed popularity and commercial success on the Internet to date, will find increasing opportunities and applications for use. Emerging streaming applications include, for example, high-quality video distribution in the home over wireless local area networks and real-time video communication over cellular networks. The variety of access technologies and the range of applications for streaming media continue to place challenges on techniques for scalable media compression and for reliable, low-latency data transport.

For streaming, it is sensible to encode and packetize video and audio in a scalable way so that as few packets as possible contain as much of the salient information as possible.
As a result, the individual packets are of greatly different importance. Some packets can be omitted without too much degradation, while the loss of others might be disastrous. For best-effort networks with varying throughput, delay, and packet loss rate, the challenge then is to dynamically select the best packets to transmit and to dynamically allocate redundancy

to protect against packet losses. A seminal work in this area, [1], proposed a mathematical framework for controlling media packet transmissions and retransmissions in a way that gives an optimal tradeoff between the expected bit-rate required by the transmissions and the expected media reconstruction distortion that will result. The scheme continually examines the sizes of the media packets, the amount of distortion that will result depending on which packets arrive by the time they are needed, and the ongoing packet delay and loss performance of the link used to deliver media packets. Using thisinformation the scheme dynamically selects an optimal subset of packets for transmission or retransmission.

An indication of the importance of the framework presented in [1] (which we refer to as the RaDiO framework for Rate-Distortion Optimized streaming) is the number of articles that have proposed novel applications for the framework or have proposed techniques to enhance the algorithm. Examples include [2] which suggests that the RaDiO framework may be used as an alternative to Diff serv-enabled IP networks, [3] which presents branch-and-bound techniques for reducing the computational complexity, and [4, 5] which extend RaDiO to the case when the video stream is available at multiple servers.

In this article we present techniques that our research group at Stanford University has recently developed to improve the performance of the RaDiO framework for video streaming.

In one technique, performance is improved by providing the algorithm with richer, more reliable feedback information. In another, the algorithm is extended to consider multiple arrival deadlines for media packets. A third technique significantly modifies the RaDiO framework in order to utilize an improved statistical model characterizing packet delay and loss. Another seeks to minimize congestion when media is streamed over a bottleneck link.

We have organized the article as follows. In Sec. 2 we reviewthe RaDiO framework proposed in [1]. In Sec. 3 we discuss the scheme for utilizing richer, more reliable feedback. In Sec. 4 we discuss the extension that allows the consideration of multiple arrival deadlines for packets. In Sec. 5 we present the technique that benefits from a more accurate statistical model characterizing packet loss and delay. In Sec. 6, finally, we discuss the alternative to RaDiO that minimizes congestion over a bottleneck link.

2. R-D OPTIMIZED STREAMING

We briefly review the framework for rate-distortion optimized (RaDiO) streaming in [1] which serves as the base-line for the extensions and variations described in later sections.

2.1. The RaDiO Framework

The framework assumes that a compressed media representation has been assembled into a collection of packetized data units. Each data unit l has a size in bytes B_l and a time deadline by which it must arrive in order to be useful for decoding. In addition, each data unit is associated with a *distortion reduction* ΔD_l, a value for the expected amount of distortion that will be removed from the decoded video if the unit is decodable when needed. Each data unit is also associated with a set of decoding interdependencies with other data units. In the RaDiO framework, interdependencies are expressed in a single directed acyclic graph. An example dependency graph for an SNR-scalable encoding with Intra (I), Predicted (P) and Bidirectionally predicted (B) frames is shown in Fig. 1. In this example, each square represents a data unit and the arrows indicate the decoding order.

The RaDiO framework can be used to choose an optimal set of packetized data units to transmit at successive discrete-in-time transmission opportunities. Because of decoding dependencies among data units, the importance of transmitting a packet at a given transmission opportunity often depends on which packets will be transmitted in the near future. The scheduler therefore makes transmission decisions based on an entire optimized plan governing all the transmissions that are anticipated during a finite time horizon into the future.

The plan governing packet transmissions that will occur during

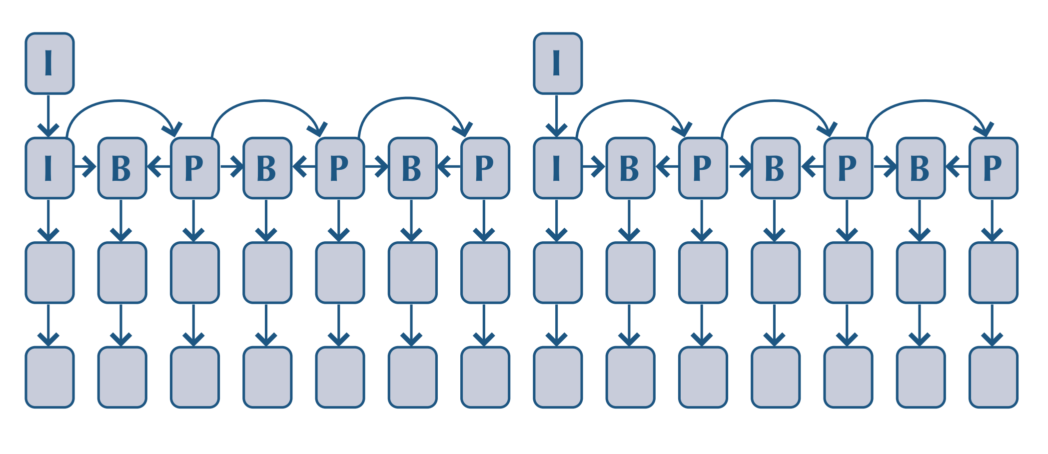

FIGURE 1:
A DIRECTED ACYCLIC GRAPH EXPRESSING DECODING DEPENDENCIES FOR AN SNR-SCALABLE ENCODING. SQUARES REPRESENT DATA UNITS AND ARROWS INDICATE DECODING ORDER.

a time horizon of discrete transmission opportunities is called a *transmission policy*, π. Assuming a time horizon of length-N, π can be represented as a collection of length-N binary vectors π_l, with one such vector for each packetized data unit l under consideration for transmission. In this representation, the N binary elements of a policy vector π_l indicate whether, under the policy, the data unit l will be transmitted or not at each of the next N transmission opportunities, unless an acknowledgement sent by the receiver arrives to indicate that the packet has been successfully received and should no longer be transmitted. A transmission policy is associated with an error probability, $\in(\pi_l)$, where an error is defined as the event that a packet does not arrive at the receiver by the time it is needed. The policy is also associated an expected number of times that the packet is transmitted under the policy, $\rho(\pi_l)$.
The goal of the framework is to find a transmission policy π that gives an optimal tradeo between expected transmission rate and expected reconstruction distortion. At any transmission opportunity the optimal π is the one that minimizes the Lagrangian cost function

$$J(\pi) = D(\pi) + \lambda R(\pi)\,, \tag{1}$$

where, for a given transmission policy, $D(\pi)$ is the expected reconstruction distortion and $R(\pi)$ is the expected transmission rate. λ controls the trade-off between the expected rate, given by

$$R(\pi) = \sum_l \rho(\pi_l) B_l \tag{2}$$

and the expected distortion given by

$$D(\pi) = D_0 - \sum_l \Delta D_l \prod_{l' \le l} (1 - \varepsilon(\pi_{l'})). \tag{3}$$

In (3) D_0 is the reproduction distortion of the group of media frames if no packets arrive, ΔD_l is the expected amount of distortion that is removed if packet l is decodable by its deadline, and the product term $\prod_{l' \le l}(1 - \in(\pi'_l))$ is the probability that packet l is decodable. $l' \le l$ refers to the set of packets which must be present to decode packet l.

Delays and losses experienced by packets transmitted over the network are assumed to be random and independent from transmission to transmission. Packet loss is modeled as Bernoulli with some probability, and packets not lost are assumed to be delayed according to a shifted-Γ distribution. Expressions for $\in(\pi_l)$ and $\rho(\pi_l)$ are given in terms of the cumulative distribution functions for the Γ-distributed delays, the transmission policies and transmission histories, and the data units' arrival deadlines.

The scheduler re-optimizes the entire policy π at each transmission opportunity to take into account information learned since the previous transmission opportunity, and then takes the transmission actions specified by the optimal π for the current time. As a method of actually finding the optimal transmission policy, exhaustive search is not generally tractable as noted in [6]. Avoiding an exhaustive search of the entire space of transmission policies is a major contribution of [1]. In [1], the authors introduce an iterative descent algorithm that simplifies the search for an optimal π. The descent algorithm iteratively minimizes (1) with respect to the transmission policy π_l of one data unit while the transmission policies of other data units are held fixed. Data units' policies are optimized in round-robin order until the Lagrangian cost converges to at least a local minimum.

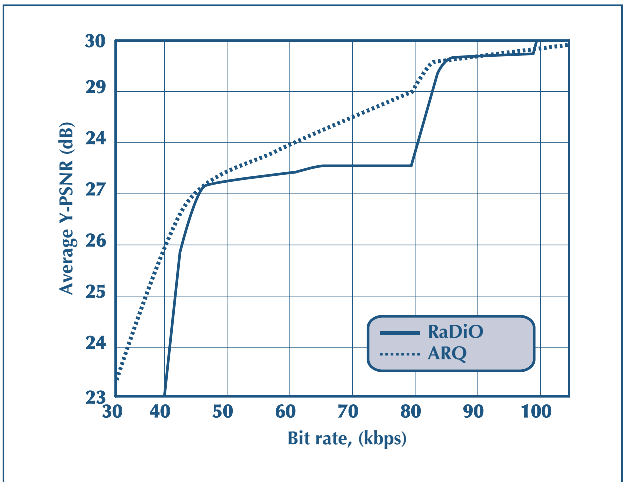

Figure 2: PSNR vs. transmitted bit-rate for a streaming system that uses heuristic deadline-constrained prioritized ARQ and for a streaming system that uses RaDiO transmission scheduling. The results are for an H.263+ [10], SNR scalable encoding of Foreman.

Fig. 2 demonstrates improved streaming performance achieved with the RaDiO scheme. Luminance PSNR versus transmitted bit-rate is plotted for streaming simulations using an H.263+ two-layer SNR scalable encoding of the Foreman sequence. The frame rate is 10 fps, a GOP consists of one I-frame followed by 9 P-frames. The encoded source rate is 32 kbps for the base layer alone, and 69 kbps when the enhancement layer is added. The results are for a simulated channel in which packet losses occur independently with loss rate 20%, and packet delays are drawn as independent, shifted- Γ random variables with mean delay 50 ms and standard deviation 25 ms. The traces in the figure plot PSNR verses transmitted bit-rate for a heuristic, prioritized ARQ system and for the R-D optimized system. In the ARQ system, the client requests retransmissions for packets that do not arrive by a time interval after they are expected, and the server transmits these requested packets with priority as long as the requested packet may still reach the client in time for playout. When the capacity of the channel falls below the source rate for the enhanced stream, the ARQ system sends only the base layer packets. Both the ARQ and the R-D optimized system use an initial pre-roll delay of 400 ms.

In Fig. 2 we see that by continuously optimizing its packet transmission choices, the optimized system makes use of the SNR and temporal scalability of the source encoding to finely tune the source rate to the available channel capacity, yielding substantial gains.

The framework can also be used for streaming over heterogenous networks, where, for instance, the last hop to the client is wireless [7]. In addition, the authors of the RaDiO framework have presented simplified, low-complexity methods to compute approximately optimized policies. Furthermore, as shown in [8], the framework appears to be robust against simplifications to the algorithm and approximations to ΔD_l, the information characterizing the value of individual packets with respect to reconstruction distortion.

2.2 Receiver Driven Streaming

It is often desirable to minimize the computational resources that a media server must provide in order to support a stream. In applications like video-on-demand, for instance, added computation reduces the number of streams that a server can simultaneously support. As shown in [8], rate-distortion optimized streaming can be performed with the algorithm running at the client so that very little computation is required at the server.

In the *receiver-driven* paradigm, the receiver is given information about the sizes, distortion reduction values and interdependencies

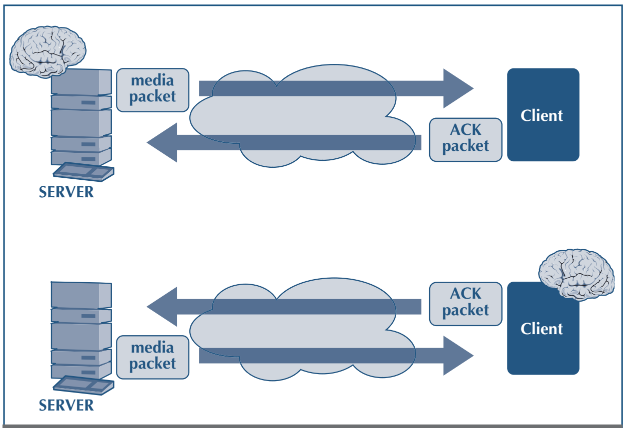

Figure 3: In sender-driven streaming (a), the server computes an optimal sequence of media packet transmissions and the client acknowledges packets upon receipt. In receiver-driven streaming (b), the complexity is shifted to the client. The client computes an R-D optimized sequence of requests to send to the server and the server only needs to respond to the client's requests.

of the data units available at the server. The receiver uses this information to compute a sequence of request packets to send to the server which specify the sequence of media packets that the server should transmit. As shown in [8], it is straightforward to adapt the algorithm discussed in Sec. 2.1 to compute a sequence of requests that yield an optimal tradeo between the expected transmission rate of the media packets that the server will send and the expected reconstruction distortion that will result. Fig. 3 summarizes the differences between sender-driven and receiver-driven streaming.

3. RICH ACKNOWLEDGEMENTS

In a recently-proposed extension to the RaDiO framework, streaming performance is improved through the use of *rich acknowledgements* [9]. In sender-driven RaDiO streaming using conventional acknowledgements, when a client receives a media packet, the client sends an acknowledgement packet (ACK) to the server. If the ACK packet is lost, the server may make non-optimal packet trasmission decisions. The server may unnecessarily retransmit the packet, for example.

With rich acknowledgements, the server does not separately acknowledge each data unit it receives. Instead, it periodically transmits a packet that acknowledges all packets that have arrived by the time that the feedback packet is sent and negatively acknowledges (NACK) packets that have not arrived. The new feedback mechanism results in changes to the RaDiO framework described in Sec. 2.

As shown in [1], a transmission policy π_l for a data unit can be understood in terms of a Markov decision process. At discrete times t_i the server makes an observation o_i and then takes a

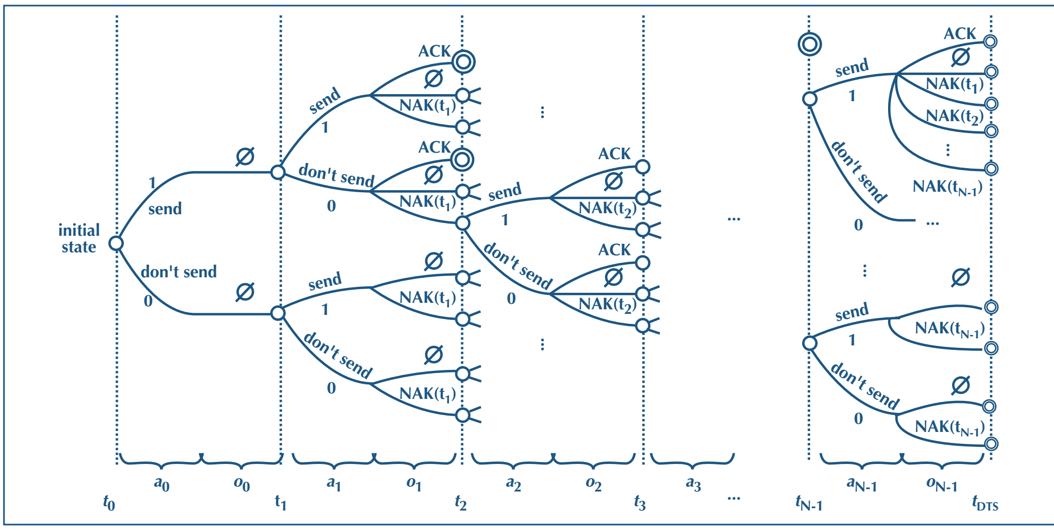

Figure 4: State space for the Markov decision process associated when rich feedback is used.

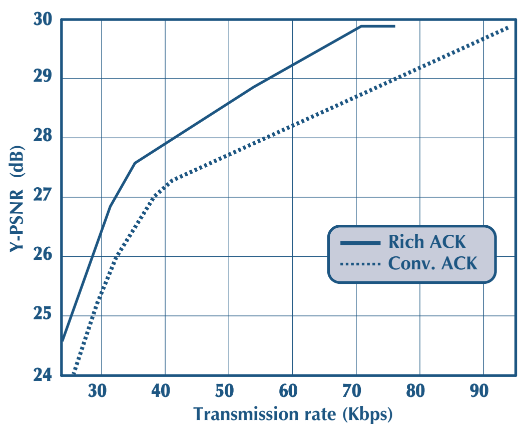

Figure 5: Rich vs. conventional acknowledgments for rate-distortion optimized streaming of QCIF Foreman

transmission action ai specifying send or not send. Sequences of observation and action pairs (o_i, a_i) in time can be enumerated in a Markov decision tree. Each node q_i in the tree specifies a particular history of observations and actions (a_0, o_0), (a_1, o_1),. …, (a_i, o_i). A transmission policy specifies what transmission action will be taken as a function of what state q_i is reached in the tree.

A Markov decision tree is shown in Fig. 4. The tree shown enumerates the possible sequences of observation-action pairs for the transmission of a data unit using the rich feedback scheme. In the tree, possible actions *a* are: send or don't send. Possible observations *o* are: no relevant feedback has arrived (Ø), a feedback packet has acknowledged the reception of the data unit (ACK), or a feedback packet has indicated that the packet has not been received by the feedback packet's send-time (NACK packets with different send-times are distinct observations). In contrast, in the conventional feedback scheme in which each packet is acknowledged individually upon receipt, there are only two possible observations, ACK and Ø. Regardless of the scheme, the optimization algorithm calculates the probabilities of each path through the tree given a policy, and then chooses the policy that yields the best tradeo between expected number of transmissions p (π_l) and loss probability $\in(\pi_l)$.

Fig. 5 compares average PSNR versus transmitted bit-rate for the 13-second *Foreman* sequence streamed using the rich feedback scheme and using the conventional acknowledgement scheme. Two-layer SNR-scalable H.263+ is used for the encoding[10]. The bit-rate of the base layer alone is 32 kbps with average PSNR 27 dB. When the enhancement layer is added, the encoded rate becomes 69 kbps with average PSNR is 30.5 dB.

The results are for simulation experiments with 10% loss rate both for media packets and feedback packets. Delays for packets not lost were distributed according to independent shifted-distributions with shift $\kappa = 50$ ms, mean $\mu = 25$ ms, and standard deviation $\sigma = 35$ ms.

In Fig. 5 we see that for all of the transmission rates considered the rich acknowledgement scheme outperforms the conventional RaDiO scheme. A maximum PSNR improvement of 1.3 dB is seen at a transmitted bit-rate of 70 kbps. The improved performance of the rich acknowledgement scheme is due to the redundancy in feedback information.

With rich feedback, the effect of a lost feedback packet is lessened because subsequent feedback packets will contain the same information. In addition, because the rich feedback packets also provide NACKs, there is less ambiguity for the server to resolve. In the case of conventional feedback, a non-acknowledged transmission may be due to a lost media packet or to a lost acknowledgement packet.

4. MULTIPLE DEADLINES

In the RaDiO framework described in Sec. 2, ΔD_l is the expected amount of distortion that is removed if data unit l is decodable by its deadline. An assumption made by the framework is that there is one deadline by which a data unit l must arrive in order for its distortion reduction ΔD_l to be realized, and in order for data units dependent on l to be decoded. Often, however, a data unit arriving after it its deadline is still useful for decoding. For instance, consider the case of bidirectional prediction with a sequence of frames I-B-B-B-P. In this example, the deadline, according to the RaDiO framework, for the P frame's data unit(s) would be related to the decoding time of the first B frame. If the P-frame's data unit(s) arrive late, however, it(they) may still be useful for decoding subsequent B frames. Thus there are several deadlines associated with the P-frame's data units [11]. Another example where a data unit may be associated with multiple deadlines is the case of decoders which allow Accelerated Retroactive Decoding (ARD). Proposed in another context in [12], ARD makes use of the ability of many streaming clients to decode video faster than real-time. With ARD, when late-arriving data units finally do arrive, the decoder goes back to the frames corresponding to the late arriving packets and in an *accelerated* manner re-decodes the dependency chain up to the current playout position, this time without error. In this way the remaining pictures in the GOP can be decoded and displayed error-free.

Fig. 6 shows performance gains due to the multiple deadline formulation in the case when ARD is implemented at the decoder. PSNR-versus-rate results are shown for the Foreman sequence streamed in a low-latency application in which the pre-roll delay is 100 ms. In the simulation experiments, frames of video were due for decoding 100 ms after they became available for

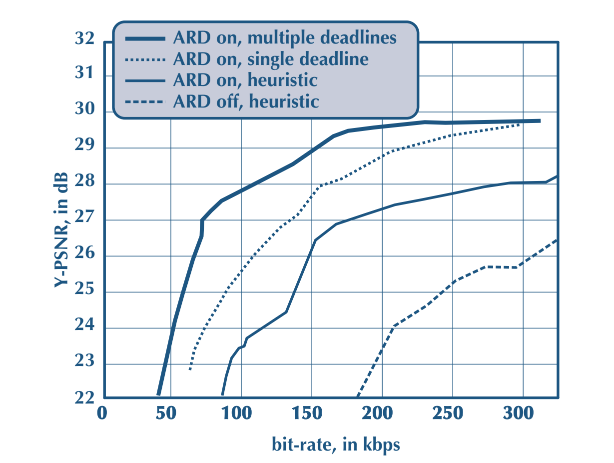

Figure 6: Rate-Distortion performance of schedulers for the Foreman sequence streamed over a simulated channel with iid shifted-Γ-distributed packet delays, and 20% Bernoulli loss. End-to-end latency D = 100 ms. A PSNR improvement of up to 3.15 dB is observed for the optimizing scheduler that considers multiple deadlines compared to the one that considers a single deadline.

transmission at the server. The packet loss rate was 20% in both directions, and delays for packets not lost were iid shifted-Γ with shift κ= 10 ms, mean μ = 40 ms, and standard deviation σ= 23 ms. The sequence was encoded using a two-layer SNR-scalable H.263+ [10], at 10 fps with prediction structure I-P-P-P ... and GOP length of 20 frames. The base and enhancement layer bit-rates and PSNRs were similar to those of the sequence in Sec. 3.

In the figure, PSNR-versus-rate traces are shown for the multiple deadline and single deadline schemes and for a heuristic, non-optimizing scheme. The heuristic scheme, providing a basis for comparison, used prioritized, deadline-limited ARQ in which base layer retransmissions had highest priority, followed by base layer transmissions, enhancement layer transmission and enhancement layer retransmissions. Retransmissions were triggered when packets were not ACKed within the 0.90 point of the round-trip time cdf. From Fig. 6 we see that the multiple deadline formulation yields up to a 3 dB improvement over the single deadline scheme. The improvement is due to the single-deadlines scheme's inability to recognize the value of a data unit arriving after its original deadline. The R-D optimizing schemes outperform the heuristic schemes regardless of whether the heuristic schemes are used with ARD-enabled decoders.

5. MARKOV DELAY MODELING

In the R-D optimized video streaming algorithms discussed in Sections 2, 3, and 4, the delays of successive packets have been modeled as independent, identically distributed (iid) shifted-Γ random variables, with loss also occurring independently as described in [1]. The iid model is advantageous because it simplifies calculations for $\in(\pi_l)$, the error probability due to a transmission policy, and for $p(\pi_l)$, the expected number of transmissions that will result from a transmission policy. The iid model, however, has the obvious shortcoming that it does not capture any dependence among successive delay times.

In streaming simulations that employ measured Internet delay traces, we have observed that the iid model can lead to sub-optimal scheduling performance. For example, Fig. 7 shows simulation results when packets were delayed according to a delay trace measured over a 14-hop Internet path with a cable modem last hop, as described in [13]. In the figure we see that at transmission rates above 80 kbps, the multi-deadline R-D optimizing formulation described in Sec. 4 is outperformed by the simple heuristic ARQ scheme (also described in Sec. 4). The reason for the suboptimal performance at high rates is due to the iid delay model assumed by the R-D optimization algorithm. With the iid model, policies which specify repeated transmission of a data unit at successive opportunities yield lower calculated error probabilities for errors due to late loss. The algorithm mistakenly believes that if the data unit is delayed the first time it is transmitted, subsequent transmissions may arrive earlier and on time. Thus at higher rates, the algorithm sends packets multiple times even though in our measured trace the loss probability is very low (0.014%), and packets always arrive in the order they are transmitted.

We have observed that R-D performance can be improved by modeling packet delays at successive transmission time-slots

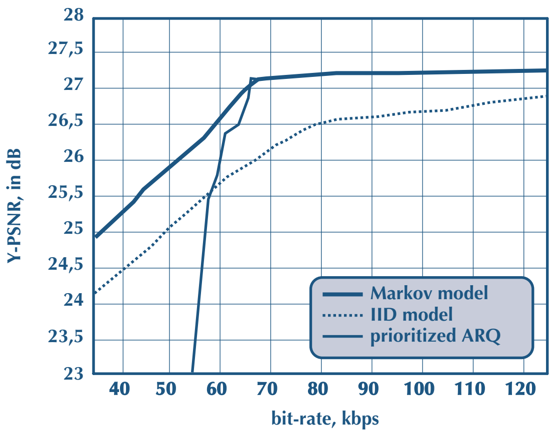

Figure 7: Rate-Distortion performance of schedulers for the Foreman sequence stramed over a measured trace file channel model. End-to-end latency d = 150 ms. The RaDio scheduler that models delay as iid is sub-optimal at high rates where it is outperformed by heuristic prioritized ARQ scheduler. The schedules that models delays as a Markov random process yields PSNR improvement of up to 1.1 dB iver the iid schedular.

as a first-order, discrete Markov random process [13]. In [14] we have presented an R-D optimization scheme that uses this model. In the scheme, feedback packets inform the server about the delay over the channel in the recent past. Using this feedback and a family of conditional delay distributions, the scheme can more accurately calculate the expected distortion $D(\pi)$ and the expected transmission rate $R(\pi)$ resulting from a transmission policy π.

Our scheme requires accurate estimates of conditional delay distributions $p\ (x_i |\ x_{i-1})$, where the random variables X_{i-1} and X_i are the delays for packets transmitted at successive transmission opportunities. We show in [13], that the family of conditional delay distributions, $p\ (x_i | x_{i-1})$, required by our scheme can be well-approximated with shifted- Γ distributions, with parameters taken from accumulated feedback information. Returning to Fig.7, we see that the R-D optimization scheme using the Markov channel model outperforms the RaDiO scheduler that uses iid delay modeling by up to 1.1 dB and is not outperformed by the heuristic scheduler at low rates. We note that the mean PSNR for all of the traces is limited by the fact that the measured delays in the 14-hop cable modem trace are often greater than the 150 ms pre-roll delay in this simulated low latency streaming application. Because the decoder for all traces uses the ARD scheme discussed in Sec. 4, and because the packet loss rate is nearly zero, the heuristic scheme, which uses time-out triggered retransmissions with the timeout set to 2 · (estimated RTT), actually performs nearly optimally at high transmission bit rates.

6. Congestion-Distortion Optimized Scheduling

A drawback of the RaDiO scheme and the extensions described above is that they do not consider the effect that transmitted media packets may have on the delay of subsequently transmitted packets. Delay is modeled as a random variable with a parameterized distribution; parameters are adapted slowly according to feedback information. In the case when the media stream is transmitted at a rate that is negligible compared to the minimum link speed on the path from server to client, this may be an acceptable model. In the case where there is a bottleneck link on the path from server to client, however, packet delays can be strongly affected by congestion resulting from previous transmissions.

In [15] a congestion-distortion optimized (CoDiO) algorithm is proposed which takes into account the effect of transmitted packets on delay. The scheme is intended to achieve similar R-D performance to the RaDiO scheme but specifically schedules packet transmissions in a way that yields an optimal (at least approximately) tradeoff between reconstruction distortion and congestion, measured as delay, on the bottleneck link.

As in the RaDiO schemes, transmission actions are chosen at discrete transmission opportunities by finding an optimal policy specifying transmission actions over a time horizon. In CoDiO, the optimal policy is the one that minimizes the Lagrangian cost $D + \lambda\Delta$ where D is the expected distortion due to the policy and Δ is the expected end-to-end delay which serves as the congestion metric in the scheme. As in RaDiO, the algorithm iteratively minimizes the Lagrangian with respect to the policy of one data unit at a time to avoid an exhaustive search of a policy space that grows exponentially both in the length of the time horizon and in the number of packets considered for transmission.

The performance of the CoDiO scheme has been evaluated using ns-2 simulation experiments [15, 16]. Using ns-2, a link consisting of two hops was simulated. The first hop was a high-bandwidth 45 Mbps link with 22 Mbps exponential cross-traffic. The second hop was a 50 kbps link which only carried the video traffic. In the scheduler, the delay of this channel was modeled as exponentially distributed with a time-varying shift reflecting the delay at the bottleneck. The scheduler dynamically estimated the shift to find parameters for this distribution, and calculated delay probabilities during the simulation according to the distribution. Fig. 8 also presents results for a scheduler that assumed a fixed value for the shift.

The video encoding used is the same as that described in Sec. 4. The pre-roll delay for the experiments was 600 ms. In Fig. 8, the traces plot luminance PSNR versus average end-to-end delay for the CoDiO and the RaDiO scheme.

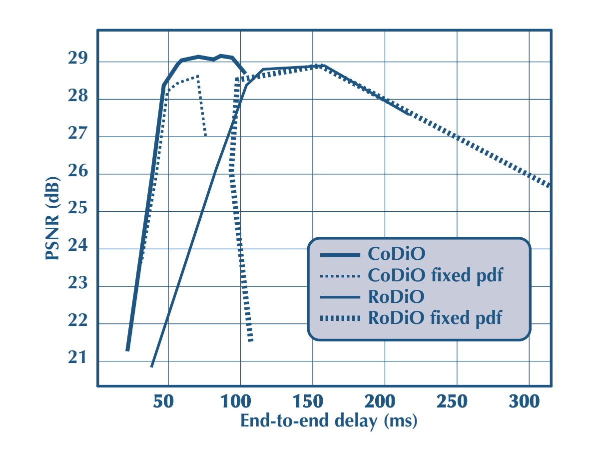

FIGURE 8: PERFORMANCE COMPARISON OF RADIO AND CODIO.

The various points on the curves were generated by varying λ which trades-off congestion-distortion in the case of CoDiO and rate-distortion in the case of RaDiO. The graphs show that the CoDiO scheme resulted in end-to-end delays that were approximately half of those measured for the RaDiO scheme at the same PSNR. Transmission rates versus PSNR for both schemes (not shown) were identical.

7. CONCLUSION

We have reviewed extensions and enhancements to the framework for rate-distortion optimized (RaDiO) media streaming [1] that have been recently developed in our research group at Stanford University, and we have shown experimental results for these new techniques.

We have shown that by improving the feedback mechanism in the RaDiO scheme through the use of rich acknowledgements, a boost of up to 1.3 dB in video PSNR is attainable. We have also shown a more than 3 dB gain in PSNR attained by extending RaDiO to consider multiple deadlines for packets. In Sec. 5, a significant modification to the RaDiO algorithm that utilizes a Markov model of successive packet delays has been shown to correct sub-optimal performance of the RaDiO algorithm which arises due to its simplistic channel model. Finally, we have discussed a method of optimized scheduling, CoDiO, that gives the same PSNR performance of RaDiO, but has been observed to reduce the congestion, measured in terms of end-to-end delay, of a bottleneck link by asmuch as 50%.

8. ACKNOWLEDGEMENTS

The generous support of our work by STMicroelectronics through their Founding Membership of the Stanford Networking Research Center (SNRC) is gratefully acknowledged.

REFERENCES

[1] P. A. Chou and Z. Miao, "RATE-DISTORTION OPTIMIZED STREAMING OF PACKETIZED MEDIA," (submitted) IEEE Transactions on Multimedia, 2001.

[2] A. Sehgal and P. Chou, "RATE-DISTORTION OPTIMIZED STREAMING OVER DIFFSERV NETWORKS," in Proceedings of the IEEE International Conference on Multimedia and Expo 2002, Lausanne, Switzerland, August 2002.

[3] M. Roder, J. Cardinal, and R. Hamzaoui, "ON THE COMPLEXITY OF RATE-DISTORTION OPTIMAL STREAMING OF PACKETIZED MEDIA," in Proceedings of the Data Compression Conference 2004, Snowbird, Utah, March 2004, pp. 253–262.

[4] A. C. Begen, Y. Altunbasak, and M. A. Begen, "RATE-DISTORTION OPTIMIZED ON-DEMAND MEDIA STREAMING WITH SERVER DIVERSITY," in IEEE International Conference on Image Processing, Barcelona, Spain, September 2003.

[5] J. Chakareski and B. Girod, "SERVER DIVERSITY IN RATE-DISTORTION OPTIMIZED MEDIA STREAMING," in IEEE International Conference on Image Processing, Barcelona, Spain, September 2003.

[6] M. Podolsky, S. McCanne, and M. Vetterli, "SOFT ARQ FOR LAYERED STREAMING MEDIA," Tech. Rep. UCB/CSD-98-1024, University of California, Computer Science Department, Berkeley, CA, November 1998.

[7] J. Chakareski and P. A. Chou, "APPLICATION LAYER ERROR CORRECTION CODING FOR RATE-DISTORTION OPTIMIZED STREAMING TO WIRELESS CLIENTS," in IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings ICASSP '02, Orlando, FL, May 2002.

[8] P. A. Chou and A. Sehgal, "RATE-DISTORTION OPTIMIZED RECEIVER-DRIVEN STREAMING OVER BEST-E ORTNETWORKS," in Packet Video Workshop, Pittsburgh, PA, USA, April 2002.

[9] J. Chakareski and B. Girod, "RATE-DISTORTION OPTIMIZED VIDEO STREAMING WITH RICH ACKNOWLEDGMENTS," in Proceedings SPIE Visual Communications and Image Processing VCIP-2004, Santa Clara, CA, January 2004.

[10] ITU-T, "VIDEO CODING FOR LOW BITRATE COMMUNICATION: RECOMMENDATION H.263, VERSION 2," 1998.

[11] S. Wee, W. Tan, J. Apostolopoulos, and M. Etoh, "OPTIMIZED VIDEO STREAMING FOR NETWORKS WITH VARYING DELAY," in Proceedings of the IEEE International Conference on Multimedia and Expo 2002, Lausanne, Switzerland, August 2002.

[12] M. Ghanbari, "POSTPROCESSING OF LATE CELLS FOR PACKET VIDEO," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 6, pp. 669–678, December 1996.

[13] M. Kalman and B. Girod, "MODELING THE DELAYS OF SUCCESSIVELY-TRANSMITTED INTERNET PACKETS," in IEEE Conference on Multimedia and Expo, Taipei, Taiwan, June 2004.

[14] M. Kalman and B. Girod, "CONGESTION-DISTORTION OPTIMIZED SCHEDULING OF VIDEO OVER A BOTTLENECK LINK," in IEEE Workshop on Multimedia Signal Processing, Sienna, Italy, September 2004.

[15] E. Setton and B. Girod, "CONGESTION-DISTORTION OPTIMIZED SCHEDULING OF VIDEO OVER A BOTTLENECK LINK," in IEEE Workshop on Multimedia Signal Processing, Sienna, Italy, September 2004.

[16] "THE NETWORK SIMULATOR - NS-2," www.isi.edu/nsnam/ns/.

■ CONTACT: ST.JOURNAL@ST.COM ■

