

# RATE-DISTORTION OPTIMIZED BANDWIDTH ADAPTATION FOR DISTRIBUTED MEDIA DELIVERY

*Jacob Chakareski and Pascal Frossard*

Signal Processing Institute - LTS4, EPFL, CH-1015 Lausanne

(Invited Paper)

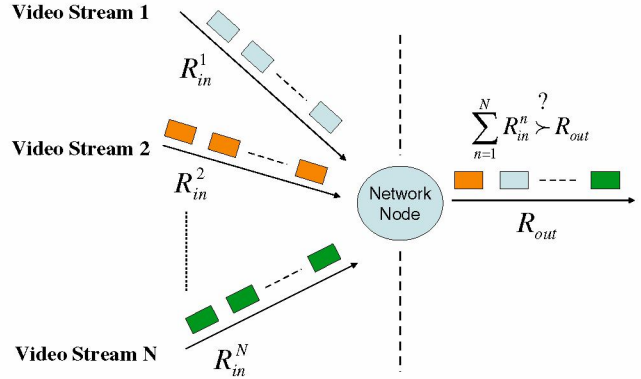
## ABSTRACT

We propose a framework for rate-distortion optimized bandwidth adaptation via packet dropping at a network node, when the incoming traffic at the node consists of multiple video streams. The framework enables the node to decide in a rate-distortion optimal sense, which packets, if any, from each stream should be discarded in order to adapt to the available outgoing bandwidth at the node, so that the overall video quality over all streams is maximized. The framework relies on a rate-distortion hint track information that is sent along with each video packet. The hint track information consists of two quantities: the size of the video packet in bits, and its importance for the reconstruction quality of the video stream. Experimental results demonstrate that our framework provides significant gains in video quality, both over all streams jointly and also over the individual videos, relative to a conventional system for bandwidth adaptation that does not take into account the different importance of the individual video packets.

## 1. INTRODUCTION

Bandwidth adaptation is a commonplace in the Internet today. It is encountered whenever the data rate on the incoming link at a network node exceeds the data rate on the outgoing link. Buffer management during transient periods of network congestion when queues overflow and transcoding at the junction point of two heterogeneous (in terms of available bandwidth) networks are two principal examples of bandwidth adaptation. In this paper, we consider the scenario of bandwidth adaptation via packet dropping at a network node when the incoming traffic at the node consists of multiple video streams that are multiplexed by the node on a single outgoing link. Such a scenario occurs whenever the incoming data rate over all streams exceeds the available outgoing data rate at the node. The distributed streaming system, as represented by the node, is interested then in optimizing the overall quality over all streams, for the given resources, as represented by the available bandwidth on the outgoing link. The scenario under consideration is illustrated in Figure 1.

Selecting which packets to drop in order to account for the reduction in available data rate can be an involved task as randomly dropping packets can have an unpredictable effect on the reconstruction quality of the video stream at the final destination. Scalable coding techniques have been developed to solve this problem, where the scalable encoding provides an inherent prioritization among the compressed data which in turn provides a natural method for selecting which portions of the compressed data to deliver while meeting the data rate constraints along the network



**Fig. 1.**  $N$  incoming video streams at a network node that have to be multiplexed on a single outgoing link.

path between the sender and the receiver. In addition, conventional MPEG coding with I, P, and B frames also lends itself to a natural method of prioritizing the delivery. However, performing bandwidth adaptation with non-scalable or non-prioritized packetized video content presents a more challenging problem as the compressed data does not suggest a straightforward way of placing delivery priorities on the video packets. In this paper, we focus on the problem of bandwidth adaptation for non-scalably coded video streams.

We propose a framework for rate-distortion optimized packet dropping for bandwidth adaptation of multiple video streams at a network node, as shown in Figure 1. The framework relies on a hint track information that is sent with each video packet. The hint track consists of two quantities: the size of the packet in bits, that is usually available in packet headers, and the importance of the packet in terms of the reconstruction distortion for the video stream. Using the framework and based on the hint track information, a network node, whether a proxy server or a gateway, can make optimal dropping decisions such that the end-to-end performance in terms of video quality over all streams is maximized, for the given available outgoing data rate at the node. The framework enables the node to trade-off rate and distortion not only over packets within a single video stream, but also over packets that belong to different video streams.

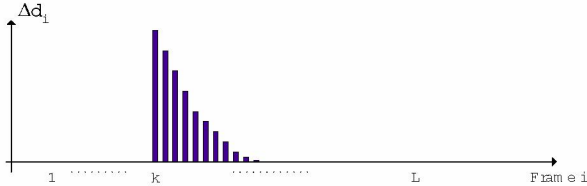
The most closely related contemporaneous works are the following. [1] proposes a strategy for dropping packets from a single incoming video stream that is encoded using the wavelet transform. Reduction in data rate is achieved either by dropping whole video frames, thereby reducing the temporal frame rate of the video, or by preferentially dropping packets carrying higher frequency bands of the encoded frames. No rate-distortion optimization is performed. In [2, 3], the authors study bandwidth adaptation via

This work has been partly supported by the Swiss National Science Foundation.

packet dropping for MPEG-2 encoded video and propose dropping strategies which in essence place different delivery priorities on the different frame types of the encoded video: I, P and B. Only a single video stream is considered and no rate-distortion optimization is performed. Another related work is [4], which examines rate-distortion optimized dropping of whole video frames from multiple video streams. The videos are non-scalably encoded using an H.264 codec. Only up to four possible drop patterns from a single stream are considered depending on the type of frame (I, P or B) that is dropped at present from the same stream. [5] considers rate-distortion optimized packet dropping in the context of proxy-caching for broadcasting of an MPEG-4 encoded single video stream. Our work is probably most closely related to [6] which introduced the concept of rate-distortion hint tracks for video streaming. In fact, the present paper can be considered as an extension of [6] to bandwidth adaptation for multiple video streams.

## 2. RATE-DISTORTION HINT TRACK INFORMATION

Let  $k$  be the index of a packet from a video stream. Then, as explained earlier, the rate-distortion (R-D) hint track (or side information) associated with packet  $k$  consists of the size of packet  $k$  in bits  $R(k)$  and the importance of packet  $k$  for the reconstruction distortion of the video stream denoted as  $D(k)$ . Specifically,  $D(k)$  is the total increase in MSE distortion that will affect the video stream if packet  $k$  is not delivered to the receiver, and is computed as  $D(k) = \sum_{i=1}^L \Delta d_i$ , where  $L$  is the number of packets in the stream and  $\Delta d_i$  is the increase in MSE distortion associated with packet  $i$  given that packet  $k$  is missing at the receiver. Note that  $\Delta d_i = 0$  for  $i < k$ . In Figure 2 we illustrate the distortions  $\Delta d_i$  for the loss of packet  $k$ , where for clarity of presentation it is assumed that each packet  $i$  corresponds to a video frame.



**Fig. 2.** Loss of frame  $k$  induces distortion in later frames.  $D(k)$  is the total distortion summed over all affected frames of the stream.

It can be seen from the figure that the MSE per frame ramps up at frame  $k$ , which is expected since the missing frame  $k$  is replaced with frame  $k - 1$  and there are no prior losses. Here, we assume that previous frame concealment is used for missing frames. Due to error propagation, which in turn is caused by the predictive nature of the encoding process, the MSE associated with subsequent frames also exhibits a nonzero value, as shown in Figure 2. However, due to the effects of spatial filtering and intra refresh [7], its amplitude gradually decreases over successive frames, till it finally becomes zero at frame  $j > k$  sufficiently apart from  $k$ .

## 3. R-D OPTIMIZED PACKET DROPPING

Let there be  $N$  video streams whose packets arrive at a network node on separate links, as shown in Figure 1. Assume that at present the node has a window  $\mathcal{W}$  of packets belonging to all

streams. The node needs to decide on dropping a subset of packets from  $\mathcal{W}$  in order to accommodate the reduced data rate that is available on the outgoing link.

Let  $\mathbf{k} = \{k_1, k_2, \dots, k_P\}$  be an arbitrary subset from  $\mathcal{W}$ . The total MSE distortion that will affect jointly all video streams if  $\mathbf{k}$  is dropped at the node can be computed as:

$$\tilde{D}(\mathbf{k}) = \sum_{j=1}^P D(k_j) \quad (1)$$

Note that the above model assumes additivity of the distortions associated with the individual packet losses, ignoring any interdependencies between their effects on the distortion, which does not necessarily hold true when individual packet losses are not spaced sufficiently far apart with respect to the intra-refresh period, as recognized for example in [8]. Still, due to its simplicity and yet good accuracy, the additive model has found a number of applications in streaming and modelling of packetized media, such as [6, 9, 10].

Let  $R^*$  be the data rate constraint on the outgoing link, measured either in bits or in number of packets. We need to decide on the subset of packets  $\mathbf{k} \in \mathcal{W}$  that should not be transmitted in order to satisfy the data rate constraint. Let  $R(\mathcal{W} \setminus \mathbf{k}) = \sum_{i \in \mathcal{W} \setminus \mathbf{k}} \gamma(i)R(i)$  be the cost-weighted rate associated with all the packets  $i$  from  $\mathcal{W}$  that will be transmitted on the outgoing link, where “ $\setminus$ ” denotes the operator “set difference”, and  $\gamma(i)$  is the weighting cost factor for packet  $i$  that depends on the content provider’s policy. For example,  $\gamma(i) < 1$  means that packet  $i$  is more important for the reconstruction quality of the corresponding video stream and should be given priority. Thus, we are interested in finding the subset  $\mathbf{k}$  such that the total distortion due to dropping  $\mathbf{k}$  is minimized, while meeting the data rate constraint, i.e.,

$$\mathbf{k}^* = \arg \min_{\mathbf{k} \in \mathcal{W} : R(\mathcal{W} \setminus \mathbf{k}) \leq R^*} \tilde{D}(\mathbf{k}) \quad (2)$$

We solve for  $\mathbf{k}^*$  by casting (2) as a non-constrained optimization problem using a Lagrangian multiplier ( $\lambda > 0$ ):

$$\mathbf{k}^* = \arg \min_{\mathbf{k} \in \mathcal{W}} \tilde{D}(\mathbf{k}) + \lambda R(\mathcal{W} \setminus \mathbf{k}). \quad (3)$$

It can be shown that the solution to (3) reduces to sorting in increasing order the packets  $i \in \mathcal{W}$  based on their distortion per unit rate utility defined as  $\lambda_i = D(i)/(\gamma(i)R(i))$ . In the case when  $R^*$  is measured in bits, the solution proceeds by dropping all the packets  $i$  from the start of the rank ordering for which it holds  $\lambda_i \leq \lambda$ . On the other hand, when  $R^*$  is measured in number of packets, then  $R(i) = 1$  for  $i \in \mathcal{W}$  and the solution simplifies to selecting (to drop) the first  $|\mathcal{W}| - R^*$  packets from the rank ordering.

## 4. EXPERIMENTAL RESULTS

In this section, we examine the performance of the proposed framework for rate-distortion optimized packet dropping denoted henceforth *RDOpt*. The video sequences used in the experiments are coded using JM 2.1 of the JVT/H.264 video compression standard [11]. Four standard test sequences in QCIF format are used: Foreman, Carphone, Mother & Daughter, and Salesman. Each has at least 300 frames at 30 fps, and is coded with a constant quantization level at an average luminance (Y) PSNR of about 36 dB. The specific rate-distortion encoding characteristics for the four sequences are shown in Table 1. The first frame of each sequence is intra-coded, followed by all P-frames. Every 4 frames a slice is

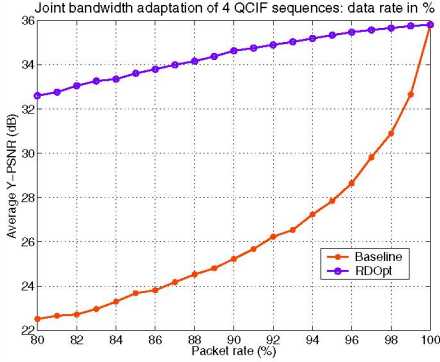


intra updated to improve error-resilience by reducing error propagation (as recommended in JM 2.1), corresponding to an intra-frame update period of  $M = 4 \times 9 = 36$  frames. An identical cost weight  $\gamma = 1$  is applied across all packets for every stream.

Sequence	Rate (Kbps)	Y-PSNR (dB)
Foreman	157.45	35.69
Carphone	171.30	36.60
Mother & Daughter	63.79	36.21
Salesman	64.31	35.01

**Table 1.** Encoding characteristics of the four sequences.

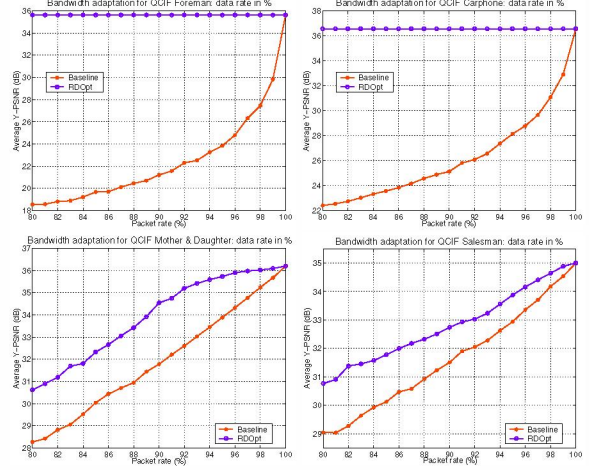
We also study the performance of a conventional system for packet dropping denoted as *Baseline*, which does not consider the distortion importance of different packets. In particular, when making packet dropping decisions, *Baseline* does not distinguish between two packets related to two different P frames, except for the size of the packets. *Baseline* randomly chooses between two P-frame packets of the same size, when adapting to reductions in available data rate on the outgoing link. In both systems, *RDOpt* and *Baseline*, video packets on each of the four incoming links are considered for dropping in non-overlapping windows of size 25, which means that the overall window  $\mathcal{W}$  introduced in Section 3 contains a total of 100 packets from all four streams. We examine first the performances of the two systems when the available data rate on the outgoing link is expressed as a percentage of packets from  $\mathcal{W}$  that can be sent on the link.



**Fig. 3.** Y-PSNR (dB) vs. Packet rate (%).

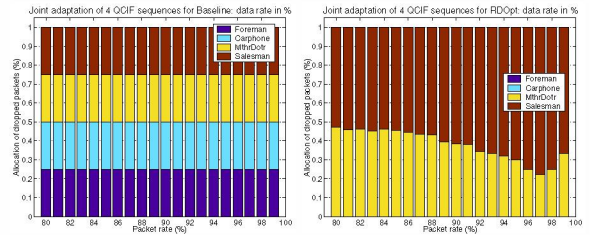
Figure 3 shows the overall performances of *RDOpt* and *Baseline* over all four sequences as a function of the available packet rate measured in percent. For example, packet rate of 99% means that 99% of the packets from  $\mathcal{W}$  can be forwarded. It can be seen that *RDOpt* outperforms *Baseline* with quite a significant margin over the whole range of values considered for the available packet rate. This is due to the fact that *RDOpt* exploits the knowledge about the effect of loss of individual video packets on the reconstructed video quality. Therefore, *RDOpt* drops the packets from  $\mathcal{W}$  that will have the least impact on the overall quality of the reconstructed videos. As can be seen from the figure, the performance gains of *RDOpt* over *Baseline* increase as the available packet rate decreases. For example, at packet rate of 90%, the performance improvement due to the optimized packet dropping decisions is close to 10 dB, which is quite impressive.

Next, in Figure 4 we show the performances of *RDOpt* and *Baseline* for the individual sequences. It can be seen from the



**Fig. 4.** Y-PSNR (dB) vs. Packet rate (%) for (top left) Foreman, (top right) Carphone, (bottom left) Mother & Daughter, and (bottom right) Salesman.

figure that also for the individual sequences a significant improvement in performance is observed relative to *Baseline* when packets are dropped in a distortion optimal way. For example, the gains over *Baseline* at packet rate of 90% are 14 dB, 11 dB, 2.7 dB, and 1.5 dB respectively for Foreman, Carphone, Mother & Daughter, and Salesman. Furthermore, the results from Figure 4 clearly depict how *RDOpt* trades-off packet-rate and distortion across the different sequences. Specifically, no packets have been dropped from Foreman and Carphone by *RDOpt* over the whole range of packet rate values under consideration. That is because these two sequences exhibit a lot of motion and therefore will exhibit a significant reduction in quality even for a small number of dropped packets. On the other hand, the sequences Mother & Daughter and Salesman have low complexity, which means error concealment can be applied quite successfully on their missing packets. Hence, *RDOpt* decided to drop exclusively packets from these two latter sequences.



**Fig. 5.** Allocation (%) of dropped packets across sequences vs. Packet rate (%) for (left) *Baseline* and (right) *RDOpt*.

The allocation of dropped packets to the individual sequences as a function of the packet rate on the outgoing link for both, *RDOpt* and *Baseline*, is shown in Figure 5. In essence, the figure summarizes what we have discussed in the previous paragraph: *Baseline* drops packets from the different sequences uniformly across the individual sequences, while *RDOpt* does that preferentially.

Figure 6 shows the overall performances of *RDOpt* and *Baseline* over all four sequences as a function of the available data rate (Kbps). It can be seen that *RDOpt* again outperforms *Baseline* with

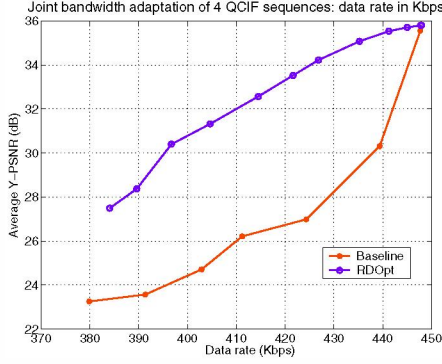


Fig. 6. Y-PSNR (dB) vs. Data rate (Kbps).

quite a significant margin over the whole range of data rate values. As explained earlier, this is due to the fact that *RDOpt* can trade-off rate and distortion in an optimal way by exploiting the hint track information associated with every packet. As can be seen from the figure, the performance gains of *RDOpt* over *Baseline* increase as the available data rate decreases. For example, at data rate of 410 Kbps, *RDOpt* outperforms *Baseline* with a margin of 6 dB.

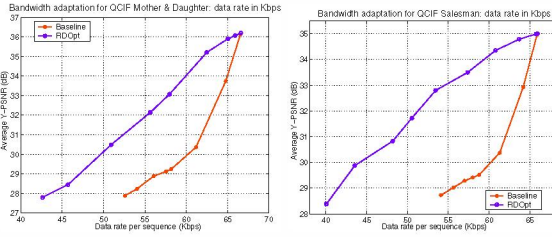


Fig. 7. Y-PSNR (dB) vs. Data rate per sequence (Kbps) for (left) Mother & Daughter, and (right) Salesman.

Next, in Figure 7 we show the performances of *RDOpt* and *Baseline* for Mother & Daughter, and Salesman. It can be seen from the figure that also for the individual sequences a significant improvement in performance is observed relative to *Baseline* when packets are dropped in a rate-distortion optimal way. For example, when these two sequences are transmitted on the outgoing link at 60 Kbps each, gains of 4 dB are registered over *Baseline*. Note that we do not show in Figure 7 the corresponding results for Foreman and Carphone. That is due to the fact that no packets from Foreman and only a few packets from Carphone are dropped by the optimization algorithm. As explained earlier, these two sequences are quite complex and therefore can affect the overall performance significantly even for a small number of dropped packets. Hence, *RDOpt* trades-off packets from Mother & Daughter, and Salesman for those of Foreman and Carphone in order to maximize the overall performance over all sequences.

Finally, the allocation of data rates to the individual sequences as a function of the overall data rate on the outgoing link for both, *RDOpt* and *Baseline*, is shown in Figure 8. It can be seen from the figure that in essence *Baseline* allocates rates in proportion to the encoding rate of each sequence and independently of the data rate on the outgoing link. On the other hand, *RDOpt* assigns increasingly larger shares of the overall rate to Foreman and Carphone, as the data rate is decreased. This is expected and is due to the fact that these two sequences have a more significant impact on the

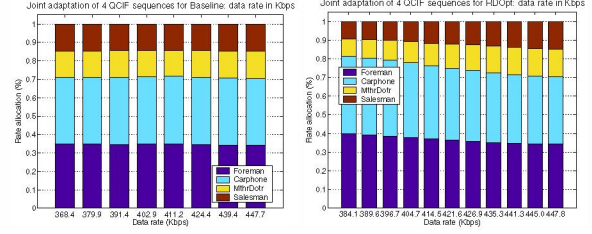


Fig. 8. Allocation (%) of the available data rate (Kbps) on the outgoing link for (left) *Baseline* and (right) *RDOpt*.

overall performance, as explained earlier. As the data rate is increased, *RDOpt* gradually decreases the shares allocated to Foreman and Carphone, and increases those for Mother & Daughter, and Salesman. This is due to the fact that at these overall data rates there is already enough rate for the former two sequences, so the optimization algorithm can allocate now increasingly more rate to the less important sequences, i.e., the latter two.

## 5. CONCLUSIONS

A framework for rate-distortion optimized bandwidth adaptation via packet dropping from multiple incoming video streams at a network node is presented. The framework enables the node to perform optimal dropping decisions so that the overall video quality across all sequences is maximized for the given available data rate on the outgoing link. Significant gains in performance on the order of several dBs, both jointly across all the videos and also for the individual streams, are registered over conventional systems for packet dropping which do not take into account the distortion information associated with the video packets.

## 6. REFERENCES

- [1] R. Keller, S. Choi, M. Dasen, D. Decasper, G. Fankhauser, and B. Plattner, "An active router architecture for multicast video distribution," in *Proc. INFOCOM*, Tel-Aviv, Israel, Mar. 2000, IEEE, vol. 3, pp. 1137–1146.
- [2] R.G.N. Balakrishnan and K.R. Ramakrishnan, "Active router approach for selective packet discard of streamed MPEG video under low bandwidth conditions," in *Proc. ICME*, New York, NY, July 2000, IEEE, vol. 2, pp. 739–742.
- [3] Y. Bai and M.R. Ito, "Network-level loss control schemes for streaming video," in *Proc. ICME*, Taipei, Taiwan, June 2004, IEEE.
- [4] W. Tu, W. Kellerer, and E. Steinbach, "R-D optimized video frame dropping on active network nodes," in *Proc. Packet Video Workshop*, Irvine, CA, Dec. 2004.
- [5] I. Bouazizi, "Size-distortion optimized proxy caching for robust transmission of MPEG-4 video," in *Proc. Int'l Workshop on Multimedia Interactive Protocols and Systems (MIPS)*, Napoli, Italy, Nov. 2003, pp. 131–142.
- [6] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "R-D hint tracks for low-complexity R-D optimized video streaming," in *Proc. Int'l Conf. Multimedia and Exhibition*, Taipei, Taiwan, June 2004, IEEE.
- [7] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE JSAC*, vol. 18, no. 6, June 2000.
- [8] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. VCIP*, San Jose, CA, Jan. 2001, SPIE, vol. 4310, pp. 329–409.
- [9] E. Masala and J.C. de Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. ICME*, Baltimore, MD, USA, July 2003, IEEE, vol. 3, pp. 345–348.
- [10] J. Chakareski, J. Apostolopoulos, W.-T. Tan, S. Wee, and B. Girod, "Distortion chains for predicting the video distortion for general packet loss patterns," in *Proc. ICASSP*, Montreal, Canada, May 2004, IEEE, vol. 5, pp. 1001–1004.
- [11] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.264*, Mar. 2003.