Rate-Distortion Hint Tracks for Adaptive Video Streaming

Jacob Chakareski, John G. Apostolopoulos, *Member, IEEE*, Susie Wee, *Member, IEEE*, Wai-tian Tan, *Member, IEEE*, and Bernd Girod, *Fellow*

***Abstract*—We present a technique for low-complexity rate-distortion (R-D) optimized adaptive video streaming based on the concept of rate-distortion hint track (RDHT). RDHTs store the precomputed characteristics of a compressed media source that are crucial for high performance online streaming but difficult to compute in real time. This enables low-complexity adaptation to variations in transport conditions such as available data rate or packet loss. An RDHT-based streaming system has three components: 1) information that summarizes the R-D attributes of the media; 2) an algorithm for using the RDHT to predict the distortion for a feasible packet schedule; and 3) a method for determining the best packet schedule to adapt the streaming to the communication channel. A family of distortion models, denoted distortion chains, are presented which accurately predict the distortion produced by arbitrary packet loss patterns. Two distortion chain models are examined which lead to two RDHT-based techniques. We evaluate the proposed techniques for two canonical problems in streaming media, adaptation to available data rate and to packet loss. Experimental results demonstrate that for the difficult case of nonscalably coded H.264 video, the proposed systems provide significant performance gains over conventional low-complexity streaming systems, and achieve this gain with a comparable level of complexity making them suitable for online R-D optimized streaming.**

***Index Terms*—Distortion modeling, hint track (HT), low-complexity, multimedia streaming, rate-distortion (R-D), video adaptation, video coding.**

I. Introduction

VIDEO adaptation for streaming over data rate constrained and lossy packet networks has been a practically important and challenging problem for a number of years. Video streaming typically involves pre-encoded and stored compressed media, and the pre-encoded content makes it harder to adapt to the available data rate and packet loss as compared to the case where real-time encoding is performed. Video transcoding can be performed in this situation, however this requires significant complexity and computation. Scalable coding can also be used to address this situation, since it 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 transmission rate constraints, however scalable video coding typically is afflicted by a significant penalty in compression efficiency.

A variety of techniques have recently been proposed to address the problem of adaptive and error-resilient video streaming, including intra/intermode switching [1], [2], dynamic control of prediction dependencies, forward error correction [3] and multiple description coding. One important recent advance in streaming technology is the emergence of rate-distortion optimized (RaDiO) streaming techniques that take into account packet importance and knowledge about the channel using a Lagrangian R-D cost function $J = D + \lambda R$. In this approach, 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 [4]. The flexibility of the framework has allowed its application to a number of streaming scenarios. Still, there were some important limitations of the initial framework that were overcome by an advanced framework for RaDiO video streaming proposed in [5]. Using this framework, advanced streaming scenarios such as streaming over multiple network paths [5], distributed streaming from multiple servers, streaming from an intermediate network proxy, and most recently streaming with rich acknowledgments and rich requests have been addressed. In general, however, the performance improvements due to the RaDiO streaming come at the price of increased computational complexity due to the optimization framework employed for computing the optimal schedules. This effect is exacerbated by the fact that optimal packet schedules need to be recomputed at every packet transmission opportunity. Therefore, conventional RaDiO techniques appear quite inappropriate for online optimized video streaming.

To address this issue in the present paper we propose a method for designing and operating media streaming systems that can perform optimized streaming while still being low complexity. Specifically, during encoding of a video sequence, a rate-distortion hint track (RDHT) is generated that contains side information that is often difficult to compute on a realtime basis,

Manuscript received June 7, 2004; revised March 20, 2005. This work was performed in part when J. Chakareski was at Hewlett-Packard Laboratories, Palo Alto, CA. This work was presented in part at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, QC, Canada, May 2004, at the International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, R.O.C., July 2004, and at the International Conference on Image Processing (ICIP), Singapore, October 2004. This paper was recommended by Associate Editor F. Pereira.

J. Chakareski was with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA. He is now with the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland (e-mail: cakarz@stanford.edu).

J. G. Apostolopoulos, S. Wee, and W.-T. Tan are with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: japos@hpl.hp.com; dtan@hpl.hp.com; swee@hpl.hp.com).

B. Girod is with the Information Systems Laboratory, Stanford University, Stanford, CA 94304 USA.

Digital Object Identifier 10.1109/TCSVT.2005.854227

but is useful to a general optimized streaming algorithm. The RDHT is a "track" because it is stored in the same file as the compressed media data but can be easily demultiplexed. It is a "hint track" in the sense that it provides "hints" for performing high quality streaming. Example information in the RDHT includes the importance of each packet in an R-D sense. The computation of hints at encoding time relieves the burden of optimized streaming servers, which can simply read the hints from the RDHT rather than estimating them on a realtime basis.

The term "hint track" has been used in the popular MPEG-4 File Format (MP4) [6], and in related streaming systems. An MP4 hint track contains information about media type, packet framing and timing information. With an MP4 hint track, media streaming is greatly simplified, both in terms of complexity and computation. This is because the streaming server no longer needs to 1) understand the compressed media syntax and 2) analyze the media data in realtime to generate packet framing and timing information.

The RDHT is also designed to reduce the complexity of streaming, but unlike the conventional hint tracks which simplify packet framing and timing, in our case the goal is to enable low-complexity RaDiO streaming. Specifically, the R-D attributes of the media are computed and summarized in the RDHT to enable low-complexity RaDiO streaming. MP4 hint track already has syntax to optionally associate a single priority value for each packet, providing a limited basis for optimization. The notion of RDHT is independent of MPEG-4 file format, even though RDHT can be potentially implemented using MPEG-4 file format.

Related work on low-complexity and RaDiO streaming is [7], where R-D information is placed in each packet header, thereby enabling efficient RaDiO streaming and adaptation at the sender, or at a mid-network node or proxy, for scalable media content. In addition, [8] proposed a framework for scalable media delivery, with similar attributes to the online optimization algorithms of, e.g., [4], but with a fast greedy search algorithm (and computationally simpler distortion metric) for determining the transmission schedule with significantly lower complexity.

In this paper we design a technique for low-complexity RaDiO streaming that employs RDHT information. The technique computes optimal packet schedules in a Lagrangian setting with a dramatically reduced complexity as compared to conventional RaDiO streaming, e.g., [4], [5]. We examine two instances of the proposed technique depending on the employed RDHT information. Each one of them can be applied toward solving two canonical problems in media streaming: data rate adaptation and packet loss adaptation. In addition to examining the conventional problem of RaDiO video streaming where the goal is to minimize the distortion subject to a bit-rate constraint, we also show how the proposed framework can be applied to minimize the distortion subject to a packet-rate constraint. While the packet-rate constraint problem is significantly different from R-D theory, it is an important problem for practical video streaming systems. Our work has been presented in part in [9]–[11]. The present paper extends our prior work by examining higher order distortion models, by evaluating performance in practical scenarios with packet loss and variable delay on both forward and backward channels leading to delayed and imperfect knowledge of packet loss, and by providing examples of the RDHT information for two distortion models.

The rest of the paper is structured as follows. We continue in Section II by presenting an overview of the general technique of RDHT-based media streaming. The RDHT-based technique is composed of three components, where the first two involve a method for predicting the reconstructed video quality at the receiver as a function of the received packets, and the third component is a search for the best packet transmission schedule to maximize the quality. To address the first two components of RDHT-based systems, Section III presents a mathematical framework, denoted distortion chains, which is designed to predict the distortion caused by an arbitrary packet loss pattern based on a small number of measurements involving lost video packets. In Section IV we present two instances of the proposed RDHT-based technique for low-complexity RaDiO streaming, which are based on two distortion chain models. The performance of these proposed approaches is evaluated in Section V through simulation experiments involving H.264 encoded packetized video. First, the accuracy of the distortion chains framework for predicting the distortion for different packet loss patterns is examined. Then, we examine the performance of the proposed RDHT-based techniques for streaming the packetized video content while adapting to variations in transport conditions such as available data rate or packet loss. In addition, we explore the performance gains that the technique provides over conventional non-RaDiO systems and the performance loss of this technique relative to conventional (high-complexity) RaDiO streaming systems. Finally, concluding remarks are provided in Section VI.

II. RDHT-Based Video Streaming

The central issue of optimized streaming is to determine the best packet schedule that maximizes the reconstructed quality at the receiver, subject to transmission constraints such as available data rate or packet loss. A system that employs RDHT achieves this goal using the following three components:

1) obtain information that summarizes the R-D attributes of the media, i.e., calculate the RDHT;
2) a method to use the RDHT information to predict the distortion for a feasible packet schedule, e.g., for a specific subset of received packets;
3) determine the best packet schedule for the channel.

There are a number of tradeoffs in the design of RDHT. Providing high performance requires calculating an "informative" RDHT, accurately modeling the distortion of different feasible packet schedules, and performing a comprehensive search for the best schedule. On the other hand, it is desirable to use relatively little storage and computation. This limits the amount of information represented by the RDHT. It also constrains the methods used for predicting the distortion of various packet schedules and searching for the best packet schedule for the channel.

Section III presents a family of models, referred to as distortion chain models, for predicting the mean square error (MSE) distortion in the reconstructed video at the receiver for different subsets of received packets. These models provide examples of the first two components of an RDHT-based streaming system listed above. In Section IV we investigate two instantiations of the proposed RDHT-based technique based on two different distortion chain models. We then examine this technique for two canonical problems in streaming media: 1) adapting to available data rate and 2) adapting to packet loss. Each one of these problems includes important subproblems. For example, in the context of adapting to available data rate, if the data rate constraint is measured in number of packets (ignoring packet size) that can be transmitted then the problem is simpler than if the data rate constraint is measured in bits, in which case there may be many more different subsets of different numbers of packets that must be examined for potential transmission. To adapt to the available data rate we must solve the problem of what is the best subset of packets to drop to meet a transmission rate constraint, which in turn can be given either in number of packets or in number of bits that we are allowed to transmit at present. To adapt to potential loss of previously transmitted packets we must solve the problem of what is the best schedule for transmitting new packets and retransmitting previous potentially lost packets to meet a transmission rate constraint.

III. Distortion Chains

Prior work on modeling the effect of packet loss on the reconstruction distortion of a video sequence at the receiver generally models the total distortion afflicting the video sequence as being proportional to the number of lost packets that occur [2], [12]. For example, a model is proposed in [2] for the total distortion associated with a single (isolated) packet loss which accounts for the effects of error propagation, intra refresh, and spatial loop filtering. Then, using this model the total distortion for multiple losses is defined as being proportional to the total number of losses. Specifically, with this *stationary linear* model, the expected total distortion (D_{Linear}) is computed as

$$D_{\text{Linear}} = \#\text{Losses} \cdot \frac{1}{L}\sum_{l=1}^{L} D(l) \tag{1}$$

where $D(l)$ is the total distortion that is associated with the loss of packet l (assuming that all other packets are correctly received), L is the total number of packets in the video sequence, and $(1/L)\sum_{l=1}^{L} D(l)$ is the average single packet loss total distortion. The quantity $D(l)$ is defined precisely later on in this section. Finally, given that the total number of losses is linearly (hence the name of the model) related to the average packet loss rate (PLR), i.e., $\#\text{Losses} = \text{PLR} \cdot L$ (for $\text{PLR} \leq 1$) we can write (1) as

$$D_{\text{Linear}} = \text{PLR} \cdot \sum_{l=1}^{L} D(l). \tag{2}$$

The linear model above is accurate for isolated losses that are sufficiently far apart, and for burst losses that do not result in the loss of more than one video frame. For example, this model is accurate for low-bit-rate video, where each coded video frame fits within a single packet, when single losses occur that are spaced sufficiently far apart with respect to the intra-refresh period,[1] e.g., when the loss rate is low and the losses are not bursty. This model is also accurate for burst losses, in the case of high-bit-rate video where each video frame fits within multiple packets, as observed in [13] for example, and only if the length of the burst loss is less than the number of packets required to send a single frame. That is, this linear model is accurate as long as the burst loss does not lead to the loss of more than a single frame in a row.

However, in many important applications, for example low-bit-rate video communication (where each coded frame may fit within a single packet) over the Internet or over a wireless link, there may be burst losses that result in the loss of multiple frames. In [14], it was recognized that the length of a burst loss has an important effect on the resulting distortion, where longer burst lengths generally led to larger distortions. This was extended in [15] where a simple model was proposed that distinguishes loss events based on the length of the burst loss and explicitly accounts for the different distortions that result for different burst lengths. In [16], a model is proposed that builds upon the prior work by capturing the correlation between the error frames associated with single (isolated) packet losses in order to describe more accurately the distortion resulting from a burst loss pattern.

In the following, we propose a model, which we refer to as the distortion chains model, for predicting the MSE distortion at the receiver in the event of packet loss. This model provides a simple, causal approach for predicting the distortion in the reconstructed video for general packet loss patterns.

A. Distortion Produced by Packet Loss

We first introduce some necessary notation and background. We analyze the case where a video sequence starts with an I-frame, followed by P-frames that have a certain number of macroblocks periodically Intra updated for increased error-resilience. For simplicity, we assume that each frame is coded into a single packet, so that the loss of a packet corresponds to the loss of an entire frame. This corresponds to the practically important case of low bit rate video communication over lossy packet networks, e.g., a 30 f/s QCIF video clip at 120 kb/s yields an average packet size of 500 bytes, which can be transported as a single packet in many networks. However, the results in this paper can also be extended to the case when each frame is coded into multiple packets.

To simplify notation, the two-dimensional (2-D) array of $M_P = M_{P1} \times M_{P2}$ pixels in each frame k are sorted in the one-dimensional (1-D) vector $f[k]$ (of length M_P) in line-scan order. We use the 1-D vector $f[k]$ to represent an original video frame, $\hat{f}[k]$ to denote the loss-free reconstruction of the frame, and $g[k]$ to denote the reconstruction at the decoder after loss concealment. The error frame at frame k introduced by one or more packet losses that occurred earlier is defined as

$$e[k] = g[k] - \hat{f}[k]$$

[1] The intrarefresh period is the number of frames between successive intra-coded (I) frames plus one when I-frames are used, or the number of frames until all of the macroblocks in a frame have been intra coded, when partial intra coding is used.

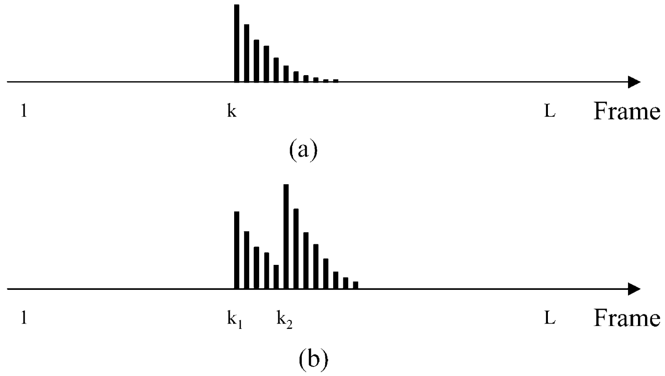


Fig. 1. (a) Loss of single frame k induces distortion in later frames. $D(k)$ is the total distortion summed over all affected frames. (b) $D(k_1, k_2)$ is total distortion summed over all frames caused by losing frames k_1 and k_2.

which is also a 1-D vector. We assume previous frame loss concealment, however please note that the proposed RDHT framework can handle any specific concealment strategy that is known a priori. Therefore, if frame k is the first occurrence of packet loss then $g[k] = \hat{f}[k-1]$. Since our primary concern is the effect of channel loss, quantization error is not included in our study. Finally, the MSE associated with error frame $e[k]$ is given by $\sigma^2[k] = (e^T[k] \cdot e[k])/M_P$.

The MSE above quantifies the error power introduced in a single frame due to previous packet losses. Now, let L be the length of a video sequence in frames and let $\boldsymbol{k} = (k_1, k_2, \ldots, k_N)$ denote a loss pattern of length N, i.e., N frames are lost during transmission where $k_i < k_j$, for $i < j$, are indexes of lost frames. Then, the total distortion, denoted by $D(\boldsymbol{k})$, due to the loss pattern is the sum of the MSE's over all the frames affected by the loss pattern $\boldsymbol{k}$, i.e.,

$$D(\boldsymbol{k}) = \sum_{l=1}^{L} \sigma^2[l] = \sum_{l=k_1}^{L} \sigma^2[l]. \tag{3}$$

For example, Fig. 1(a) illustrates the distortion $D(k)$ that afflicts a video sequence caused by the loss of frame k. It can be seen 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. Due to error propagation, which in turn is caused by the predictive nature of the encoding process as explained earlier, the MSE associated with subsequent frames also exhibits a nonzero value, as shown in Fig. 1. However, due to the effects of spatial filtering and intra refresh [2], its amplitude gradually decreases over successive frames, till it finally becomes zero at frame $j > k$ sufficiently apart from k.

Similarly, Fig. 1(b) illustrates the total distortion $D(k_1, k_2)$ introduced in the video sequence by losing frames k_1 and k_2, for $k_1 < k_2$. As shown in [16], the MSE $\sigma^2[j]$ for frames $j = k_2, k_2 + 1, \ldots$, of a video sequence, when frame k_1 is already lost prior to k_2, can be smaller, equal, or larger than the corresponding MSE for the case when there are no packet losses prior to k_2, depending on the particular video sequence in question, its encoding parameters and the distance in number of frames between k_1 and k_2. We will elaborate more on this in Section IV-B.

B. Distortion Chain Model for Predicting Distortion

We define $D(k_{N+1}|\boldsymbol{k})$ to be the additional increase in distortion due to losing frame $k_{N+1} > k_N$ given that frames $k_1, \ldots, k_N$ are already lost, i.e.,

$$D(k_{N+1}|\boldsymbol{k}) = D(k_1, \ldots, k_{N+1}) - D(k_1, \ldots, k_N). \tag{4}$$

The distortion chain model of order N uses the last N losses to predict the distortion for the current packet; hence, is comprised of the distortion quantities $D(\boldsymbol{k})$ for every loss pattern $\boldsymbol{k}$ of length N satisfying $k_i < k_j$, for $i < j$, and of $D(k_{N+1}|\boldsymbol{k})$ for every loss pattern $(\boldsymbol{k}, k_{N+1})$ of length $N+1$ satisfying $k_N < k_{N+1}$. These quantities can be generated at the encoder by simulating the corresponding loss events, decoding the video sequence, and then computing the resulting distortions. We next examine how $D(\boldsymbol{k})$ and $D(k_{N+1}|\boldsymbol{k})$ can be used to predict the total distortion for loss patterns of lengths greater than N.

Let DC^N denote our distortion chain of order N for a given video sequence. DC^N can be used to estimate the total distortion for an arbitrary packet loss pattern $\boldsymbol{k} = (k_1, \ldots, k_P)$ with P losses, where $N < P \leq L$ and L is once again the length of the video sequence in frames. Then, let $\widetilde{D}(\boldsymbol{k})$ denote the estimate of the total distortion due to the loss pattern $\boldsymbol{k}$ obtained from DC^N as follows:

$$\widetilde{D}(\boldsymbol{k}) = D(k_1, \ldots, k_N) + \sum_{i=N}^{P-1} D(k_{i+1}|(k_{i-N+1}, \ldots, k_i)). \tag{5}$$

This general formulation suggests that we need the distortions (conditional and unconditional) associated with any loss pattern of length N in order to predict the distortion for an arbitrary packet loss pattern of length $P > N$. While this may be impractical for large N, in our work we have found that even small values of N, still provide good prediction results. Moreover, when packet losses are spaced far apart (further than the intra refresh period of an encoded video sequence), they become decoupled since their effects are independent, as discussed earlier. This reduces the complexity of the algorithm associated with generating the distortion chain DC^N, as explained next.

For example, for the distortion chain DC^1 we need to store the distortion values $D(k)$ associated with losing frame $k = 1, \ldots, L$, illustrated in Fig. 1. In addition, we also need to store the quantities $D(j|k)$ from (4), which represent the additional increase in distortion when frame j is lost, given that frame k is already lost, for $1 \leq k < j \leq L$. Note that storing $D(j|k)$ is equivalent to storing $D(k, j)$ as apparent from (4), where $D(k, j)$ is the total distortion associated with losing frames k and j, illustrated in Fig. 1(b). Now, if $D(k, j)$ was stored for every possible pair (k, j), then the total storage cost would be quadratic in L since there are L isolated losses contributing to $D(k)$ and $L(L-1)/2$ distinct $D(k_1, k_2)$ values. However, since the distortion coupling between dropped packets decreases as the distance between the packets increases, one practical simplification is to assume $D(k, j) = D(k) + D(j)$ for $|j - k| > M + 1$, where M depends on the compression. For example, for a video encoding with a group of pictures (GOP) size of 15 frames, M is at most the number of packets in the GOP, i.e., 15. This approximation reduces the required storage and computation for DC^1 to being linear in L, precisely $(L - M -$

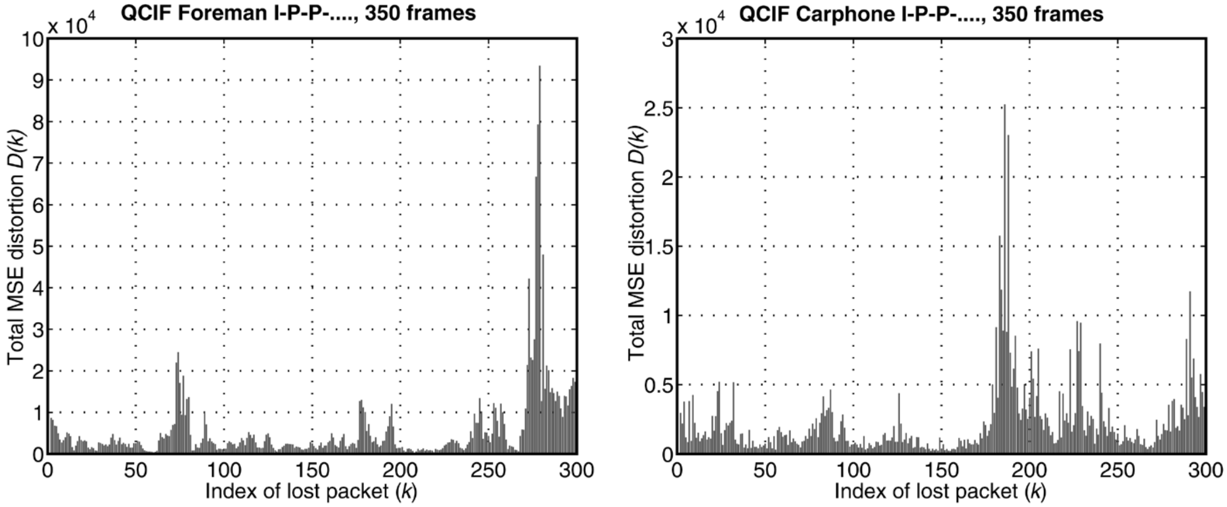


Fig. 2. Example of RDHT using DC^0 for Foreman and Carphone video sequences. Each sample point in the graphs identifies the total distortion $D(k)$ associated with the loss of a single frame k.

$1)(M+1)+M(M-1)/2$. Furthermore, the storage cost of a zeroth-order distortion chain DC^0 is even smaller. This model assumes no memory $(N = 0)$ and its storage cost is simply equal to L, since there are L possible single packet losses whose total distortion values $D(k)$ need to be recorded.

The accuracy of the distortion chain framework for predicting the distortion for arbitrary packet loss patterns is examined in Section V-A, where DC^0, DC^1, and DC^2 models are evaluated. The Section IV examines RDHT-based systems using DC^0 and DC^1 models, which provide a good tradeoff between storage and performance.

IV. Low-Complexity RaDiO Streaming Using RDHT

This section presents two low-complexity RDHT-based streaming techniques which employ two different distortion chain models. In particular, we discuss how these instantiations of RDHT-based streaming can address the two canonical media streaming problems under consideration for both types of transmission rate constraints: number of packets or number of bits. The performance of these techniques is evaluated in Section V.

A. Linear Size RDHT Using DC^0

For the first RDHT-based technique, we employ DC^0 to model the distortion associated with a packet loss pattern $\boldsymbol{k}$. This results in a linear storage cost of L numbers, as discussed in Section III-B, where L is the size of the video sequence in frames. Specifically, we simply store the total distortion in MSE $D(k)$ afflicting the video presentation caused by the loss of frame k, assuming no other frames have been lost. Fig. 2 illustrates an example RDHT using DC^0 for the video sequences Foreman and Carphone. Notice the huge variability in total distortion that results from losing different frames in the sequence. In Fig. 2 only the first 300 frames, out of a 350-frame video sequence, are considered as possible candidates to be lost in order to properly account for the error propagation effect.

This variability is quantified in Table I where we see that there exists significant variation in the total distortion produced by the loss of different P-frames.

TABLE I
Mean of the Total MSE Distortion $D(k)$ and Mean-Normalized Versions, Respectively, of the Minimum, Median, 95-Percentile, and Maximum Values of $D(k)$ for RDHT Using DC^0 for Different Sequences

Sequence	Mean	Min	Median	95%	Max
Foreman	5615.88	0.04	0.49	3.18	16.64
Mother & Daughter	247.77	0.06	0.61	3.71	6.92
Carphone	2254.80	0.10	0.60	3.33	11.19
Salesman	284.38	0.06	0.61	3.35	5.86

When N frames $\boldsymbol{k} = (k_1, k_2, \ldots k_N)$ are lost, the predicted total distortion obtained by DC^0 is simply given by

$$\widetilde{D}(\boldsymbol{k}) = \sum_{i=1}^{N} D(k_i). \qquad (6)$$

Recall 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 intrarefresh period, as recognized in [14]–[16]. Still, due to its simplicity and convenience for mathematical manipulations the additive model has found a number of applications in streaming and modeling of packetized media, such as [9], [10], and [17].

As mentioned earlier, we need to find the best transmission schedule for the packets of a video stream subject to a transmission data rate constraint. This problem can be formalized as follows. Let $\mathcal{W}$ be a window of packets considered for transmission and let R^* be the data rate constraint, measured either in bits or number of packets. We need to decide on the subset of packets $\boldsymbol{k} \in \mathcal{W}$ that should not be transmitted in order to satisfy the data rate constraint. Let $R(\mathcal{W}\backslash\boldsymbol{k})$ be the rate associated with the packets from $\mathcal{W}$ that will be transmitted, where "\" denotes the operator "set difference". Thus, we are interested in finding the subset $\boldsymbol{k}$ such that the total distortion due to dropping $\boldsymbol{k}$ is minimized, while meeting the data rate constraint, i.e.,

$$\boldsymbol{k}^* = \underset{\boldsymbol{k}\in\mathcal{W}:R(\mathcal{W}\backslash\boldsymbol{k})\leq R^*}{\arg\min} \widetilde{D}(\boldsymbol{k}). \qquad (7)$$

Now, consider first solving (7) in the case when the transmission data rate R^* is expressed in number of packets. Assume that $R^* = m$, i.e., we need to drop m packets from $\mathcal{W}$. Then $\boldsymbol{k}^*$ is easily found by sorting the distortions $D(j)$ for every packet $j \in \mathcal{W}$ in increasing order, and selecting the first m packets from the rank ordering (those with the m smallest associated distortions). In addition, if the problem changes to determine the best $m+1$ packets to drop, the solution then directly builds on the prior solution. Specifically, the selection of the best subset of m packets to drop is contained in the best subset of $m+1$ packets to drop. In contrast, an approach that does not provide this property would have to perform a completely new search for every m. The optimal schedule can, therefore, be obtained with very little computation.

Next, consider the case when R^* is measured in bits. This problem is more difficult than the packet-data rate constraint case. We denote $R(j)$ as the number of bits for packet j. We can compute an approximate solution by casting (7) as a nonconstrained optimization using a Lagrangian multiplier $(\lambda > 0)$

$$\boldsymbol{k}^* = \arg \min_{\boldsymbol{k} \in \mathcal{W}} \widetilde{D}(\boldsymbol{k}) + \lambda R(\mathcal{W} \setminus \boldsymbol{k}). \tag{8}$$

It can be shown that the solution to (8) reduces to dropping every single packet $j \in \mathcal{W}$ such that $\lambda_j = D(j)/R(j) \leq \lambda$, where λ_j can be thought of as the utility associated to packet j, measured in terms of distortion per bit. Therefore, once again $\boldsymbol{k}^*$ can be determined by sorting the packets in $\mathcal{W}$ in increasing order, but now based on their utility λ_j, and selecting to drop all the packets from the start of the rank ordering for which the above inequality is true. In this manner, once again we have an embedded search strategy with the associated low complexity benefits. Adjusting the Lagrange multiplier λ according to the rate constraint R^* is usually done in an iterative fashion using fast convex search techniques such as the bisection search technique [18].

B. A (Slightly Larger) Linear Size RDHT Using DC^1

Here, we employ DC^1 to model the distortion associated with a packet loss pattern $\boldsymbol{k}$. As described in Section III-B this still results in a linear (in L) storage and computation cost for DC^1, though somewhat larger than the corresponding one for DC^0. In particular, in addition to the total distortion $D(k)$ associated with all possible isolated losses k, we also need to store the conditional increase in distortion $D(j|k)$ associated with losing frame j given that frame k is lost, for $k < j$. Fig. 3 illustrates an RDHT example using DC^1.

Notice that there are some values of $D(j|k)$ which are negative. This is an interesting phenomenon and to the best of our knowledge has not been reported earlier in works on distortion modeling, except for the study of burst losses in [16] where negative correlation was identified to sometimes exist in neighboring lost frames. Having negative conditional distortions leads to the surprising result that sometimes it is better to drop two frames instead of dropping only one frame. For example, sometimes it is better to drop the two frames k and j together, instead of only dropping the single frame k, since the total distortion for dropping both frames k and j is less than that for dropping frame k only. Having this knowledge can be very useful for adaptive video streaming.

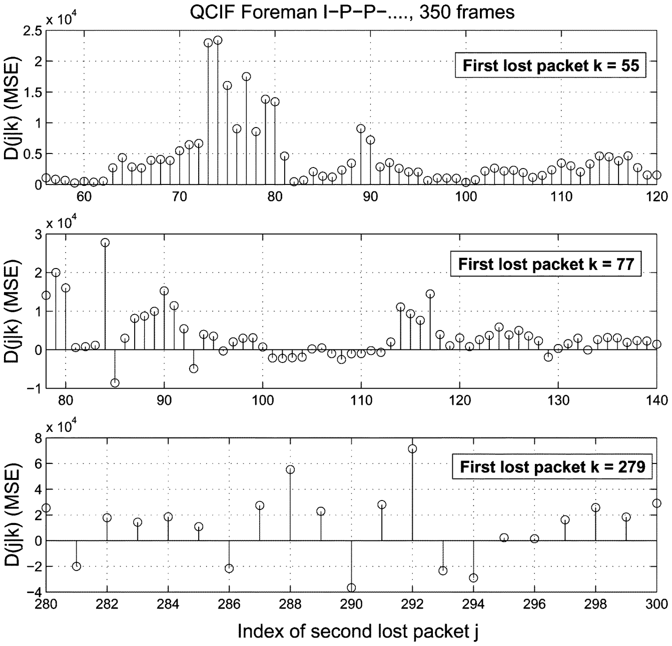


Fig. 3. Example of RDHT information using DC^1 for the Foreman video sequence. Each sample point in the graph is $D(j|k)$ which corresponds to the increase in total distortion due to the loss of frame j given that frame $k < j$ is already lost. Notice that there are a number of $D(j|k)$ which are negative, for example $D(290|279)$. In these cases, instead of only dropping one frame (e.g., 279), it is better to drop two frames (e.g., 279 and 290) since that will produce a smaller total distortion.

When N frames $\boldsymbol{k} = (k_1, k_2, \ldots k_N)$ are lost, the predicted total distortion obtained by DC^1 is given by

$$\widetilde{D}(\boldsymbol{k}) = D(k_1) + \sum_{i=1}^{N-1} D(k_{i+1}|k_i). \tag{9}$$

Searching for the optimal packet schedule [solving (7) exactly] in this scenario is computationally more expensive than in Section IV-A due to the interdependencies between the lost packets in $\boldsymbol{k}$ imposed by the underlying distortion model. Therefore, we employ an iterative descent algorithm in which we minimize the objective function $\widetilde{D}(\boldsymbol{k})$ one variable at a time while keeping the other variables constant, until convergence. In particular, consider first the case when R^* is expressed in number of packets and assume that $R^* = m$. Then, at iteration n, for $n = 1, 2, \ldots$, we compute the individual entries of the optimal drop pattern $\boldsymbol{k} = (k_1, \ldots, k_m)$ using

$$k_j^{(n)} = \arg \min_{k_j \in \mathcal{W}_j^{(n)}} \widetilde{D}(\boldsymbol{k}), \quad \text{for } j = 1, \ldots, m \tag{10}$$

where the sets $\mathcal{W}_j^{(n)} = \left\{ k_{j-1}^{(n)} + 1, \ldots, k_{j+1}^{(n-1)} - 1 \right\}$. Therefore, starting with a reasonable initial solution for $\boldsymbol{k}$, at each iteration we perturb the subset of selected packets $\boldsymbol{k}$ in order to find a subset that produces reduced distortion. At each iteration a subset with less or equal distortion is found, therefore, the algorithm is guaranteed to converge, though not necessarily to the global optimum.

We solve the case when R^* is measured in bits using Lagrangian optimization, as we did for RDHT based on DC^0. First, for a Lagrangian multiplier $\lambda > 0$, using the gradient descent algorithm from above we find the drop pattern $\boldsymbol{k} \in \mathcal{W}$ of length m that is the solution to (8). Then, we repeat this procedure for different values of m, e.g., $m = 1, \ldots, M$, where here M denotes the maximum number of packets considered for dropping from $\mathcal{W}$. Finally, we select the drop pattern $\boldsymbol{k}$ with the smallest minimum Lagrangian $J(\boldsymbol{k}) = \widetilde{D}(\boldsymbol{k}) + \lambda R(\mathcal{W} \setminus \boldsymbol{k})$ over all m.

C. Proposed Framework When Using B-Frames

We note that the proposed framework is generic and accounts straightforwardly for B-frames as well as frames coded with other more sophisticated types of prediction dependencies. This is because the proposed framework does not differentiate between frames based on their coding type, but rather based on their importance (in terms of the total distortion that would be produced if they were missing). Therefore, B-frames are handled in the same way as any other frame (e.g., I- or P)—they are treated based on their importance. Typically this results in B-frames being discarded before P-frames before I-frames, but this is not always the case. A B-frame can also be thought of as a P-frame with zero error propagation.

While the theoretical framework remains unchanged with the use of B-frames, the distortion chain implementation becomes simpler when B-frames are used since no frames depend on the B-frames and, therefore, there is no error propagation for their loss. While this is true for DC^0, it is of even more importance for higher order distortion chains, e.g., DC^1, since the use of B-frames requires fewer computations to examine the possible distortions that may occur and also less storage to store the associated results.

V. Experimental Results

In this section, we examine the performance of the proposed RDHT-based streaming technique for low-complexity RaDiO video streaming. The video sequences used in the experiments are coded using JM 2.1 of the JVT/H.264 video compression standard [19], using coding tools of the Main profile. Two standard test sequences in QCIF format are used, Foreman and Carphone. Each has at least 300 frames at 30 f/s, and is coded with a constant quantization level at an average luminance (Y) peak signal-to-noise ratio (PSNR) of about 36 dB. The first frame of each sequence is intracoded, 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 intraframe update period of $M = 4 \times 9 = 36$ frames. This section continues by examining the prediction accuracy of the distortion chain models. After that the performance of two RDHT-based streaming systems is evaluated.

A. Distortion Prediction

We study the performance of the distortion chains framework by simulating different packet loss patterns on the test video sequences. We compare the measured total distortion for each pattern with that predicted by distortion chain models of different order $N = 0, 1, 2$.

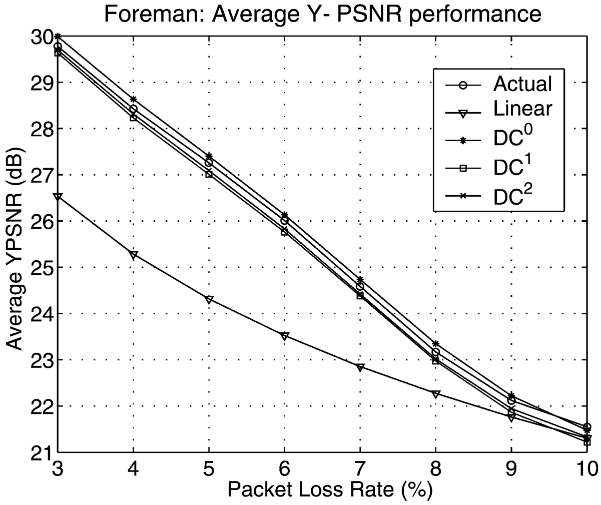


Fig. 4. PSNR of the actual and predicted total MSE distortions for *Foreman*.

In the first set of experiments, conducted using the Foreman sequence, the prediction performance is examined across the range of PLRs 3%–10%. Note that for lower PLRs the distortion chain framework can often perfectly predict the distortion since it can typically exactly account for the lost packets at the low PLRs. For each packet loss rate we generate a corresponding set of 50 000 random packet loss patterns. For each loss pattern $\boldsymbol{k} = (k_1, k_2, \ldots)$ we decode the video and record the resulting total MSE distortion $D(\boldsymbol{k})$ of the luminance component of the video. At the same time, we generate predictions of $D(\boldsymbol{k})$ using, respectively the Linear model [as defined in (2) in Section III] and the proposed distortion chains DC^0, DC^1, and DC^2. The predicted distortion values are denoted $\widetilde{D}(\boldsymbol{k})$ as in Section III-B. Finally, we compute the PSNR of these quantities using $10 \log_{10} \left(255^2/(D/N_F)\right)$, where D is either $D(\boldsymbol{k})$ or $\widetilde{D}(\boldsymbol{k})$ and N_F is the number of frames in the video sequence.

In Fig. 4, we show these PSNR values, averaged over all 50 000 loss patterns that correspond to a particular loss rate, as a function of the PLR. There are a few observations that follow from Fig. 4. First, all of the distortion chains provide better predictions of the expected distortion than the Linear model. Second, on average DC^1 and DC^2 underestimate the Y-PSNR as computed above, while DC^0 overestimates it, i.e., on average DC^1 and DC^2 overestimate the actual distortion, while DC^0 underestimates it.

Note that the performance difference between Linear and the distortion chain models is larger for low PLRs and it gradually decreases as the packet loss rate increases. Specifically, at $PLR = 3\%$ the distortion chain models provide a performance gain of roughly 3.5 dB, while at $PLR = 10\%$ the gain is practically negligible. In essence, this is due to the large variability in total distortion produced as a function of the specific packet which is lost (see Fig. 2). For example, let us assume that we lose only *one* packet in the sequence. Then, based on the specific lost packet l, for some l the total distortion will be much larger than the average single packet loss total distortion, $(1/L) \sum_{l=1}^{L} D(l)$, while for other l the total distortion will be much less than the average. Hence, our models allow us to explicitly capture the variability as a function of l, while

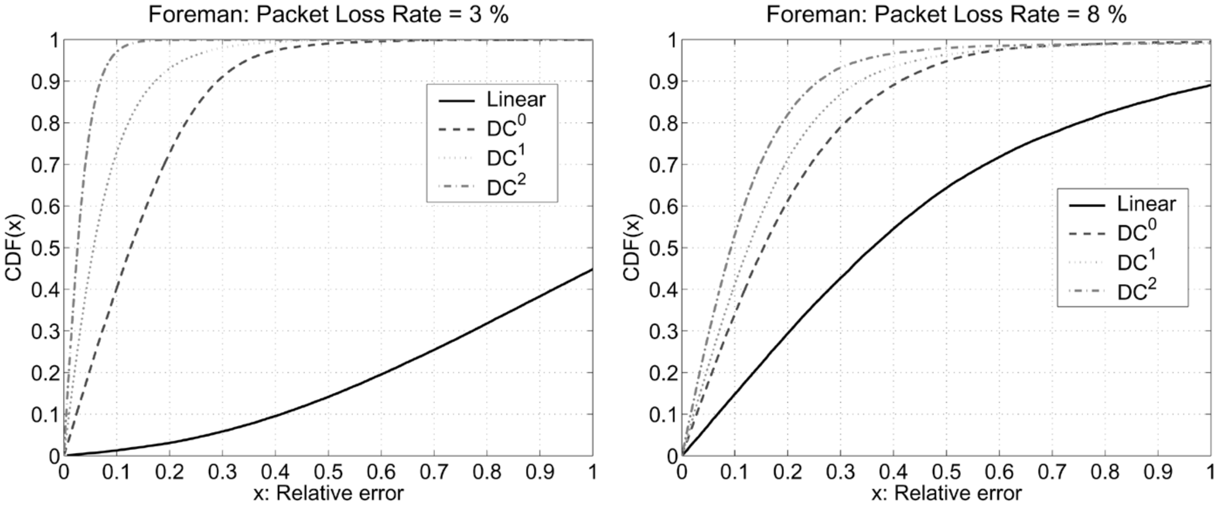


Fig. 5. The cdf of $\Delta D(\boldsymbol{k})$ for $PLR = 3\%$ (left) and $PLR = 8\%$ (right).

the Linear model does not provide that. On the other hand, as the number of losses increases (assuming for simplicity that the loss effects are independent) the resulting total distortion will approach $\#\text{Losses} \cdot (1/L)\sum_{l=1}^{L} D(l)$, since more averaging (over the lost packets) occurs and, therefore, the penalty that the Linear model pays decreases.

Next, we define $\Delta D(\boldsymbol{k}) = |D(\boldsymbol{k}) - \widetilde{D}(\boldsymbol{k})|/D(\boldsymbol{k})$ to be the relative error of a predicted distortion $\widetilde{D}(\boldsymbol{k})$ for a packet loss pattern $\boldsymbol{k}$. In essence, the relative error informs us how big the prediction error of $\widetilde{D}(\boldsymbol{k})$ is relative to the actual value $D(\boldsymbol{k})$ for a given loss pattern $\boldsymbol{k}$. We next examine the distribution of the relative error $\Delta D(\boldsymbol{k})$ over the 50 000 packet loss patterns $\boldsymbol{k}$ that correspond to a given PLR. Fig. 5 shows the cumulative density functions (cdfs) of the relative errors for all four distortion models considered here, for both PLR=3% and 8%. The first observation is that all of the distortion chain models perform significantly better than the linear model. In addition, for PLR=3% we see that DC^0, DC^1, and DC^2 provide estimates that are within a 10% error bound 40%, 75%, and 95% of the time, respectively, while the linear model achieves this less than 10% of the time. Similarly, DC^0, DC^1, and DC^2 provide estimates that are within a 20% error bound 74%, 93%, and 99% of the time, respectively, while the linear model does that only 5% of the time.

Fig. 5 (right) also shows that the distortion chain models provide improved accuracy as compared to the linear model at 8% PLR, though the improvement is lower due to the reduced accuracy as a result of the higher packet loss rate.

B. *Adaptive Streaming*

This section examines the end-to-end performance of the two RDHT-based streaming techniques proposed in Section IV. Performance is measured in terms of the average Y-PSNR in decibels of the decoded video frames at the receiver as a function of different channel parameters, namely, available transmission data rate and packet loss rate. Three scenarios are considered. In the first one, the network is lossless, but there is insufficient transmission data rate to send all video packets across the channel. Therefore, the sender needs to decide which packets to send and which packets to drop. In the second scenario, there is sufficient data rate to transmit every packet of the video once, however the network is lossy and some of the transmitted packets are lost. Hence, the sender needs to decide at each transmission opportunity whether (1) to retransmit a previous lost packet, or (2) to transmit a new packet which has not been transmitted before. Finally, the third scenario under consideration represents a combination of the first two with the addition that transmitted packets here that are not lost experience a random delay in the network. Specifically, in this scenario we examine streaming performance when simultaneously the transmission data rate can be variable and the network exhibits random packet loss and delay.

In addition to the RDHT-based systems using DC^0 and DC^1, we also study the performance of a streaming system referred to as *Oblivious* since it does not consider the distortion that results from dropping a frame. In particular, when making transmission decisions, *Oblivious* does not distinguish between two packets that contain two different P-frames, except for the size of the packets. *Oblivious* randomly chooses between two P-frame packets of the same size, for example, when it needs to reduce the number of transmitted packets. Similarly, transmissions of new packets and retransmissions of old lost packets are also performed in a random order. In all three systems, packets are considered for transmission in nonoverlapping windows of size $W = 100$. That is, at every transmission instance the sender considers 100 new packets for transmission. No retransmissions occur after the packets from the last transmission window are sent.

Adapting to Available Data Rate: Fig. 6 shows the performances of RDHT using DC^0, RDHT using DC^1, and *Oblivious* for streaming Foreman and Carphone as a function of the available packet rate measured in percent. For example, packet rate of 99% means that 99% of the packets in a transmission window can be transmitted. It can be seen that both RDHT using DC^0 and RDHT using DC^1 outperform *Oblivious* with quite a significant margin over the whole range of values considered for the available packet rate. This is due to the fact that RDHTs DC^0 and DC^1 exploit the knowledge about the effect of loss of individual video packets on the reconstructed video quality.

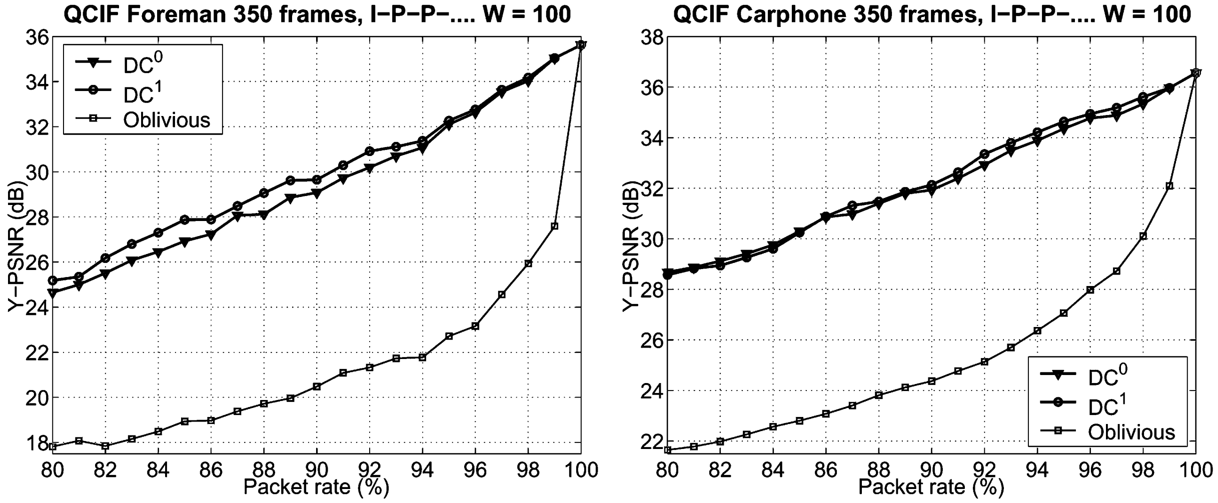


Fig. 6. Y-PSNR (dB) versus packet rate (%).

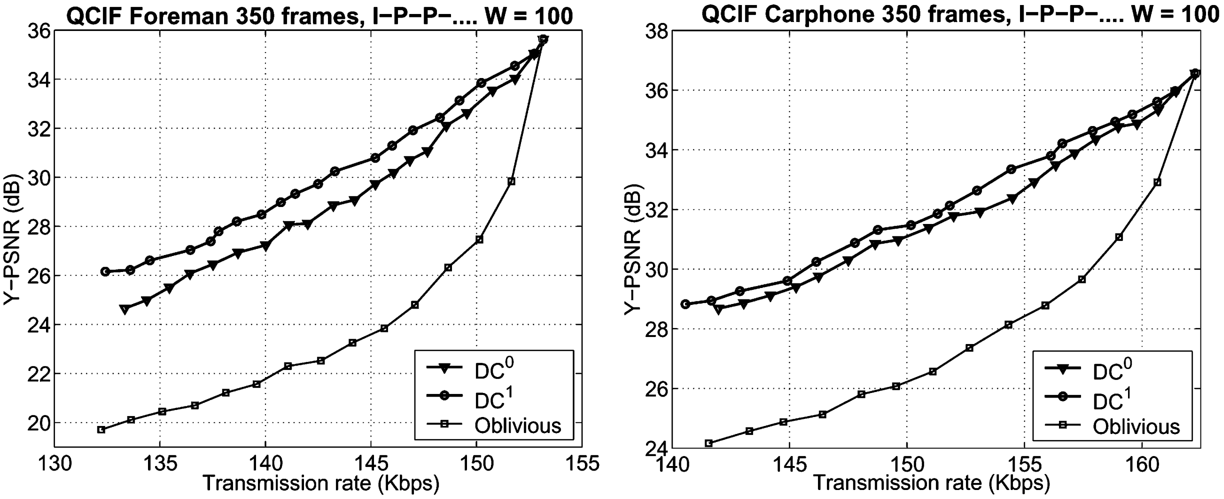


Fig. 7. Y-PSNR (dB) versus transmission rate (kbps).

Therefore, both RDHT DC^0 and RDHT DC^1 drop the video packets that will have the least impact on the quality of the reconstructed video. As seen in Fig. 6 the performance gains reach up to 8 dB for Foreman and 7 dB for Carphone for packet rates of 86–96%. In addition, even outside this range the gains in performance are still impressive and do not drop below 5 dB, except of course when we can send all the packets. Finally, note that in this scenario the difference in performance between RDHT DC^0 and RDHT DC^1 is quite small.

Fig. 7 examines the performances of RDHT DC^0, RDHT DC^1 and *Oblivious* for streaming when the transmission constraint is in kilobits per second (kbps), rather than packets as in the prior experiment. Again, both RDHT DC^0 and RDHT DC^1 provide substantial performance gains over *Oblivious* over the whole range of available transmission rates. The gains in performance remain steadily around 5–6 dB almost over the whole range of transmission rates under consideration, for both Foreman and Carphone. Note that in this case the performance difference between the low-complexity RDHT techniques and *Oblivious* is not as large as in the previous case. Having a transmission constraint expressed in bits makes predicting the resulting distortion at the receiver due to a packet drop pattern more difficult for the distortion chain based systems, as the number of dropped packets may need to vary over different transmission windows. Finally, we observe that the performance difference between the RDHT DC^0 and RDHT DC^1 is somewhat larger in this case.

Adapting to Packet Loss: Reactive Adaptation: The performance of the three streaming systems is now examined in the second scenario where we have packet loss. In contrast to the first scenario, here there is sufficient channel data rate to transmit once every packet of the video. However, there is random packet loss on the forward channel and the sender needs to decide whether it should retransmit previous lost packets or instead transmit new packets which have not been transmitted yet. In other words, in addition to the W packets from the current transmission window, the sender also considers for the present transmission past packets from previous transmission windows that have been lost during transmission. Note that this leads to a slight increase in complexity as now the number of packets to be considered for transmission at each transmission opportunity increases from W to W plus the number of previous lost packets. These experiments assume the following: 1) the forward channel exhibits no packet delay, but only loss;

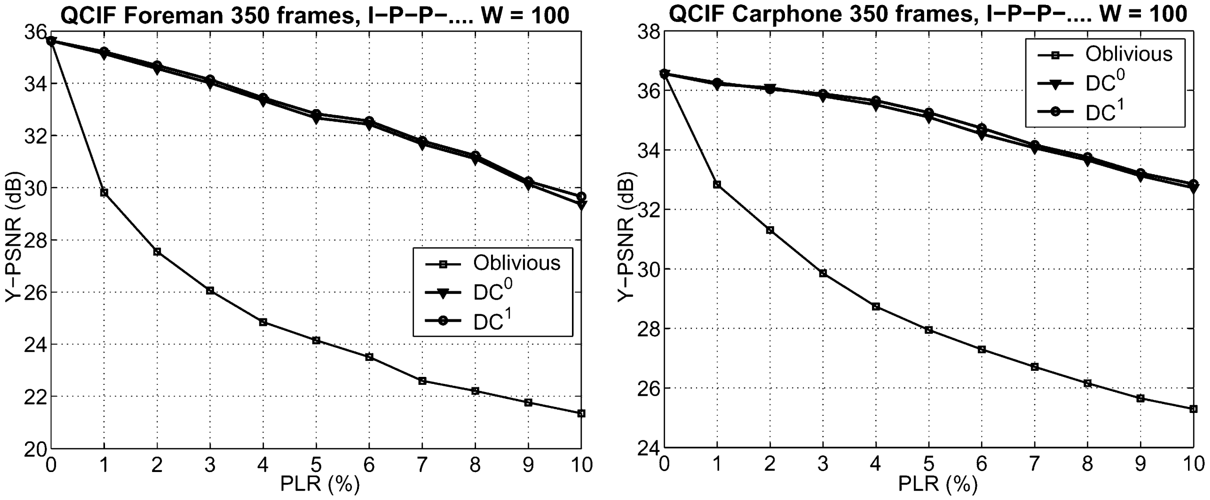


Fig. 8. Y-PSNR (dB) versus PLR (%).

2) the sender is immediately notified of each lost packet (an ideal feedback channel); and 3) successive packet losses are independent and identically distributed.

It can be seen from Fig. 8 that the performances of the three streaming systems for this scenario are equivalent to those shown in Fig. 6, which is expected. Therefore, we do not discuss these results in detail. Instead, we just note that as in the scenario associated with Fig. 6, also here the two RDHT-based systems provide substantial performance improvement over *Oblivious*.

Adapting to Packet Loss: Active Adaptation: This section investigates the end-to-end performance for the scenario where the available transmission rate can be varied and the network exhibits random packet loss and delay on both forward and backward channels. Four standard test video sequences in QCIF format are used in these experiments: Foreman, Mother and Daughter (MthrDhtr), Carphone, and Coastguard. Two sets of experiments are performed. In the first set of experiments, each sequence is coded at 10 f/s, resulting in 130 coded frames, with a constant quantization level for an average Y-PSNR of about 36 dB, and a GOP size of 20 frames, where each GOP consists of an I-frame followed by 19 consecutive P-frames. The second set of experiments are similiar to the first, however B-frames are used with a GOP structure of IBBBP (three B-frames between each pair of reference frames). Three sender-driven streaming systems are employed in the experiments. *RDHT* and *Oblivious* are streaming systems that were introduced earlier. Specifically, *RDHT* is the RDHT-based system using DC^0. *Conv. RaDiO* is a streaming system that employs a conventional RaDiO technique for packet scheduling such as the one from [5]. The Lagrange multiplier λ is fixed for the entire presentation according to the available transmission rate for the two RaDiO systems, i.e., *Conv. RaDiO* and *RDHT*. The playout delay is 600 ms, and the time interval between transmission opportunities $T = 100$ms.

In all three systems, packets are considered for transmission in overlapping windows of variable size as in [4]. In all three systems, for every arriving packet on the forward channel the receiver returns immediately to the sender an acknowledgment packet on the backward channel. At each transmission opportunity *RDHT* and *Oblivious* consider for retransmission only those packets from the transmission window whose last transmission has not been acknowledged within $\mu_R + 3\sigma_R$ seconds from the current transmission opportunity, where μ_R and σ_R are, respectively, the mean and the standard deviation of the round-trip time. This time-out value is frequently used in ARQ systems, e.g., TCP [20].

The forward and the backward channel on the network path between the sender and the receiver are modeled 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 the forward and backward delay densities p_F and p_B are modeled as 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.

For comparison purposes, in the following figures we also show the performance of an "ideal" R-D optimal sender-driven system denoted as "RD bound". Specifically, the performance of "RD bound" is computed using the R-D characteristics of the video sequence and the characteristics of the channel in the following manner. The communication channel between the sender and the receiver acts as a packet erasure channel with a drop probability of ϵ_F. Then, if the sender transmits at a data rate R_s, the data rate observed at the receiver is $R_r = (1 - \epsilon_F)R_s$ (assuming independence between packet losses and packet size). Then, for every data rate R_s at which a sender can transmit the distortion performance of "RD bound" is computed as the smallest possible distortion for R_r using an optimal pruning algorithm [18] and the R-D characteristics of the video sequence.

We first examine the performance of the three systems for streaming Foreman and Coastguard using only I and P-frames in Fig. 9. For Foreman, *Conv. RaDiO* outperforms *RDHT* over the whole range of transmission rates, with a margin of roughly 1–2 dB at the high end of transmission rates and increasing to 5–6 dB at the low end of transmission rates. However for Coastguard the performance is much closer with less than 0.5 dB difference in performance at the high end of the transmission rate. *RDHT* outperforms *Oblivious* for Foreman with a significant margin over the whole range of transmission rates, with a

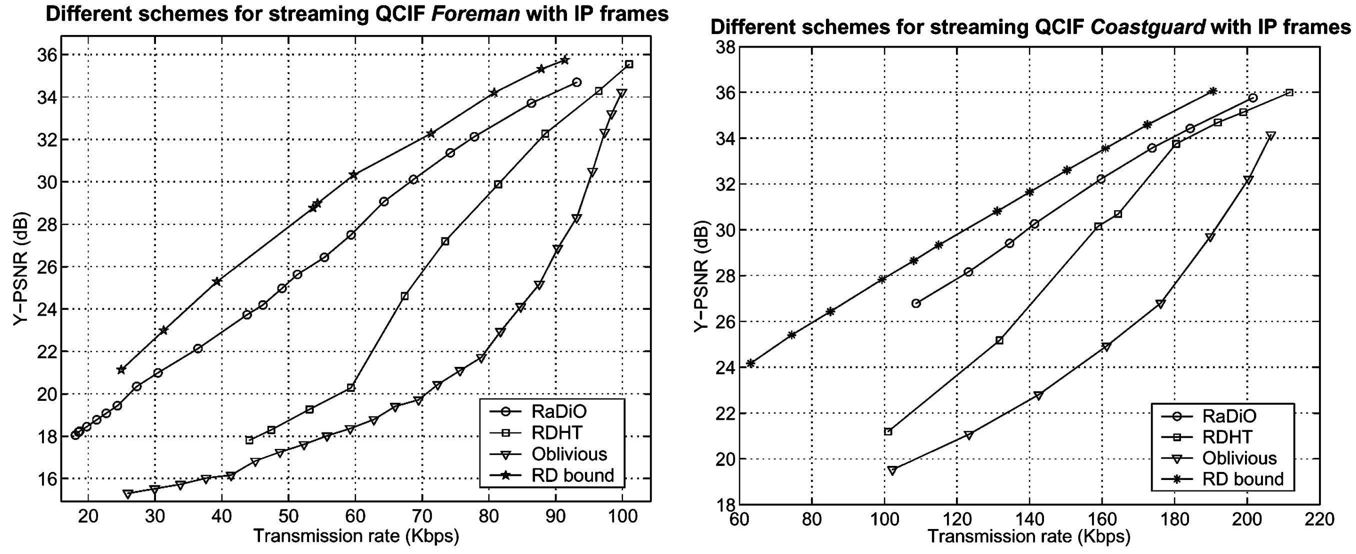


Fig. 9. R-D performance for streaming Foreman and Coastguard (coded using I- and P-frames).

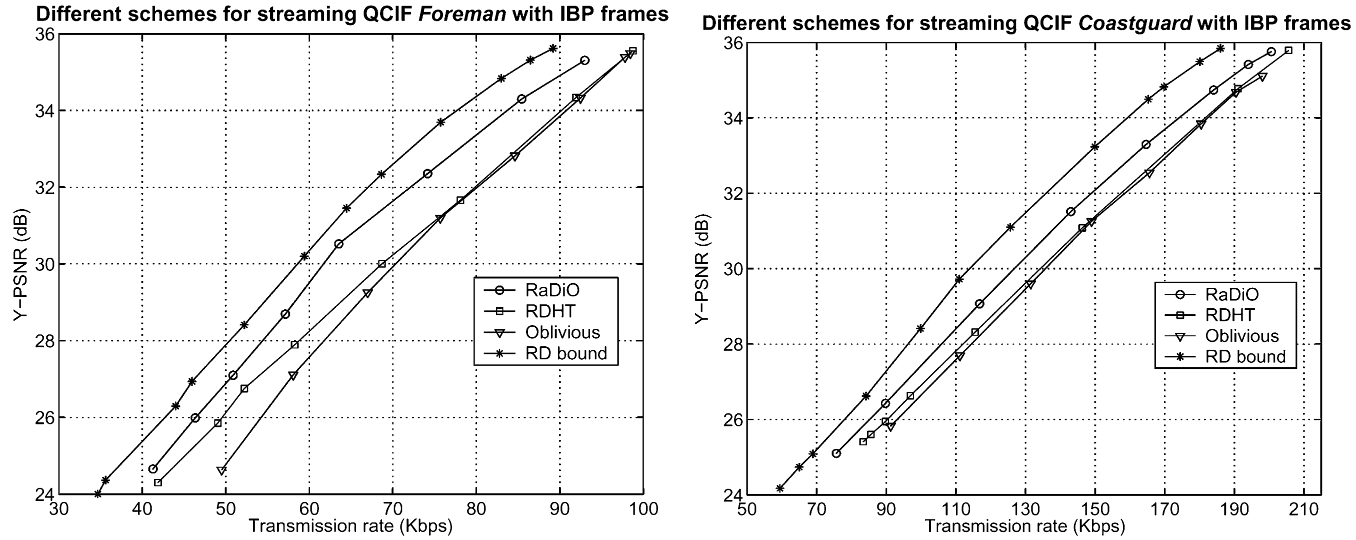


Fig. 10. R-D performance for streaming Foreman and Coastguard (coded using I-, P-, and B-frames).

gain of at least 6 dB for rates of 65–90 kb/s, and a maximum gain of about 8 dB at 80 kb/s. *RDHT* also provides similar gains over *Oblivious* for the Coastguard sequence. Finally, the performance loss of *Conv. RaDiO* with respect to "RD bound" is on the order of 1–2 dB and is due to the late loss, i.e., packets arriving at the receiver after their delivery deadline. Note that similar performance characteristics are also achieved for MthrDhtr and Carphone sequences, however these results are not shown because of the limited space.

We next examine the performance of the three systems for streaming Foreman and Coastguard using I, P, and B-frames in Fig. 10, where we have three B-frames between each pair of reference frames. In these experiments *Oblivious* discards B-frames before P-frames before I-frames. This simple heuristic is a natural approach to exploit the different importance of I, P, and B-frames and it provides significant gain. Furthermore, when B-frames are used the performance curves for all three techniques (*Conv. RaDiO*, *RDHT*, and *Oblivious*) are much closer together. Specificially, *Conv. RaDiO* provides only about 1.0-dB gain over *RDHT* which provides only about 0.25-dB gain over *Oblivious*. Therefore, the performance benefit provided by either RaDiO technique (*Conv. RaDiO* or *RDHT*) significantly decreases when B-frames are used. An interesting note is that *RDHT* occasionally decides to drop a P-frame instead of dropping a B-frame—a decision which is never made by the *Oblivious* technique. To summarize the results from the experiments using I, P, and B-frames, the proposed approach does provide a gain in performance when B-frames are used, however the performance gain is much lower than in the case when only I and P-frames are used. Specifically, the heuristic of dropping B-frames before dropping P-frames generally works quite well.

Several important observations follow from these experiments. *Conv. RaDiO* outperforms the other two streaming systems with a margin that is usually substantial. This is expected, since *Conv. RaDiO* assumes accurate statistical knowledge of the channel and employs an optimization framework for computing its transmission schedules that is far more sophisticated and accurate than the streaming techniques employed by the other two systems. For example, *Conv. RaDiO* uses models for the forward and the backward channel and given these models it computes the optimal transmission schedule that minimizes the expected distortion. However, this optimization procedure requires a much higher computational complexity which presently is unacceptable for online streaming. Furthermore, the performance of *Conv. RaDiO* is dependent on accurate and timely knowledge of the state of the forward and backward channels, which sometimes may be difficult to achieve in today's highly dynamic networks. Therefore, it is encouraging to see that *RDHT* provides a significant fraction of the performance provided by *Conv. RaDiO* while requiring significantly less complexity and without requiring channel knowledge. Moreover, the appeal of *RDHT* becomes even stronger when we note the substantial performance gains, reaching up to 8 dB, that it offers over systems such as *Oblivious*, which can be thought of as a representative example of streaming systems used in practice today. In particular, *RDHT* provides this significant performance gain with a complexity that is of the same order as that of *Oblivious*. Note that the performance improvement of *RDHT* over *Oblivious* is only minor for the case of streaming with I, P, and B-frames, as seen in the last set of results. However, it should be mentioned that in this scenario the *Oblivious* system is not so "oblivious" to the importance of the packets' content, as it performs prioritized (re)transmissions based on the associated video frame type of the packets. This in essence contributes to having both systems, *RDHT* and *Oblivious*, performing the same transmission decisions almost all of the time, which in turn means that also the latter system is "almost" RaDiO. In addition, we expect that the performance difference between *RDHT* and *Oblivious* will increase and will become more similar to what we observed in the earlier results as a larger fraction of coded frames are used as references for predicting other frames, i.e., the proportion of non-B-frames is increased.

This section concludes by briefly describing the computational requirements of *Conv. RaDiO* and *RDHT* for steady-state operation, and specifically provides upper bounds on the number of operations per video packet. The complexity of *Conv. RaDiO* is on the order of $N_i|\mathcal{W}|\left(C|\mathcal{N}_c| + 2^{NM}\right)$, where N_i is the number of iterations that the optimization algorithm [4], [5] performs until convergence (typically on the order of 2–3) and $|\mathcal{W}|$ is the size of the transmission window $\mathcal{W}$ during which a data unit is considered for transmission. The multiplicative factor $N_i|\mathcal{W}|$ is needed because the optimal schedule for a data unit is computed at every iteration of the optimization algorithm and as long as the data unit is in the transmission window $\mathcal{W}$. However, this is an upper bound as once the reception of the data unit is acknowledged, there is no further computing cost associated with it although the data unit may continue to be present in the transmission window $\mathcal{W}$. In addition, $|\mathcal{N}_c|$ is the size of the concealment set for a data unit and it signifies the number of concealment events that are considered when the optimal schedule for a data unit is computed. C is a constant that can be very large and that depends on the sizes of the ancestor and descendant sets for the data units that are involved in the concealment events in $\mathcal{N}_c$. Note that the ancestor set for data unit l represents the set of data units that must be received on time in order to decode that data unit; similarly, the descendant set for data unit l represents the set of data units for which data unit l must be received on time in order to decode them (further details in [4], [5]). Finally, N is the number of transmission opportunities over which the optimal schedule is computed for a data unit. The complexity of *RDHT* is substantially smaller as this technique requires at most $|\mathcal{W}|$ computing operations (on average $(1/2)|\mathcal{W}|$) to find the appropriate location for a video packet j (based on its utility per bit λ_j) in the sorted list of packets that are already in the transmission window $\mathcal{W}$. Note that this operation is performed once per packet, when it first enters the transmission window.

VI. Conclusion

This paper proposed RDHT-based systems for performing low-complexity RaDiO adaptive streaming. Specifically, the RDHT-based streaming systems enable low-complexity online adaptive algorithms which adapt to the available data rate, packet loss, and the R-D characteristics of the video source. Experimental results demonstrate that for the difficult case of nonscalably coded H.264 video (one I-frame followed by all P-frames), the proposed RaDiO adaptive system provides several decibel gain over the conventional non-RD-optimized oblivious streaming system. Furthermore, the proposed system provides a significant fraction of the gain provided by the high-complexity RaDiO system, without requiring neither 1) accurate information about the forward and the backward channel statistical characteristics, nor 2) the computationally expensive optimization, that conventional RaDiO requires. We believe that these properties make the proposed RDHT-based systems quite promising for practical, high-quality adaptive streaming over real-world networks, and in particular can be implemented within the context of the popular MPEG-4 File Format (MP4).

References

[1] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[2] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[3] W.-T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. Circuits Sys. Video Technol.*, vol. 11, no. 3, pp. 373–387, Mar. 2001.

[4] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Microsoft Research, Beijing, China, Tech. Rep. TR-2001-35, 2001.

[5] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. IEEE DCC*, Mar. 2003, pp. 203–212.

[6] *ISO Media File Format Specification*, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1, 2001.

[7] S. J. Wee and J. G. Apostolopoulos, "Secure scalable video streaming for wireless networks," in *Proc. IEEE ICASSP*, May 2001, pp. 2049–2052.

[8] Z. Miao and A. Ortega, "Fast adaptive media scheduling based on expected run-time distortion," in *Proc. Asilomar Conf.*, Nov. 2002, pp. 1305–1309.

[9] 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. IEEE ICASSP*, May 2004, pp. 1001–1004.

[10] 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. IEEE ICME*, Jun. 2004, pp. 1387–1390.

[11] J. Chakareski, J. Apostolopoulos, and B. Girod, "Low-complexity rate-distortion optimized video streaming," in *Proc. IEEE ICIP*, Oct. 2004, pp. 2055–2058.

[12] I.-M. Kim and H.-M. Kim, "A new resource allocation scheme based on a PSNR criterion for wireless video transmission to stationary receivers over gaussian channels," *IEEE Trans. Wireless Commun.*, pp. 393–401, Jul. 2002.

[13] A. Reibman and V. Vaishampayan, "Quality monitoring for compressed video subjected to packet loss," in *Proc. IEEE ICME*, Jul. 2003, pp. 17–20.

[14] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. SPIE VCIP*, Jan. 2001, pp. 329–409.

[15] J. Apostolopoulos, W.-T. Tan, S. Wee, and G. Wornell, "Modeling path diversity for multiple description video communication," in *Proc. IEEE ICASSP*, May 2002, pp. 2161–2164.

[16] Y. Liang, J. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst-length matter," in *Proc. IEEE ICASSP*, Apr. 2003, pp. 684–687.

[17] E. Masala and J. de Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. IEEE ICME*, Jul. 2003, pp. 345–348.

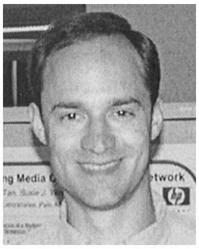
[18] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 1, pp. 1445–1453, Sep. 1988.

[19] *Video Coding for Low Bitrate Communication*, Draft ITU-T Recommendation H.264, 2003.

[20] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocolsx*. Reading, MA: Addison-Wesley, 1994.

Jacob Chakareski received the B.S. degree from Ss. Cyril and Methodius University, Skopje, Macedonia, in 1996 and the M.S. degree from Worcester Polytechnic Institute, Worcester, MA, in 1999, both in electrical engineering. He was a Ph.D. student in electrical engineering at Rice University, Houston, TX, from 1999 to 2002 and at Stanford University, Stanford, CA, from 2002 to 2005.

He is a Postdoctoral Researcher in the LTS4 group at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. His fields of interest are multimedia networking, statistical signal processing, and communication theory.



John G. Apostolopoulos (S'91–M'97) received his B.S., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge.

He joined Hewlett-Packard Laboratories, Palo Alto, CA, in 1997, where he is currently a Principal Research Scientist and Project Manager for the Streaming Media Systems Group. He also teaches at Stanford University, Stanford, CA, where he is a Consulting Assistant Professor of electrical engineering. He contributed to the U.S. Digital Television and JPEG-2000 Security (JPSEC) standards. His research interests include improving the reliability, fidelity, scalability, and security of media communication over wired and wireless packet networks.

Dr. Apostolopoulos has served as an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and of IEEE SIGNAL PROCESSING LETTERS, and as a member of the IEEE Image and Multidimensional Digital Signal Processing (IMDSP) technical committee. He received a best student paper award for part of his Ph.D. dissertation, the Young Investigator Award (best paper award) at VCIP 2001 for his paper on multiple description video coding and path diversity for reliable video communication over lossy packet networks, and in 2003 was named "one of the world's top 100 young (under 35) innovators in science and technology" (TR100) by Technology Review.

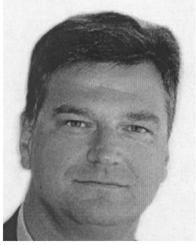
Susie Wee (M'96) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, where she studied scalable video coding and video communications over wireless broadcast channels.

She is the director of the Multimedia Communications and Networking Department at Hewlett Packard Laboratories, which includes research in streaming media systems, networking overlays, video, audio, computer vision and graphics, and multimedia security.She is a Consulting Assistant Professor at Stanford University, Stanford, CA. Her research interests broadly embrace multimedia networking and secure streaming. She is also the co-editor of the JPEG-2000 Security standard (JPSEC). Susie has 10 granted patents, over 30 pending patents, and over 40 international publications

Dr. Wee received Technology Review's Top 100 Young Investigators award in 2002. She is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Wai-tian (Dan) Tan (M'01) received the B.S. degree from Brown University, Providence, RI, in 1992, the M.S.E.E. degree from Stanford University, Stanford, CA, in 1993, and the Ph.D. degree from the University of California, Berkeley, in 2000.

He joined HP Laboratories, Palo Alto, CA, in December 2000 and is a Member of the Media Communications and Networking Department. He worked for Oracle Corporation, Redwood Shores, CA, from 1993 to 1995. His research focuses on adaptive media streaming, both at the end-points and inside the delivery infrastructure.



Bernd Girod (S'80–M'80–SM'97–F'98) received the M. S. degree in electrical engineering from Georgia Institute of Technology, Atlanta, in 1980 and the Ph.D. degree (with highest honors) from the University of Hannover, Hannover, Germany, in 1987.

He is presently a Professor of Electrical Engineering with the Information Systems Laboratory, Stanford University, Stanford, CA. He also holds a courtesy appointment with the Stanford Department of Computer Science and he serves as Director of the Image Systems Engineering Program at Stanford. His research interests include networked media systems, video signal compression and coding, and three-dimensional image analysis and synthesis. Until 1987, he was a Member of the Research Staff with the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, working on moving image coding, human visual perception, and information theory. In 1988, he joined Massachusetts Institute of Technology, Cambridge, first as a Visiting Scientist with the Research Laboratory of Electronics and then as an Assistant Professor of Media Technology at the Media Laboratory. From 1990 to 1993, he was a Professor of Computer Graphics and Technical Director of the Academy of Media Arts, Cologne, Germany, jointly appointed with the Computer Science Section of Cologne University. He was a Visiting Adjunct Professor with the Digital Signal Processing Group, Georgia Institute of Technology, Atlanta, in 1993. From 1993 until 1999, he was the Chaired Professor of Electrical Engineering/Telecommunications, University of Erlangen-Nuremberg, Nuremberg, Germany, and the Head of the Telecommunications Institute I, codirecting the Telecommunications Laboratory. He has served as the Chairman of the Electrical Engineering Department from 1995 to 1997 and as Director of the Center of Excellence "3-D Image Analysis and Synthesis" from 1995 to 1999. He was a Visiting Professor with the Information Systems Laboratory of Stanford University during the 1997–1998 academic year. As an entrepreneur, he has worked successfully with several start-up ventures as founder, investor, director, or advisor. Most notably, he has been a cofounder and Chief Scientist of Vivo Software, Inc., Waltham, MA (1993–1998); after Vivo's aquisition, Chief Scientist of RealNetworks, Inc. (1998–2002), and an outside Director of 8 × 8, Inc. (1996–2004). He has authored or coauthored one major textbook, two monographs, and over 250 book chapters, journal articles, and conference papers in his field, and he holds about 20 international patents.

Prof. Girod has been a member of the IEEE Image and Multidimensional Signal Processing Committee from 1989 to 1997. He was named "Distinguished Lecturer" in 2002 by the IEEE Signal Processing Society. Together with J. Eggers, he was the recipient of the 2002 EURASIP Best Paper Award. He has served on the Editorial Boards or as an Associate Editor for several journals in his field and is currently Area Editor for Speech, Image, Video Signal Processing of the IEEE TRANSACTIONS ON COMMUNICATIONS. He has served on numerous conference committees, e.g., as Tutorial Chair of ICASSP-97 in Munich, Germany, and ICIP-2000 in Vancouver, ON, Canada, as General Chair of the 1998 IEEE Image and Multidimensional Signal Processing Workshop in Alpbach, Austria, and as General Chair of the Visual Communication and Image Processing Conference (VCIP) in San Jose, CA, in 2001.