Efficient Bandwidth Resource Allocation for Low-Delay Multiuser MPEG-4 Video Transmission

Guan-Ming Su and Min Wu
Department of Electrical and Computer Engineering, University of Maryland, College Park, USA

***Abstract*— In this paper, we propose an efficient bandwidth resource allocation algorithm with low delay and low fluctuation of quality to transmit multiple MPEG-4 Fine Granularity Scalability (FGS) video programs to multiple users. By exploring the variation in the scene complexity of each video program and jointly redistributing available system resources among users, our proposed algorithm provides low fluctuation of quality for each user and consistent quality among all users. Experimental results show that compared to a traditional look-ahead sliding-window approach, our scheme can achieve comparable perceptual quality and channel utilization at a much lower cost of delay, computation, and storage.**

I. Introduction

The capability of real-time transmission of video over network enables a number of emerging applications, which allows us to communicate and entertain from almost every corner of the world. In such applications as digital video on-demand service, broadband wireless video streaming and conferencing, and Direct Broadcast Satellite (DBS) service, multiple encoded video programs will be transmitted or relayed through a central server. The overall bandwidth of the outbound video streams is limited by the server's outbound communication capacity. To efficiently share critical resources and meet a set of quality of service (QoS) requirements, an important issue is for the server to determine how to allocate the bandwidth resource to each stream. There are two different strategies of resource allocation for multiple users, namely, a collection of single-user subsystems with independent static resource allocation for each user, and a joint dynamic resource allocation system [1]–[3]. The latter can leverage the variation of the content complexity in different video programs, aggregate the resource from all users into a common pool, and jointly allocate bandwidth resource with each user to achieve consistent perceptual quality to each other. In this paper, we study dynamic bandwidth resource allocation for a multiuser video transmission system.

Bandwidth resource allocation for streaming video involves several important issues. The first issue is the perceptual criteria. There are two types of visual quality concern: one is average mean-square-error (*aveMSE*) of all video frames, or its equivalent PSNR [4], [5]; and the other is the quality fluctuation, which can be measured by the mean absolute difference of the MSE (*madMSE*) between frames [5], [6]. Most prior work targeted optimizing one of the two measures. If the rate-distortion (R-D) characteristics of all video frames are identical, the bit rate allocated to each frame will be the same, leading to identical perceptual quality between frames and the above two measures can be simultaneously optimized [1]. In reality, however, a video has high variation in the R-D characteristics from scene to scene, making it difficult to optimize the average quality and the quality fluctuation at the same time. In this paper, we provide a general formulation and a real-time algorithm to reach a good tradeoff between these two quality criteria.

The second issue is the relationship between the video encoding/transmission rate and the corresponding visual quality. A highly scalable video codec is desirable since it provides flexibility and convenience in tuning to the desired visual quality and/or the desired bit rate. Recently, the Fine Granularity Scalability (FGS) coding [7] and Fine Granular Scalability Temporal (FGST) coding [8] have been added in MPEG-4 video coding standard. We adopt FGS codec in this work to allow convenient adjustment in video rate and quality.

Rate control for single user can be considered as a special case of multiuser bandwidth resource allocation and has been studied in the literature [9], [10]. To achieve high overall perceptual quality, rate control was formulated as an optimization problem in [4], [11]. These approaches are suitable for offline applications where the entire video content is known to the transmitter. Several rate control schemes using R-D model were proposed in [5], [12], [13]. A rate control algorithm employing a linear correlation model was proposed in [14]. Sliding window is a general approach that can be used to keep track and allocate system resources [6]. An online algorithm using a look-ahead sliding window to achieve constant perceptual quality was proposed in [1]. However, the look-ahead strategy requires an extra storage to store the look-ahead frames. Our studies show that to obtain a low fluctuation of quality, the window size should be no smaller than the size of one GOP, which leads to a nontrivial delay. For interactive real-time applications, the requirement of end-to-end delay is stringent. Furthermore, the computation complexity of the sliding window approach goes up with the increase of the window size. While extending this approach to a large-scale multiuser scenario, the required computation resources become formidable. This motivates us to investigate an efficient algorithm with low computation and storage demands.

The paper is organized as follows. We examine the R-D characteristics of FGS layer data in Section II. Section III discusses the single-user resource allocation algorithm, and Section IV presents the multi-user extension of the resource allocation algorithm. Experimental results are shown in Section V and conclusions drawn in Section VI.

II. FGS Rate-Distortion Model

Existing rate control schemes for a single-layer video stream often employ an intra-frame R-D model. The MPEG-4 FGS

This work was supported by the U.S. National Science Foundation under Award #CCR-0133704.

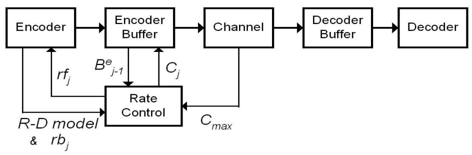


Fig. 1. Block Diagram of a single-user video streaming system

codec is a two-layer embedded scheme and its enhancement layer is encoded bit plane by bit plane. Previous studies in [1], [6] and our experiments show that a piecewise linear line is a good approximation to the R-D curve of FGS video at the frame level. The R-D function $D_j(r_j)$ can be interpolated using a set of R-D pairs (MSE_j^k,$Rate_j^k$), where MSE_j^k, $Rate_j^k$, and r_j denote the distortion after completely decoding the first k DCT bit planes, the corresponding bit rate, and the overall decoded bit rate (both base and FGS layers) for the j^{th} frame, respectively. Another useful property is that the R-D curves of FGS layer between two consecutive, predictively coded frames are similar when they are within the same scene.

III. Resource Allocation for Single User

To facilitate the investigation of the resource allocation problem in a multi-user system, we first study in this section a special case that concerns only a single user in the system.

Figure 1 illustrates a typical streaming video system. The encoder consists of a base layer encoder and a FGS layer encoder. We discretize the time line into F time slots per second, where F is the video frame rate. For the simplicity of system design and providing a primitive constant quality, we set a large fixed quantization step for all frames in the base layer codec and only perform the rate control for the FGS layer. Denote the base layer rate as rb_j, which must be sent at the j^{th} time slot to ensure the baseline quality. The FGS encoder encodes the bit planes of the residue. Both encoders analyze the R-D characteristics of the incoming video frame and pass the necessary information to the rate control module. After the rate control module determines the amount of FGS data to be transmitted, the encoded base layer and the truncated FGS layer bitstream with rate rf_j are moved to the encoder buffer. The channel then delivers video bitstream from the encoder buffer to the decoder buffer. Here we assume that the channel has a maximum rate for reliable transmission, C_{max}, although it is not necessarily in its full load all the time. The amount of channel transmission rate, C_j, is also determined by the rate control module. As with many existing works in the literature, for simplicity, we assume that the transmission delay of every packet is fixed at d^c time slots [4], [11]. The decoder fetches data from the decoder buffer, decodes it, and displays each decompressed video frame at its desired instant. By examining this system flow, we can see the major task of the rate control module is to determine rf_j and C_j.

A. System Constraints

There are three constraints imposed in this system, which has been studied in the literature [9]. The first constraint is to prevent the encoder buffer of a limited size from overflow. The dynamics of the encoder buffer can be expressed as

$$B_j^e = \max\{B_{j-1}^e + rb_j + rf_j - C_j, 0\} \leq B_{max}^e, \qquad (1)$$

where B_j^e is non-negative and describes the occupancy of encoder buffer, and B_{max}^e the maximal size of encoder buffer. In addition, the FGS rate should be non-negative. For a given C_j, we can rearrange (1) as a constraint for rf_j :

$$0 \leq rf_j \leq B_{max}^e - B_{j-1}^e - rb_j + C_j. \qquad (2)$$

The next constraint is on the channel transmission rate,C_j:

$$0 \leq C_j \leq C_{max}. \qquad (3)$$

The third constraint is on the occupancy of the decoder buffer, which should neither overflow nor underflow. We assume that the decoder fetches all the data that belongs to the next frame from the decoder buffer and decodes it within one time slot. In addition, we assume playback buffering of d^d frames, i.e., the first d^d frames are received and stored in the decoder buffer before the playback is started. The total end-to-end delay from the encoder buffer through the channel and decoder buffer to the decoder is thus $d^c + d^d$ frames delay. The decision on how much data is sent into the channel at the j^{th} time slot will directly affect the decoder buffer occupancy at the $(j+d^c)^{th}$ time slot. Denote the decoder buffer occupancy as B_j^d, and the maximal size of decoder buffer as B_{max}^d. Denote $r_j = rb_j + rf_j$. The constraint on the decoder buffer at the $(j+d^c)^{th}$ time slot is expressed as

$$B_{j+d^c}^d = B_{j+d^c-1}^d + C_j - r_{j-d^d} \in [0, B_{max}^d]. \qquad (4)$$

Combining (3) and (4), we arrive at the following constraint for the channel transmission rate C_j

$$\max\{r_{j-d^d} - B_{j+d^c-1}^d, 0\} \leq C_j \leq \qquad (5)$$
$$\min\{B_{max}^d - B_{j+d^c-1}^d + r_{j-d^d}, C_{max}\}.$$

In summary, inequalities (2) and (5) are the fundamental constraints for a single-user FGS streaming video system.

B. Problem Formulation

Our objective is to design a rate control strategy to achieve both low *aveMSE* (high *avePSNR*) and low *madMSE* (low *madPNSR*) subject to the constraints of (2) and (5). For offline applications where the entire video content is readily available before the transmission, all R-D information is known and we can formulate the rate control as an optimization problem:

$$\min_{\{rf_j, C_j\}} f(aveMSE, madMSE) \quad s.t.(2)(5) \quad \forall j. \qquad (6)$$

In (6) $f(\cdot,\cdot)$ is a function reflecting the importance and relevance of the average distortion and the quality fluctuation in the human perceptual system. A simple example is a linear combination function of *aveMSE* and *madMSE*. An optimal solution for the above offline problem can be found using standard nonlinear programming with penalty methods. The complexity for searching for the optimal solution would, however, be formidable except for short video clips. Additionally, the offline solution is not applicable to online applications where the video content is not entirely available beforehand. In this paper, we focus on a sequential resource allocation solution that has moderate computational complexity and can accommodate online video applications.

To achieve high perceptual quality, we should make full use of the available bandwidth resource and select the channel transmission rate at the upper bound in (5). After determining C_j, the allowable range of rf_j is specified by (2).

Let R_j represent the upper bound for the FGS rate at the j^{th} time slot, and R_j^p the amount of FGS data needed to achieve the same perceptual quality as the previous frame. We have:

$$R_j \equiv B_{max}^e + C_j - B_{j-1}^e - rb_j, R_j^p \equiv D_j^{-1}(D_{j-1}(r_{j-1})) - rb_j. \tag{7}$$

The strategy of choosing the effective encoding rate for the FGS layer, $\{rf_j\}$, and the channel transmission rate, $\{C_j\}$, closely depends on the relative weights of the average distortion and the perceptual fluctuation in the objective function. To achieve low *aveMSE* alone, one may employ a greedy strategy to make the encoder buffer as full as possible all the time, which leads to the desire to use the upper bound of the FGS rate R_j in (7). When encountering complex frames, which have a larger amount of data at the base layer, we will have very limited budget left for sending their associated FGS enhancement layers. This leads to a potential increase in *madMSE*. On the other hand, low *madMSE* may be achieved by assigning each frame a rate that corresponds to the same amount of distortion, as indicated by R_j^p in (7). To prevent encoder buffer from overflowing when encountering complex frames, we would have to allocate a relatively small amount of data rate for FGS layers. Thus, this second approach would not achieve a desirably low *aveMSE*.

By combining the ideas behind the above two cases, we have designed a new algorithm to achieve an improved tradeoff between the average distortion and the quality fluctuation.

C. Proposed Resource Allocation Algorithm

We introduce two weight factors in our proposed algorithm to solve the above-mentioned problems. To overcome the high fluctuation of quality problem from the lowest *aveMSE* scheme, we propose to use only a fraction of the maximal FGS rate budget, i.e. $R_j^f = \beta R_j$, where $\beta \in [0, 1]$ is a budget factor. Compared to adopting the full budget R_j, the fractional budget can keep the encoder buffer occupancy low to accommodate future I-frames and other complex frames. In other words, the rate budget available to the incoming I-frames will be close to the maximal encoder buffer size plus the full channel bandwidth, allowing for more FGS data of I frames to be sent to avoid a high increase in the *madMSE*.

To overcome the problem of low overall perceptual quality as in the lowest *madMSE* scheme, we relax the requirement of zero *madMSE* fluctuation by taking partial consideration of both the rate to maintain zero *madMSE* (defined earlier as R_j^p) and the current occupancy of the encoder buffer. We quantify this strategy using a weight factor $w_p \in [0, 1]$ and allocate the FGS rate for the j^{th} frame as

$$rf_j \quad = \quad \min\{w_p R_j^p + (1 - w_p) R_j^f, R_j\}. \tag{8}$$

As we can see, the finally allocated FGS rate rf_j is determined using two factors, β and w_p. We now examine how to select appropriate β and w_p to achieve a good tradeoff between low *aveMSE* and *madMSE*.

Table I: Single-User Resource Allocation Algorithm

1. Initialization:

j=RDSC=1, $B_0^e = B_1^d = \cdots = B_{d^c}^d = 0$, $R_0^p = rb_1$, $w = w_L$, $D_0(rb_0) = D_1(rb_1)$.

2. While the last frame of this video is not reached:

a) Calculate the budget factor and weight factor

$\overline{rb} = \frac{1}{j - \max(j-L, 0)} \sum_{i=\max(j-L,0)+1}^{j} rb_i$,

$\beta = \frac{C_{max} - \overline{rb}}{B_{max}^e + C_{max} - \overline{rb}} + \Delta\beta$, $R_j^p = D_j^{-1}(D_{j-1}(r_{j-1})) - rb_j$,

If $\left| R_j^p - R_{j-1}^p \right| / R_{j-1}^p \geq S_T$, then *RDSC=j*.

If $j \in [RDSC, RDSC + P_T]$, then $w_p = w_L$.

Else $w_p = w_L + (w_H - w_L)\left(1 - u(B_{j-1}^e - a) \times (B_{j-1}^e - a)/B_{max}^e\right)^b$,

where $u(\cdot)$ is a step function.

b) Select the channel transmission rate

$C_j = \min\{B_{max}^d - B_{j+d^c-1}^d + r_{j-d^d}, C_{max}\}$.

c) Select FGS rate

$R_j = B_{max}^e + C_j - B_{j-1}^e - rb_j$, $rf_j = \min\{w_p R_j^p + (1 - w_p)\beta R_j, R_j\}$.

d) Update the encoder buffer occupancy information

$B_j^e = \max\{B_{j-1}^e + rb_j + rf_j - C_j, 0\}$

If $B_j^e = 0$, then $C_j = B_{j-1}^e + rb_j + rf_j$

e) Update the decoder buffer occupancy information:

$B_{j+d^c}^d = B_{j+d^c-1}^d + C_j - r_{j-d^d}$

f) $j = j + 1$

1) Selection of β: For a single-scene video program, most frames may have similar R-D characteristics and the rf_j's often are in the same bit planes. In this situation and with a fixed w_p, we can show that as β becomes larger, both the *madPSNR* and the *avePSNR* will increase. However, after β passes a specific value, β_0, the improvement of *avePSNR* is dramatically reduced while the quality fluctuation becomes more significant. This β_0 value thus provides a good trade-off between *avePSNR* and *madPSNR*. Our study shows that

$$\beta_0 = (C_{max} - \overline{rb})/(B_{max}^e + C_{max} - \overline{rb}), \tag{9}$$

where $\overline{rb}$ represents the average rate of the base layer, which can be approximated using a moving average of the bitrate statistics of the past frames.

2) Selection of w_p: In general, a system with a high value of w_p has low fluctuation of quality. When consecutive frames within a video segment have similar R-D characteristics, increasing w_p affects only the *madPSNR*, while the *avePSNR* has little decrease until w_p is close to one. When two adjacent frames exhibit significant difference in R-D characteristics such as when arriving at scene boundary, we need to make adjustment in our rate control strategy to handle the future frames. We propose to dynamically adjust w_p in the range $[w_L, w_H]$ with respect to the encoder buffer occupancy to balance between the need of preventing the encoder buffer from overflowing and the control of the fluctuation of perceptual quality. To reach a tradeoff between fully utilizing the available bandwidth resource and maintaining low fluctuation of quality, we adjust w_p to a low value to utilize more available bandwidth immediately after detecting a change in R-D characteristics. The changes in R-D characteristics can be identified by directly calculating the relative rate change between R_j^p and R_{j-1}^p. The detailed resource allocation algorithm is presented in Table I.

IV. Resource Allocation for Multiple Users

In this section, we extend the proposed bandwidth resource allocation algorithm from handling single user to multiple users. A simple way to dealing with multiple users is to allocate a fixed amount of resource, which includes various buffers and channel bandwidth, to each user, and apply our proposed single-user approach to each user. We shall call this strategy *multiple single-user approach*. A more sophisticated approach allows for dynamically allocating resource among users and has the potential

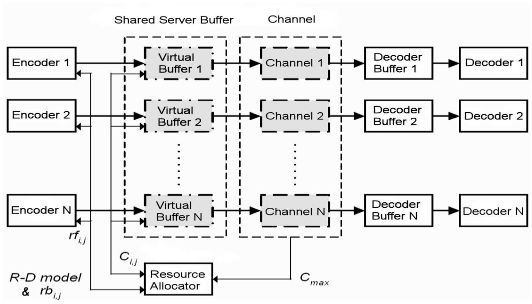


Fig. 2. A Multiuser Video Streaming System

to improve the utilization of critical resources. Multiple users share the total channel bandwidth and buffer capacity, and a central resource allocator dynamically distribute these resources to handle the transmission of each user's video sequence. We shall call this strategy *dynamic multiuser approach*. We will focus on the dynamic multiuser approach and aim at achieving high average quality and low fluctuation of quality for each user. We will examine the scenario of uniform quality of service among all users.

An N-user system is depicted in Figure 2. At the server side, each user has his/her video encoder to encode a different video program in real time. For the i^{th} user, the corresponding encoder sends the R-D models of the current j^{th} frame, $(Rate^k_{i,j}, MSE^k_{i,j})$s', to the resource allocation module. The resource allocation module determines the amount of FGS data to be transmitted. The encoder of each user then moves the base layer data at the rate of $rb_{i,j}$, and the FGS layer bitstream truncated at the allocated rate $rf_{i,j}$ to the shared server buffer whose maximal capacity is B^e_{max}. Denote the shared server buffer occupancy as B^e_j and the amount of data left by the i^{th} user in the server buffer as $B^e_{i,j}$. We can treat $B^e_{i,j}$ as a virtual encoder buffer for the i^{th} user and the sum of all pseudo encoder buffers' size equals to B^e_j. All users also share a channel whose maximal capacity is C_{max}. The resource allocation module needs to determine the channel transmission rate allocated for each user's data at the time slot j, which we denote as $C_{i,j}$ for the i^{th} user. Upon receiving the data packets of the video program, each user stores them temporarily in the decoder buffer, decodes, and renders each frame on time.

In parallel to the single-user case in Section III-A, there are three system constraints for multiuser resource allocation. Under these constraints, we determine the rate of the FGS data, $rf_{i,j}$, and the channel transmission rate, $C_{i,j}$, for each user in the system to achieve low fluctuation of perceptual quality of each program as well as the desired uniform perceptual quality among all programs. Our proposed multiuser resource allocation algorithm first allocates the channel transmission rate for each user. Then, we extend the proposed rate control strategy for single user to the multiple user case to determine the feasible range for FGS layer data of each user.

a) Selection of Channel Transmission Rate: As all users share the overall channel bandwidth in multi-user system, we need to dynamically adjust the transmission rate allocated for each user. Our strategy consists of two steps: first, we assign each user a lower bound of channel transmission rate to prevent all decoder buffers from underflowing. Second, to help drain out the virtual encoder buffers, we distribute the rest of the available bandwidth to each user proportional to his/her recent encoding rate. Thus, when a program encounters an I-frame and leaves a large amount of data in its virtual encoder buffer at the previous time slot, our strategy will assign this user a high channel transmission rate to empty his/her virtual encoder buffer at the current time slot.

b) Selection of FGS Rate: Similar to the proposed single-user strategy, to balance between low fluctuation of perceptual quality and high average quality, we introduce two weight factors to our multiuser algorithm, namely, β and w_p.

We first take an aggregated view on how much total bit rate are spent in the base layer for all users ($\sum_i rb_{i,j}$) and what the upper bound on total FGS rate is at the j^{th} time slot (R_j). The β factor is applied to R_j to obtain a fractional FGS rate budget R^f_j that helps overcome the quality fluctuation.

Next, we distribute R^f_j to each user. If the application desires uniform quality among users, the fractional rate budget for each user, $\{R^f_{i,j}\}$, is determined through the following optimization formulation:

$$\min_{R^f_{1,j},\ldots,R^f_{N,j}} D_j \tag{10}$$

$$\text{s. t.} \begin{cases} \text{Quality:} & D_j \equiv D_{1,j}(R^f_{1,j}) = \cdots = D_{N,j}(R^f_{N,j}), \\ \text{Rate:} & \sum_{i=1}^{N} R^f_{i,j} \leq \beta R_j. \end{cases}$$

Since the R-D functions are monotonically decreasing, this optimization problem with equality constraints can be easily solved using bisection search. Note that we can also provide differentiated service such that each user received different quality by changing the quality constraint.

Finally, we determine the allocated FGS rate for each user, $rf_{i,j}$, using a similar linear combination as in (8):

$$rf_{i,j} \;=\; w_p R^p_{i,j} + (1 - w_p) R^f_{i,j}, \tag{11}$$

where $R^p_{i,j}$ represents the FGS rate for the i^{th} user in the j^{th} time slot to maintain the same quality as the previous frame.

V. EXPERIMENTAL RESULTS

We compare the proposed low-delay resource allocation algorithm with low-fluctuation (*LDLF*) with two alternatives. The first alternative is the constant-bitrate (CBR) approach, which assigns a constant bit rate to each frame. The second alternative is a sliding-window algorithm (*SWLF*) with buffer constraints adapted from [1], which distributes FGS rates to current frame by solving an optimization problem that all frames within a look-ahead window have the consistent and highest possible perceptual quality subject to a given rate budget. Three statistics are used to evaluate the proposed algorithm and the two alternatives: the average PSNR (*avePSNR*), the mean of absolute difference of PSNR (*madPSNR*), and the overall channel utilization (*ChUtiliz*).

For N users in this system, we allocate $N * 80K$ bits for the server buffer and the shared maximal channel capacity is $N * 960$ kbps. For each user, the decoder buffer has maximal size $400K$ bits, the transmission delay, d^c_i, is 3 frames and initial playback delay, d^d_i, is 3 frames. We concatenate 15 QCIF video

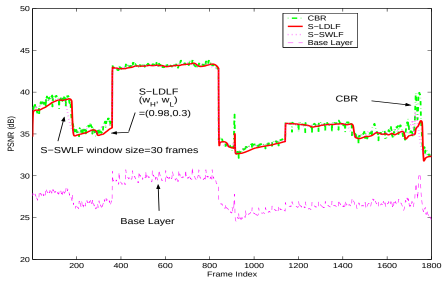


Fig. 3. PSNR results of the S-SWLF and the proposed S-LDLF

Table II: Performance Comparison for a 32-user System

Algorithm	CBR	M-SWLF			M S-LDLF	M-LDLF
(parameters)		Window size			(w_H, w_L)	(w_H, w_L)
(values)		(15)	(30)	(60)	(0.98,0.3)	(0.98,0.3)
avePSNR	37.652	37.304	37.306	37.297	37.512	37.266
madPSNR	0.1159	0.0111	0.0068	0.0045	0.0211	0.0070
ChUtiliz	100%	99.36%	99.39%	99.30%	97.89%	99.00%

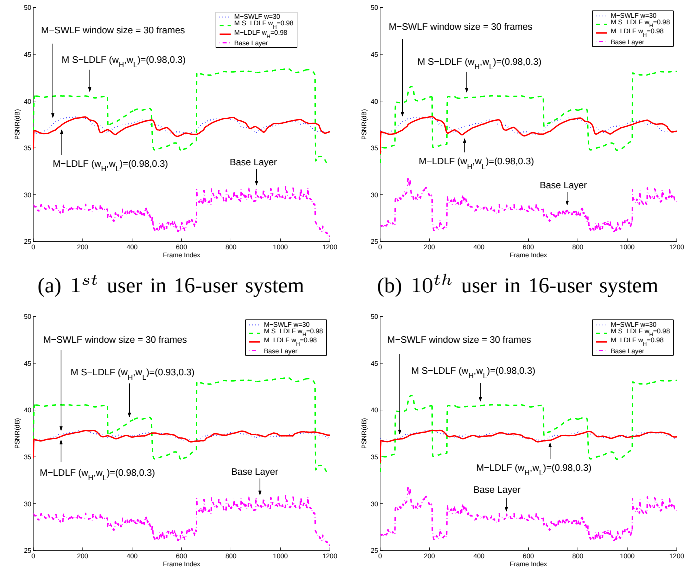
sequences (*Akiyo*, *Carphone*, *Claire*, *Coastguard*, *Container*, *Foreman*, *Grandmother*, *Hall Objects*, *Miss American*, *Mother and Daughter*, *MPEG4 news*, *Salesman*, *Silent*, *Suzie*, and *Trevor*) to form a testing video sequence of 5760 frames. The video frame rate is 30 frames per second. The base layer is generated by MPEG-4 encoder with a fixed quantization step of 30 and the GOP pattern is 29 P frames after one I frame. The FGS layer has up to six bit planes. The parameters $(a, b, P_T, S_T, \Delta\beta)$ used in the *LDLF* algorithm are set to $(B^e_{max}/4, 0.75, 3, 0.3, 0.01)$.

For the single-user system, the video content tested is from frame 361 to 2190. Figure 3 shows the PSNR results of CBR, single-user *SWLF* (*S-SWLF*), and single-user *LDLF* (*S-LDLF*). The *S-SWLF* and *S-LDLF* have similar performance in preceptual quality to each other and outperform the CBR approach. However, our proposed *S-LDLF* has lower delay than S-SWLF and requires no extra storage.

For the multi-user system, the content for the i^{th} user is 1200-frame long and starts from frame 600×(i-1)+1 of the testing video source. If the length of this video source is not long enough, we loop from the beginning of the sequence. Table II summarizes the results for a 32-user system by the multiuser *SWLF* (*M-SWLF*) approach, the multiple single-user approach using the above S-LDLF (*M S-LDLF*), and the proposed multiuser *LDLF* (*M-LDLF*) approach. Figure 4 shows the PSNRs of the first and tenth users when there are 16 and 32 users in *M S-LDLF*, *M-SWLF*, and *M-LDLF* systems, respectively. As we can see, the dynamic multiuser approaches (*M-LDLF* and *M-SWLF*) can provide more uniform quality than the multiple single-user approach even when crossing scene boundaries. When the number of users increases, we can achieve more uniform quality and less quality fluctuation. Between the two dynamic multiuser approaches, our proposed *M-LDLF* approach can achieve similar perceptual quality to that of *M-SWLF* approach with large window size, which needs a longer delay (≥1 second) and a larger storage to store the look-ahead data for all users than the proposed approach.

VI. Conclusions

In summary, we propose an efficient bandwidth resource allocation algorithm for streaming multiple MPEG-4 FGS video



(a) 1^{st} user in 16-user system (b) 10^{th} user in 16-user system

(c) 1^{st} user in 32-user system (d) 10^{th} user in 32-user system

Fig. 4. PSNR results of the 1^{st} and 10^{th} user using three algorithms

sequences. We demonstrate that dynamic multiuser approaches provide more uniform quality and less quality fluctuation than multiple single-user approaches. The proposed algorithm achieves a good tradeoff between the average quality and quality fluctuation criteria. Also, our approach has similar performance to the existing look-ahead sliding-window approach but achieves lower delay and requires lower computation resources. Hence, the proposed *M-LDLF* algorithm is a promising solution for real-time multiuser broadband communications.

References

[1] X.M. Zhang, A. Vetro, Y.Q. Shi, and H. Sun, “Constant quality constrained rate allocation for FGS-coded videos”, *IEEE CSVT*, pp. 121-130, Feb. 2003.

[2] L. Wang and A. Vincent, “Bit allocation and constraints for joint coding of multiple video programs”, *IEEE CSVT* , pp. 949-959, Sep. 1999.

[3] L. Boroczky, A. Y. Nagi, and E.F. Westermann, “Joint rate control with look-ahead for multi-program video coding”, *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 1159-1163, Oct. 2000.

[4] C.-Y. Hsu, A. Ortega, and A.R. Reibman, “Joint selection of source and channel rate for VBR video transmission under ATM policing constraints”, *IEEE JSAC* , pp. 1016-1028, Aug. 1997.

[5] L.-J. Lin and A. Ortega, “Bit-rate control using piecewise approximated rate-distortion characteristics”, *IEEE CSVT*,pp. 446-459, Aug. 1998.

[6] L. Zhao, J. Kim, and C.-C. J. Kuo, “MPEG-4 FGS video streaming with constant-quality rate control and differentiated forwarding”, *Proc. of SPIE, Conference on Visual Communications and Image Processing, 2002.*

[7] H.M. Radha, M. van der Schaar, and Y. Chen, “The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP”, *IEEE Transactions on Multimedia* , pp. 53-68, Mar 2001.

[8] M. v. d. Schaar and H.M. Radha, “A hybrid temporal-SNR fine-granular scalability for internet video”, *IEEE CSVT* , pp. 318-331, March 2001.

[9] A.R. Reibman and B.G. Haskell, “Constraints on variable bit-rate video for ATM networks”, *IEEE CSVT* , pp. 361-372, Dec. 1992.

[10] W. Ding, “Joint encoder and channel rate control of VBR video over ATM networks”, *IEEE CSVT* , pp. 266-278, April 1997.

[11] C.E. Luna, L.P. Kondi, and A.K. Katsaggelos, “Maximizing user utility in video streaming applications”, *IEEE CSVT*, pp.141-148, Feb. 2003.

[12] J. Ribas-Corbera and S. Lei, “Rate control in DCT video coding for low-delay communications”, *IEEE CSVT* , pp. 172-185, Feb. 1999.

[13] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, “Scalable rate control for MPEG-4 video”, *IEEE CSVT* , pp. 878-894, Sept. 2000.

[14] Z. He and S.K. Mitra, “A linear source model and a unified rate control algorithm for DCT video coding”, *IEEE CSVT* , pp. 970-982, Nov. 2002