Layered Coding vs. Multiple Descriptions for Video Streaming over Multiple Paths

J. Chakareski, S. Han, and B. Girod
Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford CA, 94305
{cakarz,sehan,bgirod}@stanford.edu

ABSTRACT

In this paper, we examine the performance of specific implementations of multiple description coding and of layered coding for video streaming over error-prone packet switched networks. We compare their performance using different transmission schemes with and without network path diversity. It is shown that given the specific implementations there is a large variation in relative performance between multiple description coding and layered coding depending on the employed transmission scheme. For scenarios where the packet transmission schedules can be optimized in a rate-distortion sense, layered coding provides a better performance. The converse is true for scenarios where the packet schedules are not rate-distortion optimized.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Compression (Coding); C.2.1 [**Network Protocols**]: Network Communications

General Terms

Algorithms, Performance, Experimentation

Keywords

Layered coding, multiple description coding, rate-distortion optimized streaming, Automatic Repeat reQuest (ARQ), packet path diversity.

1. INTRODUCTION

Since the introduction of the first commercial products in 1995 and its subsequent phenomenal growth, Internet video streaming has placed new demands on source coding and network transport algorithms. The challenge is to deliver video over the Internet, a channel with widely varying packet delay, loss, and throughput, in a way that simultaneously maximizes the display quality at the receiver, meets bit-rate limitations, and satisfies latency constraints. All systems, therefore, require efficient compression, some form of rate scalability, and error-resiliency techniques.

Rate scalability can be elegantly achieved by *scalable video* codecs [1, 2] that provide layered embedded bit-streams that are decodable at different bitrates, with gracefully degrading quality. Layered representations for Internet streaming have been widely studied, e.g., in [3, 4, 5]. In addition, scalable representations have become part of established video coding standards, such as MPEG and H.263+ [6, 7]. Scalable video representations aid in TCP-friendly streaming, as they provide a convenient way for performing the rate control required to mitigate network congestion, see, e.g., [8, 9, 10]. In receiver-driven layered multicasting, video layers are sent in different multicast groups, and rate control is performed individually by each receiver by subscribing to the appropriate groups [11, 12, 13]. Layered video representations have further been proposed in combination with differentiated quality of service (DiffServ) [14] in the Internet, e.g., [15, 16, 17, 18]. The idea is to transmit the more important layers with better, but more expensive, quality of service (QoS), and the less important layers with fewer or no QoS guarantees.

A scalable representation of video signals consists of a base layer and multiple enhancement layers. The base layer provides a basic level of quality and can be decoded independently of the enhancement layers. On the other hand, the enhancement layers serve only to refine the base layer quality and alone are not useful. Therefore, the base layer represents the most critical part of the scalable representation, which makes the performance of streaming applications that employ layered representations sensitive to losses of base layer packets.

Multiple description coding has been proposed as an alternative to layered coding for streaming over unreliable channels [19, 20, 21, 22]. Each description alone can guarantee a basic level of reconstruction quality of the source, and every additional description can further improve that quality. There have been a few performance comparisons between LC and MDC reported in the literature. In [23], the authors examine the performance of LC and MDC for transmission over binary symmetric channels and over erasure channels using Forward Error Correction (FEC) codes. In [24], the authors compare LC and MDC through experiments using network simulations. Furthermore, the work in [25] studies the performance of LC and MDC for streaming over a single wireless channel using transmission prioritization, while the work in [26] considers transmission of LC and MDC over

MM'03, November 2–8, 2003, Berkeley, California, USA.

multiple network paths. Finally, the authors in [27] present a comprehensive performance study of LC and MDC for video streaming over error-prone networks that confirmed most of the results reported earlier. In addition, in [28, 29, 30] combining LC and MDC is considered in order to exploit the individual benefits and at the same time to avoid the individual shortcomings of these source coding techniques.

In this paper, we study the performance of specific implementations of LC and of MDC as two viable source coding techniques for video streaming over the Internet. Their performance is compared in different transmission scenarios depending on the employed transmission scheme. The possibility of transmission over multiple network paths is also considered. One of the schemes performs Rate-Distortion Optimized (RaDiO) scheduling of the packet transmissions over the available network paths. The others are based on heuristics and do not optimize their transmission schedules in a rate-distortion sense.

Historically, packet path diversity for video streaming has been introduced in [31], where the author proposes to send complementary descriptions of a multiple description (MD) coder through two different Internet paths. The presented experimental results showed the potential benefits of the proposed system. Since then a number of studies have appeared that exploit the concept of packet path diversity in media communication. In [32] the authors employ path diversity in the context of video communication using unbalanced MD coding to accommodate the fact that different paths might have different bandwidth constraints. The unbalanced descriptions are created by adjusting the frame rate of a description sent over a particular path. In [33] the authors study image and video transmission in mobile radio networks. It is shown that combining MD coding and multiple path transport in such a setting provides higher bandwidth and robustness to end-to-end connections. In [34] a framework for video transmission over the Internet is presented, based on path diversity and rate-distortion optimized reference picture selection. Here, based on feedback, packet dependency is adapted to channel conditions in order to minimize the distortion at the receiving end, while taking advantage of path diversity. In [35] the performance of path diversity and multiple description coding in Content Delivery Networks (CDN) is studied. 20-40% reduction in distortion is reported over conventional CDNs for the network conditions and topologies under consideration. Finally, another related work is [36] where the authors consider rate-distortion optimized streaming over networks with DiffServ support.

This paper is organized as follows. We first present the transmission scheme for RaDiO packet scheduling and path selection in Section 2. Then, we employ this scheme in the first part of Section 3 to compare through experiments the performance of MDC and LC for streaming different packetized video contents over multiple network paths. The specific implementations of MDC and LC employed in the experiments are discussed on the beginning of this section. Next, in the second part of Section 3, we again compare through experiments the performance of MDC and LC now using three other transmission schemes that are not rate-distortion optimized. Finally, we end the paper with concluding remarks in Section 4.

2. RATE-DISTORTION OPTIMIZED PACKET SCHEDULING AND PATH SELECTION

The framework that is presented here addresses the application of diversity to streaming of packetized media in a rate-distortion optimized way. In a packet switched network, diversity is achieved by using multiple transmission paths for streaming the media data over the network. It is assumed that the network paths exhibit independent loss and delay characteristics. Packets may be lost in any of the paths due to congestion or erasures. The problem under consideration is illustrated in Figure 1.

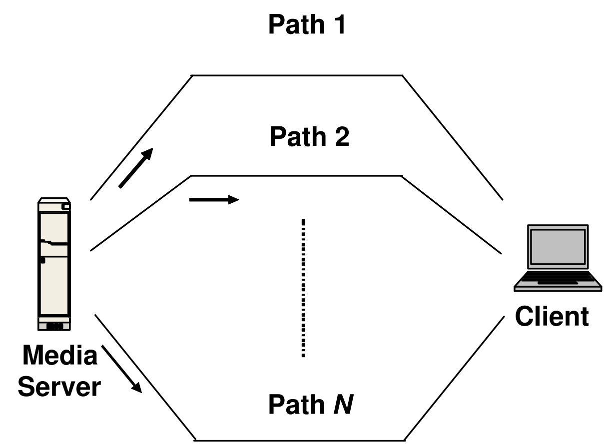

Figure 1: Media streaming over multiple network paths.

2.1 Packet Loss Probabilities

In a streaming media system, the encoded data are packetized into *data units* and are stored in a file on a media server. All of the data units in the presentation have interdependencies, which can be expressed by a directed acyclic graph. Associated with each data unit l is a size B_l, a decoding time $t_{DTS,l}$, and an importance Δd_l. The size B_l is the size of the data unit in bytes. $t_{DTS,l}$ is the *delivery deadline* by which data unit l must arrive at the client, or be too late to be usefully decoded. Packets containing data units that arrive after the data units' delivery deadlines are discarded. The importance Δd_l is the amount by which the distortion at the client will *decrease* if the data unit arrives on time at the client and is decoded.

We model the forward and backward communication channels on a network path between a server and a client, as burst loss channels using a K-state discrete-time Hidden Markov Model (HMM). The forward and the backward channel make state transitions independently of each other every T seconds, where the transitions are described by probability matrices $\mathcal{P}_{(F)}$ and $\mathcal{P}_{(B)}$, respectively. In each state the forward and the backward channel are characterized as an independent time-invariant packet erasure channel with random delay. Hence, they are completely specified with the probability of packet loss $\epsilon^k_{F/B}$ and the probability density of the transmission delay $p^k_{F/B}$, for $k = 1, \ldots, K$. This means that if the media server sends a packet on the forward channel at time t, given that the forward channel is in state k at t, then the packet is lost with probability ϵ^k_F. However, if the packet is not lost, then it arrives at the client at time t', where the forward trip time $FTT^k = t' - t$ is randomly drawn according to the probability density p_F^k. Therefore, we let $P\{FTT^k > \tau\} = \epsilon_F^k + (1-\epsilon_F^k)\int_\tau^\infty p_F^k(t)dt$ denote the probability that a packet transmitted by the server at time t, given that the forward channel is in state k at t, does not arrive at the client by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ. Then similarly, $P\{BTT^k > \tau\} = \epsilon_B^k + (1-\epsilon_B^k)\int_\tau^\infty p_B^k(t)dt$ denotes the probability that a packet transmitted by the client at time t, given that the backward channel is in state k at t, does not arrive at the server by time $t+\tau$, whether it is lost in the network or simply delayed by more than τ. Finally, we are interested in $P\{RTT^{kj} > \tau\}$, which is the probability that the server does not receive an acknowledgement by time $t+\tau$ for a packet transmitted at time t, given that the forward and the backward channel are respectively in states k and j, at t.

To derive $P\{RTT^{kj} > \tau\}$ assume first that the transmission on the forward channel occurred immediately after the channel made a state transition. If $FTT^k \leq T$, the packet is received by the client before the backward channel makes the next state transition. Then $P\{RTT^{kj} > \tau | FTT^k \leq T\} = P\{FTT^k + BTT^j > \tau | FTT^k \leq T\}$ as the client sends an acknowledgement while the backward channel is still in the current state j. The probability of this event is $P\{FTT^k \leq T\}$. However, if $lT < FTT^k \leq (l+1)T$, for $l \geq 1$, then the state of the backward channel makes l transitions before the packet actually arrives at the client. The probability of this event is $P\{lT < FTT^k \leq (l+1)T\}$. Here the state on the backward channel when the acknowledgement is sent can be any of the K possible values. Hence we compute the desired quantity as the expected value over all of them, i.e., $P\{RTT^{kj} > \tau | lT < FTT^k \leq (l+1)T\} = \sum_{p=1}^K \mathcal{P}_{jp(B)}^{(l)} P\{FTT^k + BTT^p > \tau | lT < FTT^k \leq (l+1)T\}$. Note that $\mathcal{P}_{jp(B)}^{(l)}$ is the probability of making a transition from state j to state p in l transition intervals. These probabilities are obtained using matrix power, i.e., $\mathcal{P}_{(B)}^{(l)} = \mathcal{P}_{(B)}^l$. Finally, we obtain $P\{RTT^{kj} > \tau\}$ by averaging over all possible outcomes for FTT^k.

2.2 Rate-distortion optimized policy selection

Suppose there are L data units in the media presentation. Let $\pi_l \in \Pi$ be the transmission policy for data unit $l \in \{1, \ldots, L\}$ and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ be the vector of transmission policies for all L data units. Π is a family of policies defined precisely in the next subsection.

Any given policy vector $\boldsymbol{\pi}$ induces an expected distortion $D(\boldsymbol{\pi})$ and an expected transmission rate $R(\boldsymbol{\pi})$ for the media presentation. We seek the policy vector $\boldsymbol{\pi}$ that minimizes $D(\boldsymbol{\pi})$ subject to a constraint on $R(\boldsymbol{\pi})$. This can be achieved by minimizing the Lagrangian $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$ for some Lagrange multiplier $\lambda > 0$, thus achieving a point on the lower convex hull of the set of all achievable distortion-rate pairs.

We now compute expressions for $R(\boldsymbol{\pi})$ and $D(\boldsymbol{\pi})$. The expected transmission rate $R(\boldsymbol{\pi})$ is the sum of the expected number of bytes transmitted for each data unit $l \in \{1, \ldots, L\}$, $R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l)$, where B_l is the number of bytes in data unit l and $\rho(\pi_l)$ is the expected number of transmitted bytes per source byte (under policy π_l), called the *expected cost*. The expected distortion $D(\boldsymbol{\pi})$ can be expressed in terms of the probability $\epsilon(\pi_l)$ that data unit l does not arrive at the receiver on time (under policy π_l), called the *expected error*. We borrow the expression for $D(\boldsymbol{\pi})$ from [37]

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

where $N_c^{(l)} = \{1, \ldots, l\}$ is the set of data units that the receiver considers for error concealment in case data unit l is not decodable by the receiver on time. D_0 is the expected reconstruction error for the presentation if no data units are received. $\Delta d_l^{(l_1)}$, for $l_1 \in N_c^{(l)}$, is the reduction in distortion if data unit l is not decodable and is concealed with a previous data unit l_1 that is received and decoded on time. $A(l_1)$ is the set of ancestors of l_1, including l_1. $C(l, l_1)$ is the set of data units $j \in N_c^{(l)} : j > l_1$ that are not mutual descendants, i.e., for $j, k \in C(l, l_1) : j \notin D(k), k \notin D(j)$, where $D(j)$ is the set of descendants of data unit j. "$\setminus$" denotes the operator "set difference". In deriving (1), we assume statistical independence of the losses affecting separate data units for tractability. For a further discussion, see [37].

Finding a policy vector $\boldsymbol{\pi}$ that minimizes the expected Lagrangian $J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$, for $\lambda > 0$, is difficult since the terms involving the individual policies π_l in $J(\boldsymbol{\pi})$ are not independent. Therefore, we employ an iterative descent algorithm, called Iterative Sensitivity Adjustment (ISA), in which we minimize the objective function $J(\pi_1, \ldots, \pi_L)$ one variable at a time while keeping the other variables constant, until convergence [38]. It can be shown that the optimal individual policies at iteration n, for $n = 1, 2, \ldots$, are given by

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

where $S_l^{(n)} = \sum_{l_1 \,:\, l \in N_c^{(l_1)}} S_{l,l_1}^{+(n)} - S_{l,l_1}^{-(n)} = S_l^{+(n)} - S_l^{-(n)}$ can be regarded as the *sensitivity* to losing data unit l, i.e., the amount by which the expected distortion will increase if data unit l cannot be recovered at the client, given the current transmission policies for the other data units. Note that differently from [38], the sensitivity here consists of two nonnegative terms $S_l^{+(n)}$ and $S_l^{-(n)}$. The first term increases the sensitivity associated with data unit l in case l is in the ancestor set of data unit l_2 used for concealment of a data unit l_1. On the other hand, the second term reduces the sensitivity associated with l in case l is not in the ancestor set of l_2. This result is intuitive and allows us to better model the situations where data unit l is irrelevant for concealment of another data unit. Expressions for $S_{l,l_1}^{+(n)}$ and $S_{l,l_1}^{-(n)}$ are easily obtained from (1) by grouping terms.

The minimization (2) is now simple, since each data unit l can be considered in isolation. Indeed the optimal transmission policy $\pi_l \in \Pi$ for data unit l minimizes the "per data unit" Lagrangian $\epsilon(\pi_l) + \lambda' \rho(\pi_l)$, where $\lambda' = \lambda B_l / S_l^{(n)}$. Thus to minimize (2) for any l and λ', it suffices to know the lower convex hull $\epsilon(\rho) = \min_{\pi \in \Pi} \{\epsilon(\pi) : \rho(\pi) \leq \rho\}$ of the function, which we call the expected *error-cost* function. In the next subsection we show how to compute the expected error-cost function for the family of policies corresponding to sender-driven transmission with packet path diversity.

2.3 Computing the expected error-cost function

Assume that there are M network paths over which the server can simultaneously send a data unit to the client. Furthermore assume that there are N discrete transmission opportunities $t_0, t_1, \ldots, t_{N-1}$ prior to the data unit's delivery deadline t_{DTS} at which the server is allowed to transmit a packet for the data unit on the forward channel of any $m \leq M$ paths. The server need not transmit a packet at every transmission opportunity and it does not transmit any further packets after an ACK is received on the backward channel of any of the paths.

At each transmission opportunity t_i, $i = 0, 1, \ldots, N-1$, the server takes an action $a_i = [a_{i1}, \ldots, a_{iM}]$, where $a_{im} = 1$ means that a packet is sent on the forward channel of path m and $a_{im} = 0$ means that no packet is sent on the forward channel of path m. Then, at the next transmission opportunity t_{i+1}, the server makes an observation o_i, where o_i is the set of acknowledgements received by the server in the interval $(t_i, t_{i+1}]$. For example, $o_i = \{ACK_{j_1}^{m_1}, ACK_{j_2}^{m_2}\}$ means that during the interval $(t_i, t_{i+1}]$, ACKs arrived on the backward channels for the packets sent at time t_{j_1} and t_{j_2} on the forward channels of paths m_1 and m_2, respectively. The history, or the sequence of action-observation pairs $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_i, o_i)$ leading up to time t_{i+1}, determines the state q_{i+1} at time t_{i+1}, as illustrated in Figure 2. If the final observation o_i includes an ACK, then q_{i+1} is a final state. In addition, any state at time $t_N = t_{DTS}$ is a final state. Final states in Figure 2 are indicated by double circles.

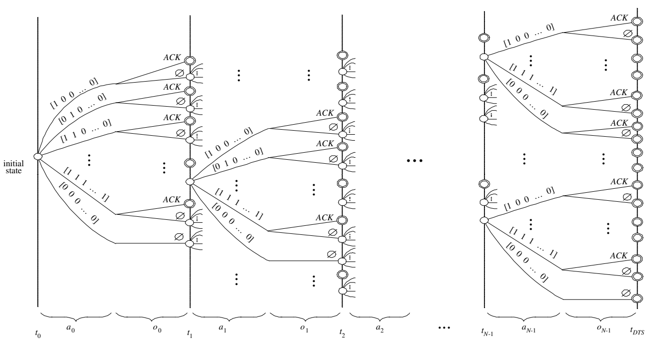

Figure 2: Markov decision tree for a data unit with packet path diversity.

The action a_i taken at a non-final state q_i determines the transition probabilities $P(q_{i+1}|q_i, a_i)$ to the next state q_{i+1}. Formally, a policy π is a mapping $q \mapsto a$ from non-final states to actions. Thus any policy π induces a Markov chain with transition probabilities $P_\pi(q_{i+1}|q_i) \equiv P(q_{i+1}|q_i, \pi(q_i))$, and consequently also induces a probability distribution on final states. Let q_F be a final state with history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{F-1}, o_{F-1})$, and let $q_{i+1} = q_i \circ (a_i, o_i)$, $i = 1, \ldots, F-1$, be the sequence of states leading up to q_F. Then q_F has probability $P_\pi(q_F) = \prod_{i=0}^{F-1} P_\pi(q_{i+1}|q_i)$, transmission cost $\rho_\pi(q_F) = \sum_{i=0}^{F-1} \sum_{m=1}^{M} a_{im}$, and error $\epsilon_\pi(q_F) = 0$ if o_{F-1} contains an ACK and otherwise $\epsilon_\pi(q_F)$ is equal to the probability that none of the transmitted packets arrives at the client on time, given q_F. Hence, we can express the expected cost and error for the Markov chain induced by policy π: $\rho(\pi) = E_\pi \rho_\pi(q_F) = \sum_{q_F} P_\pi(q_F) \rho_\pi(q_F)$, $\epsilon(\pi) = E_\pi \epsilon_\pi(q_F) = \sum_{q_F} P_\pi(q_F) \epsilon_\pi(q_F)$.

We wish to find the policy π^* that minimizes $\epsilon(\pi) + \lambda' \rho(\pi)$, as discussed in the previous section. We do that by enumerating all possible policies π, plotting the error-cost performances $\{(\rho(\pi), \epsilon(\pi))\}$ in the error-cost plane, and producing an operational error-cost function for our scenario. At every transmission opportunity t_i we find π^*, where $\{(\rho(\pi), \epsilon(\pi)) : \pi \in \Pi\}$ is calculated conditioned on q_i and all the policies under consideration are consistent with the history $(a_0, o_0) \circ (a_1, o_1) \circ \cdots \circ (a_{i-1}, o_{i-1})$ leading up to state q_i at time t_i. Then, a_i is set to the first action $\pi^*(q_i)$ of π^*, and the procedure is repeated at each successive transmission opportunity until a final state is reached.

In the following we explain how $\epsilon(\pi)$ and $\rho(\pi)$ are computed. Let t_i be the current transmission opportunity and let $C_{jm}^F, C_{jm}^B \in \{1, \ldots, K\}$ be respectively the states on the forward and the backward channel of path $m = 1, \ldots, M$ at transmission opportunity $t_j : j \leq i$. We assume that the sender has this information available. It is a reasonable assumption, as any congestion control mechanism employed by a streaming media system, such as those in [8, 39, 40, 41, 9, 42, 10, 43], will include some kind of channel estimation. This is because in the absence of explicit feedback from the network, the sender and/or the receiver must infer the state of the network by observing data units as they enter and leave the network. As explained earlier, the expected error for a policy π is simply the probability that all the transmitted packets from π as well as those from the transmission history do not arrive at the client on time. Furthermore, upon receipt of an acknowledgement packet, the server truncates its transmission pattern and does not consider sending any packets afterwards. Therefore, the cost for each transmission $a_{jp} = 1 : j \in \{i, \ldots, N-1\}, p = 1, \ldots, M$ is equal to the probability that none of the previous transmissions results in an acknowledgement packet received by the server by t_j. Hence, the expected cost is simply the sum of the individual costs over all transmission opportunities and paths. Due to space considerations, we omit here the expressions for $\epsilon(\pi)$ and $\rho(\pi)$ and refer the reader to [37].

3. EXPERIMENTAL COMPARISON

In this section, we report out experimental results. The experiments were carried out in Matlab [44]. We investigate the end-to-end distortion-rate performance of the two source coding techniques for streaming packetized video contents using different transmission schemes. First, we examine their performance when the RaDiO framework is employed to compute the packet transmission schedules. Then, we study their performance with transmission schemes that are not optimized in a rate-distortion sense. The videos used in the experiments are the QCIF image sequences *Foreman* and *Mother and Daughter*, henceforth denoted *MaD*. Using the H.263+ [45] codec, we have created a two-layer SNR scalable representation for each of the videos, shown in Figure 3. In particular, 130 frames of *Foreman* have been encoded into a base and enhancement layer with corresponding rates of 32 and 64 Kbps. Similarly, the first 130 frames of *MaD* have been encoded into two layers with rates 32 and 69 Kbps, respectively. Furthermore, for both videos the encoding frame rate was 10 fps with a Group of Pictures (GOP) consisting of an I frame followed by 9 consecutive P frames.

We also used the H.263+ codec to create MDC repre-

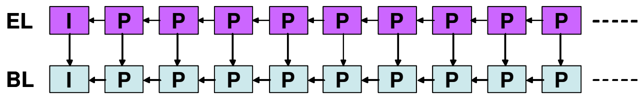

Figure 3: Two-layer SNR scalable representation.

sentations of the videos. We employed two MDC schemes, henceforth denoted MDC1 and MDC2, which are illustrated in Figure 4. MDC1 creates two completely independent descriptions by coding the odd and the even frames of a sequence separately. On the other hand, MDC2 creates two descriptions that are not completely independent, since the GOPs for both descriptions share the same I frame, as shown in Figure 4b. In fact, MDC2 is analogous to the Video Redundancy Coding (VRC) strategy proposed in [46]. In the case of both MDC1 and MDC2, each description was encoded at a bit rate that was half of that used to create the LC representation for the particular video. Moreover, the encoding frame rate was 5 fps with a GOP size of 5 frames. Therefore, combining the two descriptions gives an equivalent frame rate of 10 fps and a coding rate that closely matches the coding rate of LC for each of the videos. This was done in order to obtain equal (or quite similar) performance of MDC and LC in terms of compression efficiency and thus to emphasize the differences in end-to-end performance that are solely due to transmission aspects. Note that MDC1 is more robust to transmission losses, as the two descriptions are completely independent, which on the other hand reduces the compression efficiency compared to MDC2. Furthermore, for both MDC and LC representations it is assumed that there is a one-to-one correspondence between transmitted packets and data units, i.e., coded video frames. Therefore, from now on the terms "packet" and "data unit" are used interchangeably. Finally, as an error-concealment strategy we used the last received frame to conceal lost/late frames.

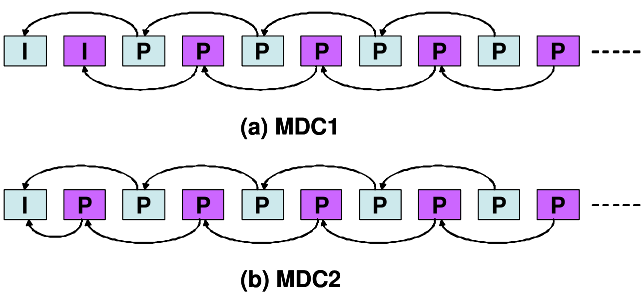

Figure 4: Two MDC schemes (a) MDC1, (b) MDC2.

For each transmission scenario comprising a source encoding technique, a test video and a transmission scheme the end-to-end distortion-rate performance was obtained through extensive simulations. A simulation run consisted of streaming the test video in Matlab, with and without path diversity, using the transmission scheme under consideration. We used $T = 100$ ms as the time interval between transmission opportunities and 600 ms for the playback delay. Furthermore, we employed a $K = 2$ state HMM for the forward and the backward channel on a network path. The state transition probabilities of the HMM were selected such that both states were equiprobable and on average state transitions occurred every 1 s. In a simulation run, a new state value for a channel was generated every T ms using a pseudo-random number generator provided by Matlab. In particular, the initial state value ($t = 0$) was generated as a realization of a random variable distributed according to the stationary state probabilities. In addition, the state value at each subsequent time instance nT, for $n = 1, 2, \ldots$, was generated as a realization of a random variable distributed according to the state transition probabilities for the state value at time $(n-1)T$. In Table 1 we specify the loss and delay characteristics for each state of a path. We kept the same characteristics for the forward and the backward channel of a path.

	Loss	Delay		
	ϵ (%)	κ (ms)	μ (ms)	σ (ms)
State 1	3	25	75	50
State 2	15	25	275	250

Table 1: Loss and delay parameters for a network path.

The delay density was modeled using a shifted Gamma distribution specified with three parameters: shift κ, mean μ and standard deviation σ. Similarly to the state generation process, the random channel effects on packet transmissions were created using a pseudo-random number generator. In particular, at each time instance nT, $n = 0, 1, \ldots$, transmitted packets on a channel are dropped at random according to the loss rate for the channel state value at nT. In addition, those packets that are not dropped received a random delay, again according to the delay distribution for the channel state at nT.

For each simulation run, we recorded the average peak signal-to-noise ratio (PSNR) in dB of the mean-square error (MSE) distortion of the luminance (Y) component of the decoded video and the average aggregate transmission rate (Kbps) that is available on the forward channel(s) of the network path(s). The distortion-rate performance for a transmission scenario was obtained then by averaging these numbers over multiple simulation runs. We performed 300 runs for each scenario in order to obtain statistically meaningful results.

We first examine the performance of LC and MDC for streaming *Foreman* with the RaDiO framework. In Figure 5 we show the performance of the two techniques when one network path is employed for streaming. It can be seen that LC provides a better performance than MDC1 over the whole range of available transmission rates. The difference in performance increases as we move from high to low transmission rates, which is explained by the following. At high transmission rates the packet scheduling framework ensures that all packets are delivered to the client on time. Therefore, here the difference in performance between the two techniques is only governed by their respective compression efficiencies. However, as the transmission rate is decreased the packet scheduling algorithm cannot ensure anymore that all packets are delivered on time due to the insufficient transmission bandwidth. Hence, the RaDiO framework needs to perform rate-distortion optimized selection of packets to be transmitted. Finally, since LC provides a more graceful degradation of reproduction quality than MDC in the event of insufficient transmission bandwidth, this increases further the performance difference between LC and MDC1 at lower transmission rates.

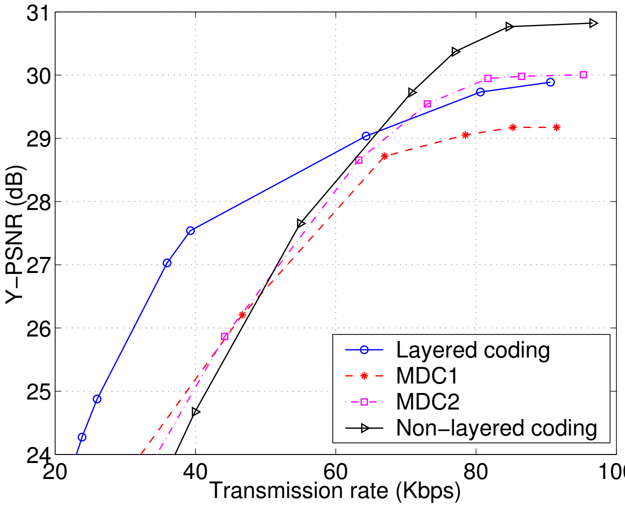

Figure 5: Layered coding vs. multiple description coding for QCIF *Foreman* and one network path. RaDiO packet scheduling is employed.

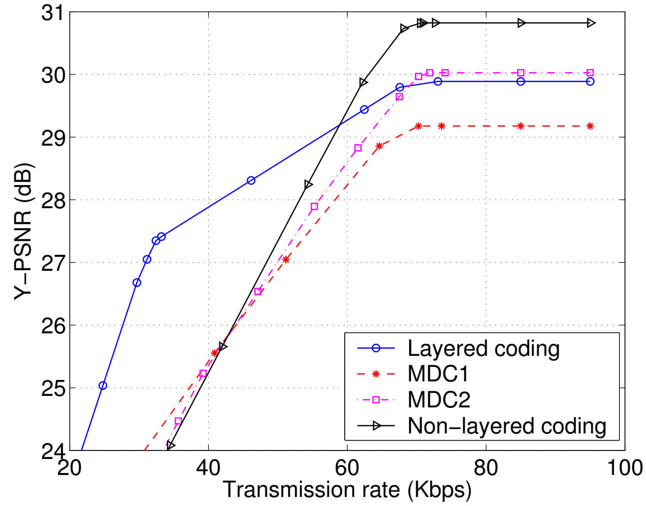

Figure 6: Layered coding vs. multiple description coding for QCIF *Foreman* and two network paths. RaDiO packet scheduling is employed.

We observe a similar relation between the performances of LC and MDC2. However, note that MDC2 is slightly more efficient in terms of compression and therefore it performs a little bit better at transmission rates larger than 65 Kbps. Lastly, for comparison purposes we also show in Figure 5 the performance of a streaming system, denoted henceforth Non-Layered Coding (NLC), based on the RaDiO framework and single layer source encoding. The single layer representation was encoded at rate of 64 Kbps using the same H.263+ codec that we used to obtain LC. The rest of the encoding parameters were kept the same as for the layered representation. It can be seen that NLC performs better than LC and MDC at high transmission rates. This is expected as NLC provides the best compression efficiency. On the other hand, NLC also exhibits the worst performance degradation as the transmission rate is decreased. This is due to the fact that SC does not exhibit any error resilience contrary to MDC and provides a much worse rate-distortion trade-off than LC.

Then, we examine the performance of LC, MDC and NLC for streaming *Foreman* in the presence of diversity again using the RaDiO framework as a transmission scheme. As explained in Section 2, in this setting, in addition to packet scheduling the RaDiO framework also performs path selection for each transmission. Furthermore, note that here we record the average aggregate transmission rate that is available on the forward channels of the network paths, as explained earlier. It can be seen from Figure 6 that the performance of each of the source coding techniques improves when two network paths are employed for streaming. In addition, we can observe from Figure 6 that the relative difference in performance between LC and MDC is more pronounced here as the transmission rate decreases, compared to the case of streaming over a single network path. This is due to the fact that with diversity the graceful degradation of reproduction quality in the event of insufficient transmission rate provided by LC can be exploited more efficiently.

Next, we examine the performance of LC, MDC and NLC for streaming *MaD* with the RaDiO framework. In Figure 7 we show their performances for the case of streaming over one network path. It can be observed that the variations in performance between the different source coding techniques follow the same pattern as those observed in Figure 5. For high transmission rates, it is the differences in compression efficiency that determine the differences in end-to-end performance. On the other hand, for medium and low transmission rates, it is the rate-distortion trade-off provided by each of the techniques that governs their relative end-to-end performance. Note, however, that the differences in performance here are not as pronounced as those for streaming *Foreman*. This is due to the fact that the *MaD* sequence exhibits considerably less motion which makes the encoding process less complex and allows more successful error concealment applied to missing video frames.

We observed similar results for streaming *MaD* over two network paths, shown in Figure 8. As in the case of *Foreman*, streaming with two network paths provides improved performance over streaming with one network path, for each of the source coding techniques. However, the gains in performance are not as significant as those for *Foreman*, due to the nature of the *MaD* sequence, as explained earlier.

A few concluding remarks can be made from the results presented so far. Layered source representations provide a more balanced end-to-end performance over lossy packet networks in the event of varying transmission bandwidth when the RaDiO framework is employed for packet scheduling. In other words, the performance of LC exhibits much less variations than that of MDC and NLC as the transmission rate is varied. This is due to the fact that the packet schedules computed by the RaDiO framework actually result in unequal error protection provided to different portions of the media stream. This overcomes the sensitivity of layered representations to transmission losses and therefore provides the necessary error resilience without reducing the compression efficiency of LC. Ultimately, this enables a streaming system based on RaDiO packet scheduling and LC to outperform a system based on RaDiO packet scheduling and MDC.

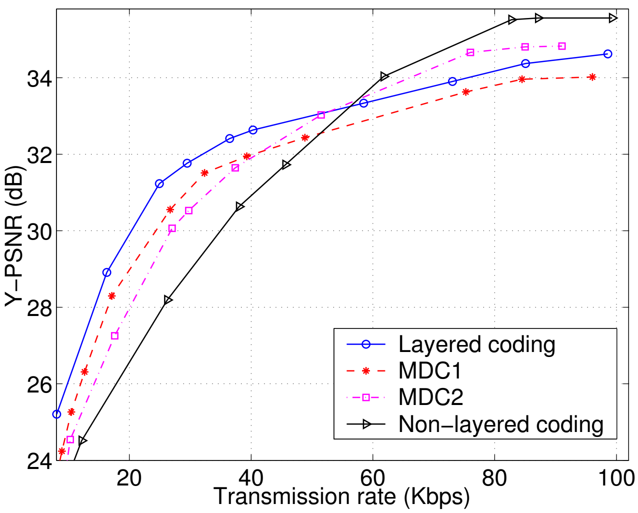

Figure 7: Layered coding vs. multiple description coding for QCIF *MaD* and one network path. RaDiO packet scheduling is employed.

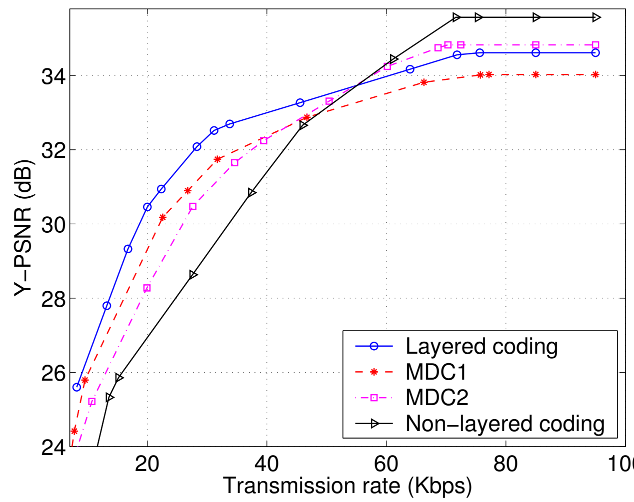

Figure 8: Layered coding vs. multiple description coding for QCIF *MaD* and two network paths. RaDiO packet scheduling is employed.

The following results represent the performance of LC and MDC with transmission schemes that are not distortion-rate optimized. In other words, these schemes do not take into account the importances of the individual data units and their interdependencies when computing the packet transmission schedules. Note that due to space considerations, we include only the results for streaming *Foreman* over a single network path. However, that does not affect the discussion and the conclusions presented here, as the results for streaming with and without diversity are analogous. First, we examine the performance of LC and MDC in a streaming scenario where data units are transmitted at most once, in a GOP order starting with those data units whose delivery deadline is closest to the current transmission opportunity. No feedback is provided by the client. In order to perform rate control, the number of data units transmitted at a transmission opportunity is kept proportional to the available transmission rate. It can be seen from Figure 9 that in this scenario both MDC1 and MDC2 outperform LC over the whole range of transmission rates. The difference in performance is quite significant and remains steadily around 2 dB almost over the whole range of employed transmission rates. It is only at high rates that the performance difference decreases to roughly 1.7 dB. The observed performance of LC and MDC in this scenario is due to the fact that having a source representation with incorporated error resilience enables MDC1 and MDC2 to provide an end-to-end performance that is less affected by packet losses.

Then we examine a scenario where each data unit can be transmitted at most three times. The data units are again transmitted in a GOP order based on their delivery deadline as in the previous scenario. Likewise, no feedback from the client is employed. In addition, at every transmission opportunity preference is given to data units that have not been transmitted yet or have been already transmitted less number of times compared to other data units. For example, if data units k and $k+1$ have been already transmitted twice and once, respectively, then data unit $k+1$ is retransmitted prior to data unit k. Rate control is performed in the same fashion as in the previous scenario. The results for this scenario are quite similar to those for the previous scenario. It can be seen from Figure 10 that again both MDC1 and MDC2 outperform LC with a quite significant margin. It is interesting to note that this transmission scheme requires a somewhat larger transmission rate to achieve the same distortion performance compared to the previous scenario, for each of the source coding techniques. This is due to the fact that each data unit can be transmitted more than once, which might not be necessary if the data unit have been already received by the client after the first transmission.

Finally, we examine a scenario of time-out driven transmission. Contrary to the previous two schemes, here the client immediately responds with an acknowledgement packet whenever a data packet arrives. Similarly to the previous two scenarios, at each transmission opportunity data units are transmitted in a GOP order based on their delivery deadline. In addition, a data unit is considered for retransmission if an acknowledgement packet is not received by the server within $3T$ time from the previous transmission of the data unit. T is the time interval between transmission opportunities introduced earlier. Finally, as in the previous scenario at each transmission opportunity preference is given to data units that have not been transmitted yet or have been already transmitted less number of times compared to other data units. It can be seen from Figure 11 that still in this scenario MDC2 outperforms LC over the whole range of transmission rates. However, the difference in performance here is not as significant as in the previous two scenarios. This is due to the fact that the employed transmission scheme is more bandwidth efficient than the previous two. The following can be summarized from Figure 11. For transmission rates less than 75 Kbps, it is the increased error resilience that enables MDC2 to perform better than LC. On the other hand, for rates greater than 80 Kbps it is the compression efficiency that is the key factor in determining the end-to-end performance since here there is anyway sufficient bandwidth for the transmission scheme to ensure on-time delivery of most of the data units. Similar con-

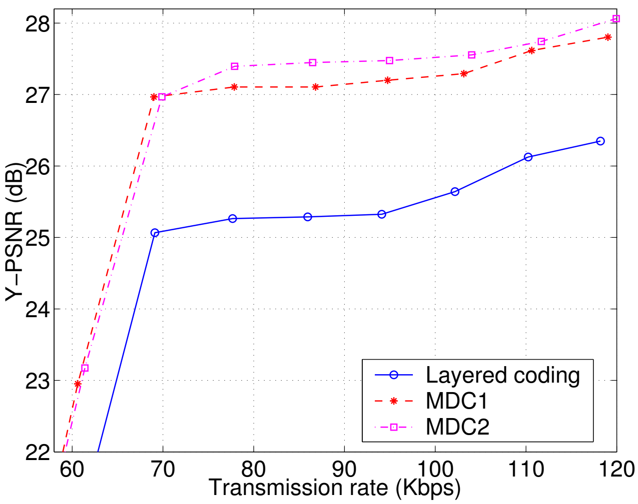

Figure 9: Layered coding vs. multiple description coding for QCIF *Foreman* and one network path. Each data unit is transmitted at most once.

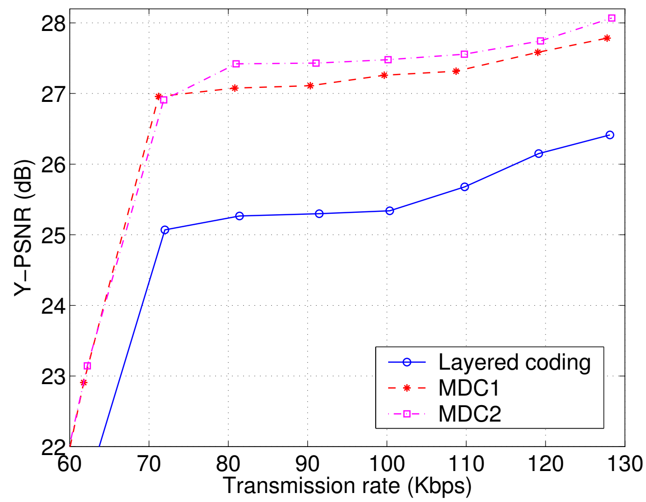

Figure 10: Layered coding vs. multiple description coding for QCIF *Foreman* and one network path. Each data unit is transmitted at most three times.

clusions apply to the performance comparison between LC and MDC1. Note that MDC1 is less compression efficient than LC, hence the observed performance in Figure 11 for transmission rates larger than 80 Kbps.

We also examined the performance of LC and MDC for streaming *MaD* with and without diversity in the last three scenarios. These results are not included here due to space considerations. However, they are similar to those obtained for streaming *Foreman* and therefore the same discussion and conclusions apply. The only notable difference is the fact that for *MaD* the performance differences between the individual source coding techniques for each of the scenarios and between different scenarios are not as pronounced as those for *Foreman*. This is due to the nature of the *MaD* video, as explained earlier.

4. CONCLUSIONS

In this paper, we study the performance of specific implementations of multiple description coding and of layered coding for streaming packetized video content over the Internet. Layered coding provides a scalable representation that enhances rate-control but is sensitive to transmission losses. On the other hand, multiple description coding provides increased resilience to packet losses by creating multiple streams that can be decoded independently. We compare the performance of these two source coding techniques using different transmission schemes. Scenarios where transmission over multiple network paths is employed are also considered. As seen from the experimental results in Section 3 the relative performance of the two techniques varies substantially depending on the transmission scenario under consideration. Given the specific implementations of multiple description coding and of layered coding employed in this paper, it is seen that layered coding outperforms multiple description coding when rate-distortion optimized scheduling of the packet transmission is employed. The converse is observed for scenarios where the packet schedules are oblivious to the importances of the individual packets and their interdependencies. These are the key results of the present work. We believe that they can be useful in determining the appropriate source representation for a streaming session depending on the available transmission scheme.

5. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable suggestions. They are also grateful to the assigned shepherd for providing many insightful comments and recommendations.

6. REFERENCES

[1] M. Ghanbari, "Two-layer coding of video signals for vbr networks," *IEEE J. Selected Areas in Communications*, vol. 7, pp. 771–781, June 1989.

[2] B.-J. Kim, Z. Xiong, , and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374–1387, Dec. 2000.

[3] M. Khansari, A. Zakauddin, W.-Y. Chan, E. Dubois, and P. Mermelstein, "Approaches to layered coding for dual-rate wireless video transmission," in *Proc. Int'l Conf. Image Processing*, Austin, TX, Oct. 1994, IEEE, vol. 1, pp. 258–262.

[4] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable Internet video using MPEG-4," *Signal Processing: Image Communication*, vol. 15, no. 1–2, pp. 95–126, Sept. 1999.

[5] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, no. 1–2, pp. 77–94, Sept. 1999.

[6] ISO/IEC, "Information technology — coding of audio-visual objects: Visual (MPEG-4)," Final Committee Draft 14496-2, JTC1/SC29/WG11, Tokyo, Mar. 1998.

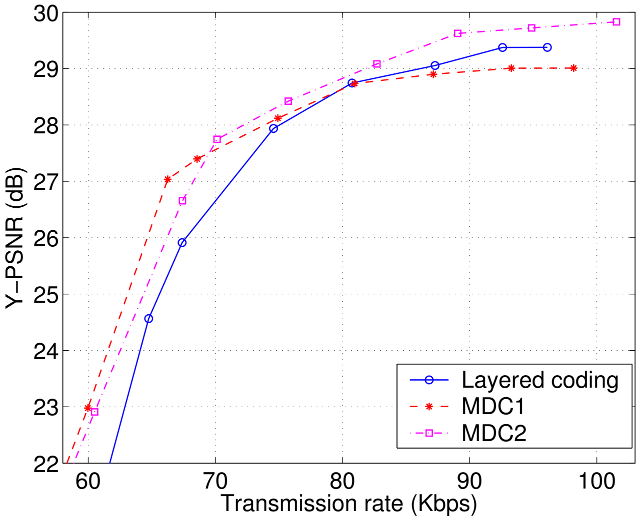

Figure 11: Layered coding vs. multiple description coding for QCIF *Foreman* and one network path. Time-out driven scheduling is employed.

[7] Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *ITU-T Recommendation H.263 Version 2*, Feb. 1998.

[8] J. Madhavi and S. Floyd, "TCP-friendly unicast rate-based flow control," *Technical note sent to the end2end-interest mailing list*, Jan. 1997, http://www.psc.edu/networking/papers/tcp_friendly.html.

[9] W.-T. Tan and A. Zakhor, "Internet video using error resilient scalable compression and cooperative transport protocol," in *Proc. Int'l Conf. Image Processing*, Chicago, IL, Oct. 1998, IEEE, vol. 3, pp. 458–462.

[10] S. Floyd, M. Handley, and J. Padhye, "Equation-based congestion control for unicast applications," Tech. Rep. TR-00-03, International Computer Science Institute, Berkeley, CA, Mar. 2000.

[11] S. R. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Selected Areas in Communications*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.

[12] W.-T. Tan and A. Zakhor, "Multicast transmission of scalable video using receiver-driven hierarchical FEC," in *Proc. Int'l Packet Video Workshop*, New York, Apr. 1999.

[13] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 108–122, Mar. 2001.

[14] S. Blake, D. Black, M. Carlson, E. Davies, and Z. Wang, "An architecture for differentiated services," Tech. Rep. RFC-2475, IETF, Dec. 1998.

[15] E. Masala, D. Quaglia, and J. C. de Martin, "Adaptive picture slicing for distortion-based classification of video packets," in *Proc. Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, IEEE, pp. 111–116.

[16] J. Shin, J. Kim, and C.-C. J. Kuo, "Relative priority based QoS interaction between video applications and differentiated service networks," in *Proc. Int'l Conf. Image Processing*, Vancouver, Canada, Oct. 2000, IEEE, vol. 3, pp. 536–539.

[17] J. Shin, J. Kim, and C.-C. J. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 219–231, June 2001.

[18] D. Quaglia and J. C. de Martin, "Delivery of MPEG video streams with constant perceptual quality of service," in *Proc. Int'l Conf. Multimedia and Exhibition*, Lausanne, Switzerland, Aug. 2002, IEEE, vol. 2, pp. 85–88.

[19] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, "Multiple description coding for video using motion compensated prediction," in *Proc. Int'l Conf. Image Processing*, Kobe, Japan, Oct. 1999, IEEE, vol. 3, pp. 837–841.

[20] D. Comas, R. Singh, and A. Ortega, "Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding," in *Proc. Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 581–586, IEEE.

[21] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," in *Proc. Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 441–447, IEEE.

[22] Y.-C. Lee and Y. Altunbasak, "A collaborative multiple description transform coding and statistical error concealment method for error resilient video streaming over noisy channels," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002, IEEE, vol. 2, pp. 2077–2080, IEEE.

[23] A. Reibman, H. Jafarkhani, M. Orchard, and Y. Wang, "Performance of multiple description coders on a real channel," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Phoenix, AZ, USA, Mar. 1999, IEEE, vol. 5, pp. 2415–2418.

[24] R. Singh, A. Ortega, L. Perret, and W. Jiang, "Comparison of multiple description coding and layered coding based on network simulations," in *Proc. Visual Communications and Image Processing*, San Jose, CA, USA, Jan. 2000, SPIE, vol. 3974, pp. 929–939.

[25] A. Reibman, Y. Wang, X. Qiu, Z. Jiang, and K. Chawla, "Transmission of multiple description and layered video over an EGPRS wireless network," in *Proc. Int'l Conf. Image Processing*, Vancouver, BC, Oct. 2000, IEEE, vol. 2, pp. 136–139.

[26] Y. Wang, S. Panwar, S. Lin, and S. Mao, "Wireless video transport using path diversity: multiple description vs. layered coding," in *Proc. Int'l Conf. Image Processing*, Rochester, NY, Sept. 2002, IEEE, vol. 1, pp. 21–24.

[27] Y.-C. Lee, J. Kim, Y. Altunbasak, and R.M. Mersereau, "Performance comparisons of layered and multiple description coded video streaming over error-prone networks," in *Proc. Int'l Conf. Communications*, Anchorage, AK, USA, May 2003, IEEE, vol. 1, pp. 35–39.

[28] Y. Su, T. Tao, J. Lu, and J. Wang, "Channel-optimized video transmission over WCDMA system," in *Proc. Vehicular Technology Conference*, Birmingham, AL, May 2002, IEEE, vol. 1, pp. 265–269.

[29] H. Wang and A. Ortega, "Robust video communication by combining scalability and multiple description coding techniques," in *Proc. Symposium on Electronic Imaging*, San Jose, CA, Jan. 2003, SPIE.

[30] P.A. Chou, H.J. Wang, and V.N. Padmanabhan, "Layered multiple description coding," in *Proc. Int'l Packet Video Workshop*, Nantes, France, Apr. 2003.

[31] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. Visual Communications and Image Processing*, Seattle, WA, Jan. 2001, SPIE, pp. 329–409.

[32] J. Apostolopoulos and S.J. Wee, "Unbalanced multiple description video communication using path diversity," in *Proc. Int'l Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, IEEE, vol. 1, pp. 966–969.

[33] N. Gogate, D.-M. Chung, S.S. Panwar, and Y. Wang, "Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 9, pp. 777–792, Sept. 2002.

[34] Y.J. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *Proc. Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, Dec. 2002, IEEE, pp. 420–423.

[35] J. Apostolopoulos, W.-T. Tan, and S. Wee, "Performance of a multiple description streaming media content delivery network," in *Proc. Int'l Conf. Image Processing*, Rochester, NY, USA, Sept. 2002, IEEE, vol. 2, pp. 189–192.

[36] A. Sehgal and P. A. Chou, "Cost-distortion optimized streaming media over DiffServ networks," in *Proc. Int'l Conf. Multimedia and Exhibition*, Lausanne, Switzerland, Aug. 2002, IEEE, vol. 1, pp. 857–860.

[37] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conference*, Snowbird, UT, Mar. 2003, IEEE Computer Society, pp. 203–212.

[38] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, 2001, submitted.

[39] M. Mathis, J. Semke, S. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, no. 3, pp. 67–82, July 1997.

[40] T. Turletti, S.F. Parisis, and J.-C. Bolot, "Experiments with a layered transmission scheme over the Internet," Tech. Rep. 3296, INRIA, Sophia Antipolis, France, Nov. 1997.

[41] D. Sisalem and H. Schulzrinne, "The loss-delay adaptation algorithm: a TCP-friendly adaptation scheme," in *Proc. Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, UK, July 1998, ACM.

[42] R. Rejaie, M. Handley, and D. Estrin, "RAP: an end-to-end based congestion control mechanism for realtime streams in the Internet," in *Proc. Conf. on Computer Communications (INFOCOM)*, New York, NY, Mar. 1999, IEEE, vol. 3, pp. 1337–1345.

[43] Q. Zhang, Y.-Q. Zhang, and W. Zhu, "Resource allocation for audio and video streaming over the Internet," in *Proc. Int'l Symp. Circuits and Systems*, Geneva, Switzerland, May 2000, IEEE, vol. IV, pp. 21–24.

[44] "Matlab version 6.5," Tech. Rep., The Mathworks, http://www.mathworks.com.

[45] ITU-T SG16/Q15 (T. Gardos, ed.), "Video codec test model number 10 (TMN-10) (H.263+)," *ITU-T SG16/Q15 document Q15-D-65*, Apr. 1998.

[46] S. Wenger, G.D. Knorr, J. Ott, and F. Kossentini, "Error resilience support in H.263+," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 867–877, Nov. 1998.