

# Robust Video Communication by Combining Scalability and Multiple Description Coding Techniques

Huisheng Wang and Antonio Ortega \*

Integrated Media Systems Center,  
Department of Electrical Engineering-Systems,  
University of Southern California, Los Angeles, CA 90089  
{huishenw,ortega}@sipi.usc.edu

## ABSTRACT

Layered coding (LC) and multiple description coding (MDC) have been proposed as two different kinds of “quality adaptation” schemes for video delivery over the current Internet or wireless networks. To combine the advantages of LC and MDC, we present a new approach – Multiple Description Layered Coding (MDLC), to provide reliable video communication over a wider range of network scenarios and application requirements. MDLC improves LC in that it introduces redundancy in each layer so that the chance of receiving at least one description of base layer is greatly enhanced. Though LC and MDC are each good in limit cases (e.g., long end-to-end delay for LC vs. short delay for MDC), the proposed MDLC system can address intermediate cases as well. Same as a LC system with retransmission, the MDLC system can have a feedback channel to indicate which descriptions have been correctly received. Thus a low redundancy MDLC system can be implemented with our proposed runtime packet scheduling system based on the feedback information. The goal of our scheduling algorithm is to find a proper on-line packet scheduling policy to maximize the playback quality at the decoder. Previous work on scheduling algorithms has not considered multiple decoding choices due to the redundancy between data units, because of the increase in complexity involved in considering alternate decoding paths. In this paper, we introduce a new model of Directed Acyclic HyperGraph (DAHG) to represent the data dependencies among frames and layers, as well as the data correlation between descriptions. The impact of each data unit to others is represented by messages passing along the graph with updates based on new information received. Experimental results show that the proposed system provides more robust and efficient video communication for real-time applications over lossy packet networks.

**Keywords:** multimedia communication, layered coding, multiple description coding, multiple description layered coding, scheduling

## 1. INTRODUCTION

In this paper we present an efficient multiple description layered coding (MDLC) system for robust video communication over unreliable channels. Recent technological developments and the rapid growth of Internet and wireless networks make it feasible and more attractive to provide real-time video services over them [1]. However the current best-effort Internet does not offer any QoS guarantees. The congestion, routing delay and fluctuations of wireless channel conditions can all result in the packet loss or large delay during the transmission, and thus greatly degrade the received video quality.

A traditional method to deal with lossy transmission environments is Layered Coding (LC) [2], in which a video sequence is coded into a base layer and one or more enhancement layers. The enhancement layer is dependent on the base layer, and cannot be decoded if the base layer is not received. Thus LC requires the base layer to be transmitted in an essentially error free channel, realized via either strong FEC or ARQ schemes. FEC has the drawback of requiring increased bandwidth, even in cases when errors do not occur, while ARQ may not be a practical alternative if the round-trip time (RTT) is long relative to the end-to-end delay in the application.

---

This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.

Another alternative to reliable communication is Multiple Description Coding (MDC) [3]. With this coding scheme, a video sequence is coded into a number of separate bit streams (referred to as multiple descriptions), and transmitted through separate channels or through one physical channel with appropriate scheduling or interleaving techniques. Each description can be decoded independently to provide an acceptable reproduction of the original signal. This approach is also scalable in the sense that each correctly received description improves the quality of the decoded signal as they contain some information that was not available in the other descriptions. The advantage of MDC over LC is that it does not require special provisions in the network to provide such a reliable subchannel, as LC requires for the base layer. However MDC requires significant redundancy between the descriptions, which reduces the coding efficiency.

Reibman *et al* [4, 5] and Singh *et al* [6] have analyzed and compared LC to MDC in different network environments, including Binary Symmetric Channels [4], Random Erasure Channels [4], real-time delay constrained networks [6] and an EGRS wireless network [5]. Though their results do not match completely due to the different LC error resilient techniques and the network environments, they all conclude that in general, the performance of MDC is better than LC in network scenarios with high error rate, long RTT or stringent real time requirements. The different characteristics of these two approaches motivate us to look for an adaptive approach to combine their advantages so as to provide reliable video communication over a wider range of network scenarios and application requirements. The main novelty of our work is to demonstrate that it is possible to combine LC with MDC, by adding a standard-compatible enhancement to MPEG-4 version 2 [7]. The new Multiple Description Layered Coding (MDLC) approach presented in this paper introduces redundancy in each layer so that the chance of receiving at least one description of base layer is greatly enhanced. Furthermore, though LC and MDC are each good in limit cases (e.g., long end-to-end delay for LC vs. short delay for MDC), the proposed MDLC system can address intermediate cases as well.

A low redundancy MDLC system can be implemented with our proposed runtime packet scheduling system based on *a priori* channel knowledge and runtime feedback information. The goal of our scheduling algorithm is to find a proper on-line packet scheduling policy to maximize the playback quality at the decoder.

There has been some recent work on the scheduling algorithms of LC video streaming systems [8–11]. Chou, Miao and Ortega addressed this problem [8, 9] with a simple source model only considering one decoding choice: a packet can be decoded only when all of its parents are received and decodable. Implicitly this approach excludes the possibility of having multiple descriptions, in which several decoding choices are possible based on which descriptions are received at the receiver. Cheung and Tan introduce a more general formulation [10] to model the packet dependency and performance. They consider all possibilities of decoding and delivery scenarios, but this increases complexity substantially. In our approach we consider several decoding paths but ignore some decoding paths that lead to poor quality solutions, thus enabling a reduced complexity search.

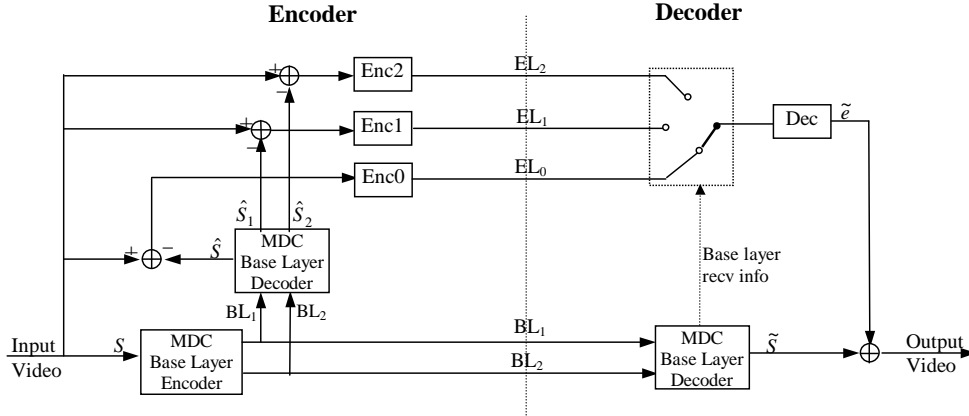
In this paper, we introduce a new Directed Acyclic HyperGraph (DAHG) to represent the data dependencies among frames and layers, as well as the data correlation between descriptions. This source model can be applied in either LC, MDC or MDLC systems, with moderate complexity increment when applied in a MDLC system. The proposed scheduling algorithm based on a DAHG takes into account the impact of delay constrained delivery, channel conditions, data dependencies among frames and layers, and data correlation between descriptions as well. We also introduce an adaptive double time window control on MDLC system which will adjust the system to behave more like LC or MDC depending on the actual channel condition. A runtime distortion value is then assigned to each packet, allowing to estimate their importance. At any given time the most important packet is sent first.

This paper is organized as follows. In Section 2 we describe the proposed MDLC system including the MDLC coding scheme and the video communication system architecture. We then describe the DAHG source model in Section 3. Section 4 presents the proposed runtime scheduling algorithm based on DAHG, and the simulation results are shown in Section 5. Finally we conclude our work in Section 6.

## 2. THE PROPOSED MDLC SYSTEM

### 2.1. MDLC Codec

Our general approach to MDLC video coding uses an MDC encoder to generate two base layer descriptions  $BL_1$  and  $BL_2$ , shown in Fig. 1. Then the base layer MDC decoder in the MDLC encoder module mimics the three



**Figure 1.** Structure of the proposed MDLC codec.

possible decoding scenarios at the receiver: both descriptions received or either one received. If both descriptions are received, it reproduces the base layer as  $\hat{S}$ , and the difference between the original video input  $S$  and  $\hat{S}$  is coded with a standard encoder like MPEG-4 FGS into an enhancement layer stream  $EL_0$ . If only one description is received, the base layer decoder generates a low quality reproduction  $\hat{S}_1$  or  $\hat{S}_2$ , and feeds the difference into two enhancement layer encoders separately to create  $EL_1$  and  $EL_2$ .

The key advantage of our MDLC scheme is that it combines the LC hierarchical scalability coding scheme with the reliability introduced by adding redundancy into base layer with multiple descriptions. With a well-designed scheduling algorithm, the sender can choose only one base layer description and its corresponding enhancement layer to be sent to the receiver, as in a standard LC system, when the channel losses are low. Or it can send both base layer descriptions and their enhancement layer streams to get the maximum protection when the channel is becoming worse.  $EL_0$  is sent instead of either  $EL_1$  or  $EL_2$  to reduce the redundancy when both  $BL_1$  and  $BL_2$  are received or expected to be received at a high probability. The sender can select the packets to be transmitted at any given time during the transmission session based on the feedback information, in such a way as to maximize the playback quality at the decoder.

The proposed decoder system, depicted in Fig. 1, is composed of two parts: base layer MDC decoder, and enhancement layer switch and decoder. The base layer MDC decoder will generate a reproduction  $\tilde{S}$  which is  $\hat{S}$ ,  $\hat{S}_1$  or  $\hat{S}_2$  depending on what was received. The enhancement layer switch then selects which  $EL$  stream to decode given what base layer was received. Finally, the decoded base layer and enhancement layer will be combined together to generate the final video output.

Our codec is based on the MPEG-4 standard [7]. The base layer is obtained by applying a coarse quantizer to the original video in DCT domain. We create our multiple base layer descriptions by repeating important information like the motion vectors in inter mode and DC coefficients in intra mode. For the rest of the DCT coefficients, we just alternate them into the two descriptions. For example, if a macroblock is put into  $BL_1$ , then its neighbor macroblocks are put into  $BL_2$ . Information from both descriptions are combined before making predictions for the future frames. This MDC coding scheme is really simple, and more complicated schemes [12, 13] may be used instead to improve performance.

## 2.2. System Architecture

Fig. 2 shows an end-to-end video transmission system, in which each video frame is encoded, transmitted and decoded in real-time within some acceptable delay period. Different video applications may have different end-to-end delay (or latency) requirements. Interactive two-way communications will have very stringent delay requirements, while others (e.g., video streaming) can allow a relatively large initial startup delay.

The input video is compressed into several base layer and enhancement layer streams with our proposed MDLC codec. For a packet-switched network, these streams are packetized and then fed into the transmission

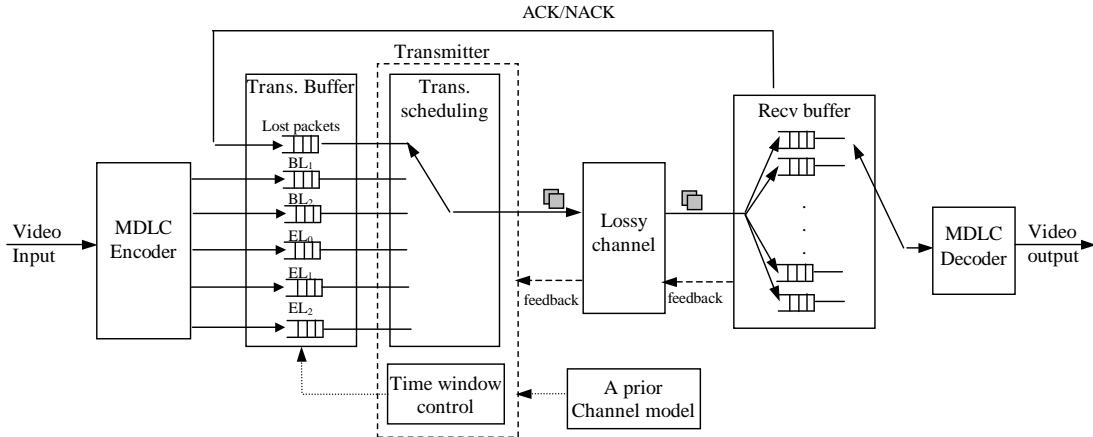


Figure 2. System architecture.

buffer. At the same time, there are also some packets in the buffer waiting for retransmission which have been reported lost based on ACK/NAK from the receiver. A transmitter includes a scheduler and a time window control. When given the observation of a channel and *a priori* model, the transmitter selects those packets which are the most important from the transmission buffer and put them on the channel. We consider here a lossy channel environment such as a packet erasure channel. When a packet arrives in the receiver buffer, the receiver responds to a correctly received packet with a positive acknowledgement (ACK), and a corrupted packet with a negative acknowledgement (NAK) as well. With the feedback information, the sender can estimate the channel packet loss rate and RTT based on the transmission history.

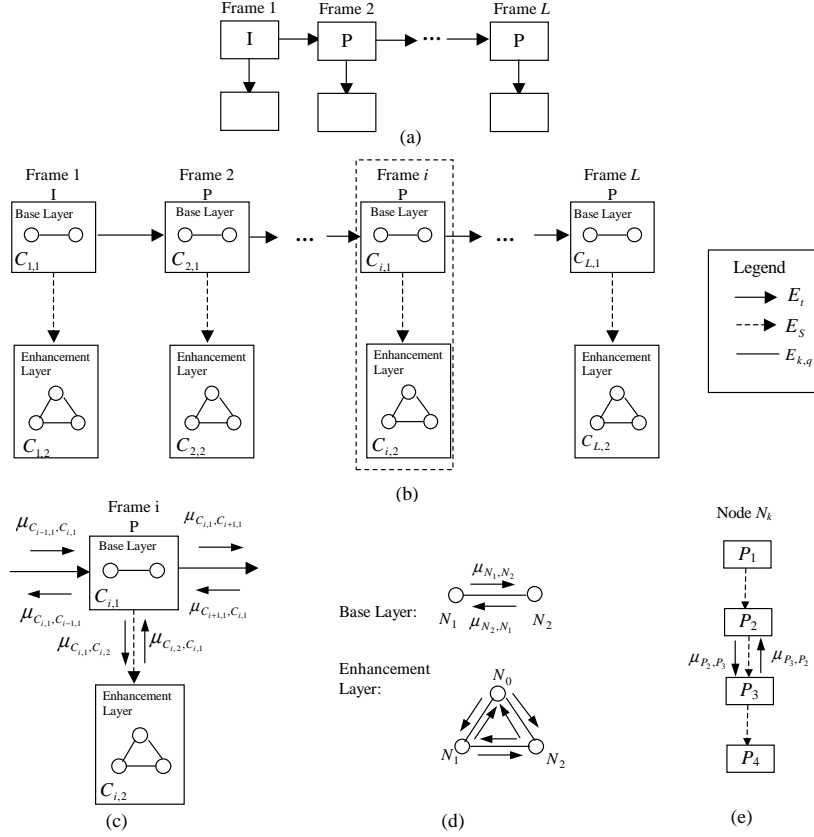
### 3. SOURCE MODEL: DIRECTED ACYCLIC HYPERGRAPH (DAHG)

In previous research [8–10], the source dependencies between a group of data units are modeled as a directed acyclic graph (DAG), in which each vertex represents a data unit, and each edge directing from data unit  $i$  to data unit  $j$  indicates the decoding dependence of  $j$  on  $i$ . That means data unit  $j$  can only be decoded if  $i$  is received and decoded. Fig. 3(a) shows a DAG of a layered coding system containing a group of I-P frames, with each frame having a base layer and an enhancement layer. Though this graph model leads to a simple and accurate representation for LC system, it cannot represent multiple descriptions in MDC or MDLC system. Since multiple descriptions of each layer in the MDLC system have no dependence relation, we cannot simply connect them by a directed edge as is done in a DAG.

Here we introduce a new model called Directed Acyclic HyperGraph (DAHG) to present the relationship between different video data units with multiple descriptions. A DAHG is like a normal DAG, but each vertex is composed of a clique which contains a set of nodes and every pair of nodes are connected by an undirected edge. Each clique  $C_{ij}$  corresponds to the  $j$ th layer of the  $i$ th frame which contains multiple descriptions, and each node  $N_k$  of the clique corresponds to the  $k$ th description. There are two kinds of edges in a DAHG: directed edges indicate a dependence relation, and undirected edges indicate a redundancy relation.

Fig. 3(b) shows an example DAHG for the MDLC system we use in this paper. Each frame  $i$  contains a base layer clique  $C_{i1}$  and an enhancement layer clique  $C_{i2}$ . A directed edge is used to represent both the SNR dependence between  $C_{i1}$  and  $C_{i2}$ , and the temporal dependence between two cliques of different frames if they are temporally dependent. Each base layer clique  $C_{i1}$  has two nodes  $N_1$  and  $N_2$ , corresponding to the two base layer descriptions of  $BL_1$  and  $BL_2$ . Similarly, there are three nodes in the enhancement layer clique  $C_{i2}$  corresponding to  $EL_1$ ,  $EL_2$  and  $EL_3$ . In MPEG-4 FGS scheme which is used in this paper, the enhancement layer is further divided into several sublayers. Those sublayers are connected by directed edges as shown in Fig. 3(e).

Thus a DAHG contains cliques, nodes and sublayers, where a clique contains several nodes, and each node may contain several sublayers as well. There are two kinds of dependencies between cliques: temporal dependency



**Figure 3.** Examples of DAG and DAHG. (a) A DAG for a LC system. (b) Static connections of a DAHG based on the proposed MDLC system for a group of I-P frames. (c) Message passing between cliques. (d) Message passing between nodes inside a clique. (e) Message passing between layered packets inside a node.  $E_t$ : temporal dependence,  $E_s$ : SNR dependence,  $E_{k,q}$ : sibling,  $\mu_{i,j}$ : message passing from  $i$  to  $j$ .

and SNR dependency. A clique  $\mathcal{C}$  is called a *parent* of clique  $\mathcal{C}'$  if there is a directed edge pointing from  $\mathcal{C}$  to  $\mathcal{C}'$ . Then  $\mathcal{C}'$  is called a *child* of  $\mathcal{C}$ .  $\mathcal{C}'$  can be decoded as long as at least one node of  $\mathcal{C}$  is decoded. The nodes inside a clique have redundancy between them, and they are called *siblings* of each other with undirected edges connecting them. The sublayers of a node also have SNR dependence among them. A parent of a sublayer is then defined as the one which has a directed edge pointing to it. We write  $i \prec j$  if object  $i$  is a parent of  $j$ ,  $i \succ j$  if  $i$  is a child of  $j$ , and  $i \sim j$  if they are siblings.

Each object in DAHG has certain quantities associated. Some of these quantities are constants determined in the encoding stage. The distortion  $D_{\mathcal{N}}$  of node  $\mathcal{N}$  is measured at the encoder by the distortion reduction of the reconstructed signal when this node is decoded. If an enhancement layer node contains several sublayers, each sublayer  $\mathcal{L}$  contributes to some distortion reduction  $D_{\mathcal{L}}$ . The ratio between  $D_{\mathcal{L}}$  and  $D_{\mathcal{N}}$  is denoted by  $\rho_{\mathcal{L}}$ , which shows how important this sublayer is in terms of the distortion. The redundancy  $I(k, q)$  between nodes  $\mathcal{N}_k$  and  $\mathcal{N}_q$  is measured as the difference between  $D_{\mathcal{N}_k} + D_{\mathcal{N}_q}$  and the actual distortion reduction when both nodes are received. That is meaningful because two descriptions, which have redundant data, cannot achieve the same distortion reduction as two uncorrelated packets with the same rate. In addition runtime status information is also stored. The impact of each object to others on an event is represented by messages passing along the edges in the graph as shown in Fig. 3(c)-(e).

For a packet-switching network, all the data are packetized before transmission. The FGS coding technique allows us to truncate the enhancement bitstream into any number of bits within each frame, to provide partial enhancement proportional to the number of bits decoded for each frame [2]. Therefore we can split the en-

hancement stream of each frame into sublayers at any bit position. Thus without loss of generality, we packetize each sublayer into a single packet. When the base layer is split into several packets, each packet is assigned a distortion value proportional to its data size. However it should be clear that a base layer node can only be decoded when all its packets are received. For simplicity, we order all the packets in a frame, and the  $j$ th packet of the  $i$ th frame is denoted by  $\mathcal{P}_{ij}$ . For each packet  $\mathcal{P}_{ij}$ , we are interested in the following parameters [9]: (1) its size  $r_{ij}$ , (2) its distortion value  $D_{ij}$  and distortion ratio  $\rho_{ij} = \frac{D_{ij}}{D_{\mathcal{N}}}$ , and (3) its playback deadline  $t_i$  defined as the time when the  $i$ th frame is to be played at the receiver.

## 4. SCHEDULING ALGORITHM WITH DAHG

### 4.1. Problem Formulation

In [8, 9], the overall distortion of a frame is calculated as the initial distortion (i.e., the distortion if no packets of the frame are decoded) less the sum of the distortion values over all packets that are decoded on time. However, in a MDLC system the actual distortion reduction due to a packet  $\mathcal{P}_{ij}$  is normally less than the measured distortion  $D_{ij}$ , because the decoder may have received other packets which have redundant information with this packet. Let  $I(\mathcal{P}_{ij}, \mathcal{P}_{ik})$  denote the redundancy between packets  $\mathcal{P}_{ij}$  and  $\mathcal{P}_{ik}$ , which can be calculated from the redundancy of their corresponding nodes. Let  $a_{ij}$  be the indicator function that is 1 if  $\mathcal{P}_{ij}$  arrives at the receiver on time and decoded, and is 0 otherwise. Thus the playback distortion  $D_v$  of a video sequence containing  $N$  frames, with  $M_i$  packets in frame  $i$ , can be expressed as

$$D_v = \bar{D}_v - \sum_{i=1}^N \sum_{j=1}^{M_i} a_{ij} \left\{ D_{ij} - \sum_{\mathcal{P}_{ik} \in S_{ij}} a_{ik} I(\mathcal{P}_{ij}, \mathcal{P}_{ik}) \right\} \quad (1)$$

where  $\bar{D}_v$  is the distortion of the whole video sequence if no packets are decoded. Our problem is actually a rate-distortion optimization problem to minimize  $D_v$  within a rate budget.

Given a set of candidate packets  $\mathcal{G}$ , which contains the packets from different frames, descriptions, layers, and possibly some retransmission packets, we want to design a scheduling algorithm which selects a group of packets to send under the constraint of rate budget, in such a way that the overall redundancy can match the channel behavior. A MDLC system can behave more like LC or MDC depending on the actual channel condition.

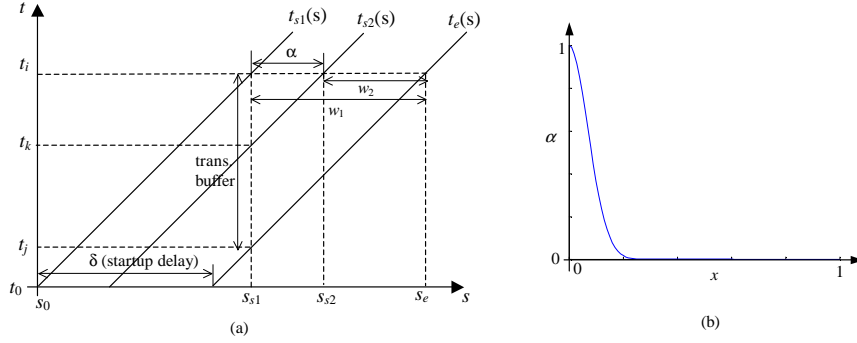
The playback distortion  $D_{\mathcal{G}}$  of the packet set  $\mathcal{G}$  can be expressed similar to (1) as

$$D_{\mathcal{G}} = \bar{D}_{\mathcal{G}} - \sum_{\mathcal{P}_{ij} \in \mathcal{G}} a_{ij} \left\{ D_{ij} - \sum_{\mathcal{P}_{ik} \in S_{ij}(\mathcal{G})} a_{ik} I(\mathcal{P}_{ij}, \mathcal{P}_{ik}) \right\} \quad (2)$$

Where  $\bar{D}_{\mathcal{G}}$  is the initial distortion of the packet set  $\mathcal{G}$ ,  $S_{ij}(\mathcal{G}) = \{\mathcal{P}_{ik} \mid j \leq k \leq M_i, \mathcal{P}_{ik} \in \mathcal{G}, \mathcal{P}_{ik} \text{ contains redundant data that are with } \mathcal{P}_{ij}\}$ . Given a set of packets  $\mathcal{G}$  and a channel model (channel bandwidth  $C$ , packet loss rate  $\epsilon$ , RTT and start-up delay  $\delta$ ), we want to design a runtime scheduling algorithm to minimize  $D_{\mathcal{G}}$ .

### 4.2. Double Time Window Control

Our double time window control scheme, shown in Fig. 4, is an extension of the method described in [8]. To simplify the discussion, we consider the case that frame rates are constant and the same at both encoder and decoder. Therefore, we can assume the end-to-end delay for each frame will be constant and equal to the initial start-up delay  $\delta$ . First we concentrate on the two straight lines  $t_{s1}(s)$  and  $t_e(s)$  shown in Fig. 4(a). These two lines describe the traditional time window control. Each frame has a limited window of transmission opportunity from  $t_{s1}(s)$  to  $t_e(s)$ . The  $i$ th frame can only be scheduled to be sent during the time period  $[s_{s1}, s_e]$ .  $s_e$  is decided by the frame's playback deadline  $t_i$  since any frame packets sent after this time cannot arrive at the receiver on time.  $s_{s1}$  indicates the earliest time that those packets can be sent. Define  $w_1$  as the window of transmission opportunity between  $s_{s1}$  and  $s_e$ . A large window size gives a frame more opportunities to be sent and may allow several retransmissions. However, a large window size also implies a long end-to-end delay, which may not be allowable in many real-time video applications. The vertical gap between  $t_{s1}(s)$  and  $t_e(s)$  decides the time window of the transmission buffer at a given time  $s$ ; any packet whose playback deadline lies in this time window



**Figure 4.** Double time window control. (a) Double time window.  $w_1$ : the window of transmission opportunity for  $BL_1$  and  $EL_1$ ;  $w_2$ : the window of transmission opportunity for  $BL_2$ ,  $EL_2$  and  $EL_0$ ;  $\alpha$ : the distance factor between the 1st and 2nd starting lines. (b) the  $\alpha$  function used in the proposed MDLC system.

has the opportunity to be selected to be sent at time  $s$ . For example in Fig. 4(a), the transmission buffer at time  $s_{s1}$  contains those packets whose frame index is between  $j$  and  $i$ .

In our MDLC system more than one description is encoded for each frame, and the choice of descriptions to be sent depends on the actual channel environment. Therefore we introduce an adaptive double time window control scheme, which gives each description a different window of transmission opportunity by adding a second starting line  $t_{s2}(s)$  in Fig. 4(a). The basic idea is that when the channel is more favorable to a LC system (e.g., low error-rate ( $\epsilon$ ) or very short RTT), we move the line  $t_{s2}(s)$  toward  $t_e$  such that there is little chance to send the second description. In this case MDLC will behave like a LC system. When  $t_{s2}(s)$  overlaps with  $t_e$ , the MDLC system becomes a pure LC system. On the other hand when  $\epsilon$  becomes very large or RTT is so long that retransmission is impossible, the MDC approach is now better than a traditional LC, and thus  $t_{s2}(s)$  moves left toward  $t_{s1}(s)$ . When  $t_{s2}(s)$  reaches  $t_{s1}(s)$ , different descriptions have the same transmission opportunity. Let  $\alpha$  be the distance factor defined as  $1 - w_2/w_1$ . The  $\alpha$  function depends on several factors including channel bandwidth ( $C$ ), error rate ( $\epsilon$ ), RTT ( $RTT$ ), and end-to-end delay ( $\delta$ ). Here we use the following function as shown in Fig. 4(b)

$$\alpha = e^{-\frac{x^2}{b}}, \quad \text{where } x = \epsilon^n \bar{C} \quad (3)$$

$n$  is the number of possible transmissions (including retransmissions if the packet is reported lost after the first transmission), and can be calculated as  $n = \frac{w_1}{RTT} = \frac{\delta}{RTT}$ . Thus  $\epsilon^n$  denotes the packet loss probability.  $\bar{C}$  is the channel bandwidth normalized by the bandwidth required ( $C_{d1}$ ) for transmitting one description of base layer and enhancement layer, i.e.  $\bar{C} = \min(1, \frac{C}{C_{d1}})$ . Intuitively, when the channel bandwidth is larger than that required for one description, it is better to use the extra bandwidth to send the other descriptions to provide more reliability, rather than to waste the bandwidth. The constant  $b$  in (3) is used to adjust the speed of decrease of  $\alpha$  for a particular MDLC system. For all the experiments shown in Section 5 we use the same  $b$ , and our experience shows that the system performance is not very sensitive to  $b$ .

### 4.3. Runtime Scheduling Algorithm

Recall that (2) is calculated on a particular set of packets  $\mathcal{G}$ , which is determined by the proposed double time window control scheme. At any given time, we use a greedy approach to select the most important packet from  $\mathcal{G}$  to send instead of an optimal scheduling algorithm (e.g., as used in [8] for a LC system). Our proposed greedy approach is based on a *a priori* channel model, the past transmission history of the packets and the feedback from the receiver. This paper will only discuss the application of the proposed MDLC system and scheduling algorithm for I-frame video sequence. The scheduling algorithm designed for a system with temporal prediction is still under development.

The following observations of our MDLC system are used in the scheduling algorithm:

**Observation 1** A clique can be decoded only if all of the input edges from its parents are activated, which means at least one of the descriptions of any of its parents must be decodable in order to decode this clique.

**Observation 2** The two descriptions are symmetric. Therefore, we only discuss  $BL_1$  and  $EL_1$  here, and the corresponding equations for  $BL_2$  and  $EL_2$  can be derived in a similar way.

**Observation 3** The cross-description decoding of  $EL_2$  on  $BL_1$  or  $EL_1$  on  $BL_2$  is ignored since the information added by the cross enhancement layer is very small.

**Observation 4** The enhancement layer uses an MPEG-4 FGS coding scheme.

### 4.3.1. Expected Runtime Packet Distortion

To estimate the importance of a packet  $\mathcal{P}_{ij}$ , we introduce an expected runtime distortion  $\hat{d}_{ij}$  for each packet, as originally proposed by [9] for a LC system. In a MDLC system, we need to take into account several factors including: (1) data dependencies between packets and cliques; (2) data correlation between nodes; (3) channel condition; and (4) end-to-end delay constraint. Since the delay constraint is already considered in the double time window control scheme, we only focus on the first three factors here. This distortion is calculated as the possible distortion reduction contributed by a given packet if it were received. Then the packet which has the largest expected runtime distortion will be scheduled to send next.

First, we fix some notations: (1)  $X_{ij}$  - parameter  $X$  for a packet  $\mathcal{P}_{ij}$ .  $X$  can be distortion measure ( $d$  or  $D$ ), probability of receiving a packet ( $P$ ) or packet distortion ratio ( $\rho$ ). (2)  $X_c^{(l)}$  - parameter  $X$  for a clique in the  $l$ th layer,  $l = 1$  for base layer,  $l = 2$  for enhancement layer; (3)  $X_{\mathcal{N}_k}^{(l)}$  - parameter  $X$  for the  $k$ th node in a clique of the  $l$ th layer. We ignore the frame index in the notation of clique and node parameters, which will not lead confusion because there is no dependency between I-frames. For a clearer presentation, we further use two operators  $\circ$  and  $\cdot$  to represent the inner product and element-by-element product of two vectors, respectively.

The expected runtime distortion of a base layer packet  $\mathcal{P}_{ij}$  can be expressed as,

$$\hat{d}_{ij} = \rho_{ij} \cdot \hat{d}_{\mathcal{N}_k}^{(1)} \quad (4)$$

where  $\rho_{ij} = \frac{D_{ij}}{D_{\mathcal{N}_k}^{(1)}}$  is the packet's distortion ratio, and  $\hat{d}_{\mathcal{N}_k}^{(1)}$  is the expected runtime distortion of its corresponding node  $\mathcal{N}_k$ . Recall that an enhancement layer packet is decoded differently based on the decoding scenarios of the base layer. Here we consider three scenarios: (1) both  $BL_1$  and  $BL_2$  are received; (2) only  $BL_1$  is received; and (3) only  $BL_2$  is received. We write the probability of occurrence of each decoding scenario as a vector  $\mathbf{P}_s = [P_{\mathcal{N}_1}^{(1)} P_{\mathcal{N}_2}^{(1)} \quad P_{\mathcal{N}_1}^{(1)}(1 - P_{\mathcal{N}_2}^{(1)}) \quad (1 - P_{\mathcal{N}_1}^{(1)})P_{\mathcal{N}_2}^{(1)}]$ , where  $P_{\mathcal{N}_k}^{(1)}$  is the probability of receiving the base layer node  $\mathcal{N}_k$ . Accordingly, we use a vector  $\mathbf{d}_{ij}$  to represent the runtime distortion reduction of an enhancement layer packet, with each element corresponding to one scenario listed above. Since the packet can only be decoded if all of its parent packets in the same node have been decoded, we express  $\mathbf{d}_{ij}$  as

$$\mathbf{d}_{ij} = \left\{ \rho_{ij} \prod_{\mathcal{P}_{il} \in \mathcal{A}_{ij}} P_{il} \right\} \cdot \hat{\mathbf{d}}_{\mathcal{N}_k}^{(2)} \quad (5)$$

where  $\mathcal{A}_{ij} = \{\mathcal{P}_{il} | \mathcal{P}_{il} \prec \mathcal{P}_{ij}, \mathcal{P}_{il} \text{ and } \mathcal{P}_{ij} \text{ are in the same node}\}$  is the parent set of  $\mathcal{P}_{ij}$ , and  $\prod_{\mathcal{P}_{il} \in \mathcal{A}_{ij}} P_{il}$  is a scalar. Please note the packet distortion ratio  $\rho_{ij}$  and runtime distortion  $\hat{\mathbf{d}}_{\mathcal{N}_k}^{(2)}$  are vectors due to multiple decoding scenarios. If packet  $\mathcal{P}_{ij}$  has children packets, we also need to take into account the possible distortion reduction by its children if  $\mathcal{P}_{ij}$  is received. Then (5) becomes

$$\mathbf{d}_{ij} = \left\{ \rho_{ij} \prod_{\mathcal{P}_{il} \in \mathcal{A}_{ij}} P_{il} + \sum_{\mathcal{P}_{il} \in \mathcal{B}_{ij}} \left[ \rho_{il} \prod_{\mathcal{P}_{im} \in \mathcal{A}_{il}, m \neq j} P_{im} P_{il} \right] \right\} \cdot \hat{\mathbf{d}}_{\mathcal{N}_k}^{(2)} \quad (6)$$



where  $\mathcal{B}_{ij} = \{\mathcal{P}_{il} | \mathcal{P}_{il} \succ \mathcal{P}_{ij}, \mathcal{P}_{il} \text{ and } \mathcal{P}_{ij} \text{ are in the same node}\}$  is the child set of  $\mathcal{P}_{ij}$ . Then the expected runtime distortion  $\hat{d}_{ij}^{(2)}$  of packet  $\mathcal{P}_{ij}$  is the average distortion reduction under all the decoding scenarios:

$$\hat{d}_{ij} = \mathbf{P}_s \circ \mathbf{d}_{ij} \quad (7)$$

Now we consider the expected runtime distortion of node  $\mathcal{N}_k$ . It reflects the importance of a node by calculating the possible distortion reduction if all the packets of this node are received. It is used in (4) and (6) to calculate the expected runtime distortion for a particular packet. The node's expected runtime distortion is affected by two factors: the redundancy introduced by its sibling nodes, and the decoding effect on its children clique if this node is inside a base layer clique. We consider the enhancement layer node first. Based on Observation (3),  $\mathcal{N}_0$  (i.e.,  $EL_0$ ) can be decoded under any of the three decoding scenarios discussed before,  $\mathcal{N}_1$  (i.e.,  $EL_1$ ) can be decoded in the first and second scenarios, and  $\mathcal{N}_2$  (i.e.,  $EL_2$ ) can be decoded for the first and third scenarios. Therefore, considering the redundancy introduced by its siblings, the runtime distortions of  $\mathcal{N}_0$  and  $\mathcal{N}_1$  can be written as

$$\begin{aligned} \hat{\mathbf{d}}_{\mathcal{N}_0}^{(2)} &= \mathbf{D}_{\mathcal{N}_0}^{(2)} - \rho_{\mathcal{N}_1}^{(2)} \times \mathbf{I}^{(2)}(0, 1) - \rho_{\mathcal{N}_2}^{(2)} \times \mathbf{I}^{(2)}(0, 2) \\ \hat{\mathbf{d}}_{\mathcal{N}_1}^{(2)} &= \mathbf{D}_{\mathcal{N}_1}^{(2)} - \rho_{\mathcal{N}_0}^{(2)} \times \mathbf{I}^{(2)}(0, 1) \end{aligned} \quad (8)$$

where  $\mathbf{D}_{\mathcal{N}_k}^{(2)}$  is the measured distortion of node  $\mathcal{N}_k$  at the encoder (refer to Section 3),  $\mathbf{I}^{(2)}(k, q)$  is a vector containing the redundancy between nodes  $\mathcal{N}_k$  and  $\mathcal{N}_q$  under each decoding scenario, and  $\rho_{\mathcal{N}_k}^{(2)}$  is the runtime distortion ratio of node  $\mathcal{N}_k$ . When a node is split into different packets, the calculation of its runtime distortion ratio is based on the delivery status of each packet. Since the FGS enhancement layer does not require all packets to be received to decode,  $\rho_{\mathcal{N}_k}^{(2)}$  is calculated as the ratio between the actual distortion reduction of received packets and the total distortion reduction (i.e., when all packets of this node are received). Thus it can be expressed as the sum of the distortion ratio of those packets which can be decoded at the receiver:

$$\rho_{\mathcal{N}_k}^{(2)} = \sum_{\mathcal{P}_{ij}} \rho_{ij} \prod_{\mathcal{P}_{il} \in \mathcal{A}_{ij}} P_{il} P_{ij} \quad (9)$$

The redundancy introduced by a node to its siblings is proportional to its runtime distortion ratio.

For a base layer node, we need to consider both factors as mentioned before. Without loss of generality we assume  $\mathcal{N}_1$  in the following discussion. Let  $\mathbf{d}_{\mathcal{C}}^{(2)}$  be the distortion vector of the enhancement layer clique for different decoding scenarios. Then the total distortion  $d_{\mathcal{C}}^{(2)}$  of the clique averaging over all the decoding scenarios is  $\mathbf{P}_s \circ \mathbf{d}_{\mathcal{C}}^{(2)}$ . Obviously this distortion depends on the probability of decoding scenarios  $\mathbf{P}_s$ . Since  $\mathbf{P}_s$  changes with the probability of receiving base layer nodes, the runtime distortion of a base layer node should consider its effect on the children cliques, and thus can be expressed as

$$d_{\mathcal{N}_1}^{(1)} = [D_{\mathcal{N}_1}^{(1)} - P_{\mathcal{N}_2}^{(1)} \cdot I^{(1)}(1, 2)] + [\mathbf{P}_{s'} \circ \mathbf{d}_{\mathcal{C}}^{(2)} - \mathbf{P}_s \circ \mathbf{d}_{\mathcal{C}}^{(2)}] \quad (10)$$

where  $\mathbf{P}_{s'}$  is the probability vector of the decoding scenarios if the node is received, and  $\mathbf{P}_s$  corresponds to the case when the node is not received. Note that a base layer node can only be decoded when all its packets are received. Therefore, the redundancy introduced by a base layer node to its sibling is proportional to its probability of being received.

A clique has two parameters to consider. The first is the probability  $\mathbf{P}_{\mathcal{C}}$  of receiving this clique. It is a probability vector with the  $k$ th element corresponding to the probability of receiving its  $k$ th node. For the enhancement layer case, since the bitstream can be truncated at any bit position, we need to use the runtime distortion ratio  $\rho_{\mathcal{C}}$  instead. The  $k$ th element of  $\rho_{\mathcal{C}}$  is the runtime distortion ratio of its  $k$ th node. The second is the clique's runtime distortion. It reflects the current distortion reduction contributed by this clique based on the transmission history of all its nodes. By considering the delivery status of its nodes and their redundancy, the runtime distortion of a base layer clique can be calculated as

$$d_{\mathcal{C}}^{(1)} = p_{\mathcal{N}_1}^{(1)} D_{\mathcal{N}_1}^{(1)} + p_{\mathcal{N}_2}^{(1)} D_{\mathcal{N}_2}^{(1)} - \min(p_{\mathcal{N}_1}^{(1)}, p_{\mathcal{N}_2}^{(1)}) \cdot I^{(1)}(1, 2) \quad (11)$$

We can also derive a similar equation for an enhancement layer clique, which is omitted in the paper.

### 4.3.2. Message Updating Rules With DAHG

From the above discussion, we know that the expected runtime distortion of a packet is affected by the transmission history of the other packets, nodes and cliques. The impact of each object to others on an event is represented by messages passing along the edges in the graph as shown in Fig. 3(c)-(e). The events include: (1) sending a packet, (2) receiving an ACK on a packet, and (3) receiving a NAK on a packet. We define the *status* of an object as its probability of being received, which is updated each time when an event occurs. The probability  $P_{ij}$  of receiving a packet is updated as follows [9]:

$$P_{ij} = \begin{cases} 0 & \text{if packet } \mathcal{P}_{ij} \text{ has not been sent,} \\ 0 & \text{if there is a NAK on packet } \mathcal{P}_{ij}, \\ 1 - \epsilon & \text{if packet } \mathcal{P}_{ij} \text{ is sent without receiving ACK or NAK,} \\ 1 & \text{if there is a ACK on packet } \mathcal{P}_{ij}. \end{cases} \quad (12)$$

The probabilities of receiving the packet's corresponding node and clique can be updated accordingly. (9) shows an example for an enhancement layer node. Please note we use  $\rho_{\mathcal{N}_k}^{(2)}$  instead of  $P_{\mathcal{N}_k}^{(2)}$  in the equation because FGS stream allows partial decoding. The expected runtime packet distortion shown in (4) and (7) can then be expressed as the functions of its probability of being received plus the messages from other objects.

Now we define the *messages* between packets, nodes and cliques. A message has different contents based on the message direction. We use  $\mu_{i,j}$  to denote the message passing from object  $i$  to  $j$ . The messages between packets are defined here only for the sublayers inside an enhancement layer node since the base layer packets have no dependence between each other. (6) shows the interdependence between enhancement layer packets, from which we could define the messages between packets (assume the packets are ordered from the lowest sublayers to the highest sublayers, i.e.,  $\mathcal{P}_{ij} \prec \mathcal{P}_{i,j+1}$ ):

$$\begin{aligned} \mu_{\mathcal{P}}(j, j+1) &= \begin{cases} P_{ij} & \text{if } \mathcal{P}_{ij} \text{ is the first packet of the node (i.e., without parents),} \\ \mu_{\mathcal{P}}(j-1, j) \cdot P_{ij} & \text{otherwise.} \end{cases} \\ \mu_{\mathcal{P}}(j, j-1) &= \begin{cases} \rho_{ij} P_{ij} & \text{if } \mathcal{P}_{ij} \text{ is the last packet of the node (i.e., without children),} \\ [\mu_{\mathcal{P}}(j+1, j) + \rho_{ij}] \cdot P_{ij} & \text{otherwise.} \end{cases} \end{aligned} \quad (13)$$

Note that  $\mu_{\mathcal{P}}(j, j+1)$  is a probability while  $\mu_{\mathcal{P}}(j, j-1)$  is actually a distortion ratio vector. The messages between node siblings are defined as their probabilities of being received for the base layer nodes, and the runtime distortion ratio for the enhancement layer nodes,

$$\mu_{\mathcal{N}}(k, q) = \begin{cases} P_{\mathcal{N}_k}^{(1)} & \text{base layer,} \\ \rho_{\mathcal{N}_k}^{(2)} & \text{enhancement layer.} \end{cases} \quad (14)$$

The messages between a base layer clique and an enhancement layer clique are defined as

$$\begin{aligned} \mu_{\mathcal{C}}(1, 2) &= P_{\mathcal{C}}^{(1)} \\ \mu_{\mathcal{C}}(2, 1) &= \mathbf{d}_{\mathcal{C}}^{(2)} \end{aligned} \quad (15)$$

Then we can replace the quantities described in Section 4.3.1 with the corresponding messages. For example, (6) can be expressed in messages as

$$\mathbf{d}_{ij} = \left\{ \mu_{\mathcal{P}}(j-1, j) \cdot [\rho_{ij} + \mu_{\mathcal{P}}(j+1, j)] \right\} \cdot \hat{\mathbf{d}}_{\mathcal{N}_k}^{(2)} \quad (16)$$

and (9) becomes

$$\rho_{\mathcal{N}_k}^{(2)} = \sum_{\mathcal{P}_{ij}} \rho_{ij} \cdot \mu_{\mathcal{P}}(j-1, j) \cdot P_{ij} \quad (17)$$

The other equations are very easy to modify just by replacing the quantities with messages, and so we omit their discussion here.

The message updating rules can be summarized as following steps:

0. Initialization: For all the packets just added to the transmission buffer (controlled by the double time window control scheme), initialize their status as (12);
1. Updates at the packet level: When an event occurs on a packet (referred to as the *starting packet*), update its status as (12). This packet will generate two messages, with one passed to its parent and the other to its child, if there is any. When a parent packet receives the message, it first updates itself and further passes information to its own parent. This process proceeds recursively till the root packet. Similar process proceeds along the children packets till the leaf packet;
2. Updates at the node level: The starting packet notifies its corresponding node of the update. Upon receiving the notification, the node updates its status and further sends messages based on (14) to its siblings to update their expected runtime distortion according to (10) and (8);
3. Updates at the clique level: After the updates at the node level finish, the corresponding clique will update its status and pass the message to either its child clique (in base layer case) or parent clique (in enhancement layer case). The parent or child clique will further update its runtime distortion.

The above message passing steps will ensure the status of each object in a DAHG updated on each event. At timestep  $t$ , the scheduler calculates the expected runtime distortion for each candidate packet in the transmission buffer, and selects the one with the greatest value to send. One thing to point out is that the status update concept used in this paper is originally from Miao and Ortega in their fast scheduling algorithm for a LC system [14]. Here we further introduce messages and message updating rules to represent the impact of each object to others on an event.

## 5. EXPERIMENTAL RESULTS

The proposed MDLC system is applicable to diverse applications and a wide range of channel conditions. To evaluate its performance, we set up a delivery system to simulate the on-line transmission with different schemes: MDLC, LC and MDC. We use MPEG-4 FGS coding scheme for the LC simulation to be comparable with the proposed MDLC system. The MDC system uses the same multiple description generating method as our MDLC for the base layer. All of these schemes use a real-time scheduler to deliver packets. The proposed MDLC scheduling algorithm is easy to be modified to be compatible with a LC or MDC system. For a LC system, we just reduce the number of nodes in each clique to one. That means only one description exists for each layer, and this is equivalent to a LC system demonstrated in [9]. For a MDC system, we only create one base layer clique for each frame. In the simulations we use the video sequence *Stefan* with 250 frames packetized with a

**Table 1.** Video data rates (in KBits/sec) for different coding schemes.

Scheme	QP	BL(or BL1/BL2)	EL(or EL0/EL1/EL2)	Total
MDLC	12	174.28/174.22	724.21/869.43/869.91	2812.06
LC	12	302.33	724.2	1026.53
MDC1	5	311.24/311.38	0	622.62
MDC2	8	227.14/227.17	0	454.31
MDC3	12	174.28/174.24	0	348.52

fixed packet size of 512 bytes. The frame rate is 25 frames/second. Table 1 shows the video data rates after compression under each scheme we used in the simulation. The playback quality is measured by PSNR of the reconstructed video frames at the receiver. Fig. 5 shows the simulation results of the playback quality of both

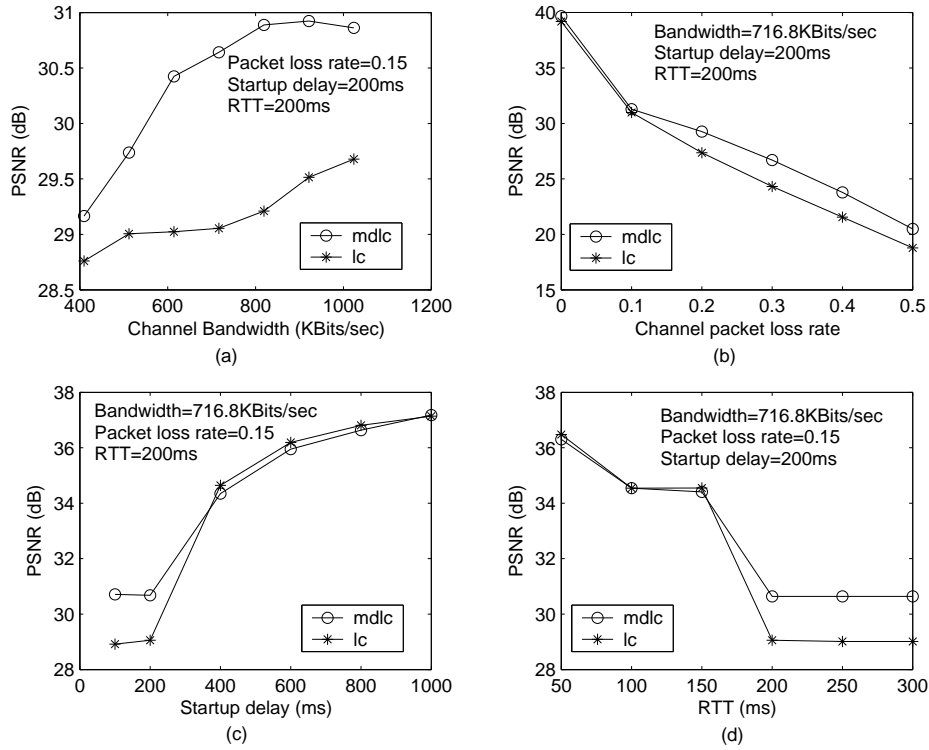


Figure 5. The comparison of PSNR between the proposed MDLC system and the LC system with various parameters.

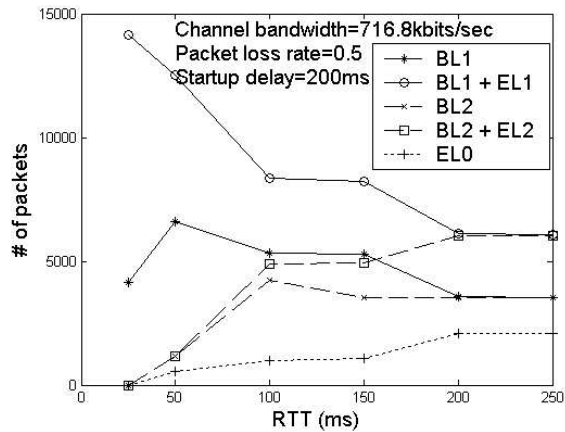
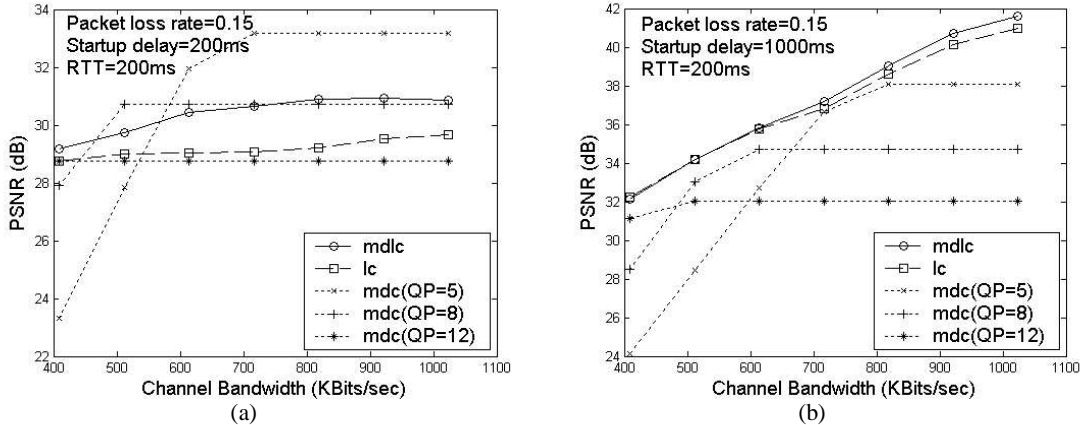


Figure 6. The change of the bandwidth allocation on different descriptions with various RTT.

MDLC and LC system with various parameters, such as channel bandwidth, packet loss rate, startup delay and RTT. The results show that by using the proposed MDLC system the playback quality improves about 2dB compared to the traditional LC system in the scenarios when a channel has large packet loss rate, long RTT or an application has stringent delay requirement. In the other scenarios the performance of MDLC system is close to that of the LC system, and the small gap between their performance is due to their different base layers and enhancement layers. From the above simulations, we can see that the MDLC system is adaptive in that it adaptively selects the packets to send to make the overall redundancy match the channel condition and provides



**Figure 7.** The comparison of PSNR between MDLC, LC and a set of MDC systems with various channel bandwidths in different startup delays: (a) startup delay = 200ms, (b) startup delay = 1000ms.

graceful degradation when the channel becomes worse. To illustrate it clearly we also plot the number of packets sent for each description during one transmission session with various RTT as shown in Fig. 6. It is clear to see that when the RTT is very short (i.e., the system has many chances to retransmit) Description 1 is dominant and the system is more like a LC system. When RTT increases, more and more packets from description 2 are sent and they eventually become similarly important. This is in part due to the double time window control.

We also describe experiments to compare the proposed MDLC system with a set of MDC schemes with different quantization steps shown in table 1. Fig. 7 shows the PSNR of different schemes with various channel bandwidths under different startup delay scenarios. In the low delay case the MDC scheme with quantization parameter 5 has better performance when the channel bandwidth is large. However it drops quickly in the case of low bandwidth. In the high delay case when there are retransmission opportunities, MDLC works much better than all of the MDC schemes, but about the same as LC. This shows that our proposed MDLC scheme can be applied in a wider range of network scenarios and application requirements.

## 6. CONCLUSION

In this paper we proposed a new approach, Multiple Description Layered Coding (MDLC), which combines the advantages of LC and MDC to provide reliable video communication over a wider range of network scenarios and application requirements. A new source model called Directed Acyclic Hypergraph (DAHG) is introduced to describe the relationship between the different data units of a MDLC system. Based on this model, a runtime scheduling algorithm with a double time window control scheme is also proposed for the MDLC system, to reduce its redundancy to match the channel behavior. Simulation results show that the proposed MDLC system provides more robust and efficient video communication for the real-time applications over lossy packet networks.

## REFERENCES

1. Y. Wang, J. Ostermann, and Y. Zhang, *Video processing and communications*, Prentice Hall, Upper Saddle River, New Jersey, 2002.
2. W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.* **11**, pp. 301–317, Mar. 2001.
3. V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.* **18**, pp. 74–93, Sep. 2001.
4. A. Reibman, H. Jafarkhani, M. Orchard, and Y. Wang, "Performance of multiple description coders on a real channel," in *Proc. ICASSP'99*, **5**, pp. 2415–2418, 1999.

5. 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. ICIP'00*, **2**, pp. 136 –139, 2000.
6. R. Singh, A. Ortega, L. Perret, and W. Jiang, "Comparison of multiple description coding and layered coding based on network simulations," in *Proc. SPIE VCIP'00*, 2000.
7. *MPEG-4 version 2 visual working draft rev. 3.0, N2202*, ISO/IEC JTC1/SC29/WG11, Mar. 1998.
8. P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *submitted to IEEE Trans. Multimedia*, Feb. 2001.
9. Z. Miao and A. Ortega, "Expected run-time distortion based scheduling for delivery of scalable media," in *12th International Packet Video Workshop*, Apr. 2002.
10. G. Cheung and W. Tan, "Directed acyclic graph based source modeling for data unit selection of streaming media over QoS networks," in *Proc. ICME'02*, 2002.
11. R. Zhang, S. Regunathan, and K. Rose, "Optimized video streaming over lossy networks with real-time estimation of end-to-end distortion," in *Proc. ICME'02*, **1**, pp. 861 –864, 2002.
12. A. Reibman, "Optimizing multiple description video coders in a packet loss environment," in *12th International Packet Video Workshop*, Apr. 2002.
13. D. Comas, R. Singh, A. Ortega, and F. Marques, "Unbalanced multiplied description video coding based on a rate-distortion optimization," *EURASIP Journal on Applied Signal Processing*, Nov. 2002.
14. Z. Miao and A. Ortega, "Fast adaptive media scheduling based on expected run-time distortion," in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, Nov. 2002.