Fine-Grained Rate Shaping for Video Streaming over Wireless Networks[1]

Trista Pei-chun Chen[2] and Tsuhan Chen[3]

Abstract—Video streaming over wireless networks faces challenges of time-varying packet loss rate and fluctuating bandwidth. In this paper, we focus on streaming precoded video that is both source and channel coded. Dynamic rate shaping has been proposed to "shape" the pre-compressed video to adapt to the fluctuating bandwidth. In our earlier work, rate shaping was extended to shape the channel coded pre-compressed video, to take into account the time-varying packet loss rate as well as the fluctuating bandwidth of the wireless networks. However, prior work on rate shaping can only adjust the rate coarsely. In this paper, we propose "fine-grained rate shaping (FGRS)" to allow for bandwidth adaptation over a wide range of bandwidth and packet loss rate in fine granularities. The video is precoded with fine granularity scalability (FGS) followed by channel coding. Utilizing the fine granularity property of FGS and channel coding, FGRS selectively drops part of the precoded video and still yields decodable bitstream at the decoder. Moreover, FGRS optimizes video streaming rather than achieves heuristic objectives as conventional methods. A two-stage rate-distortion (R-D) optimization algorithm is proposed for FGRS. Promising results of FGRS are shown.

Index Terms—fine-grained rate shaping (FGRS), rate shaping, fine granularity scalability (FGS), rate-distortion (R-D) optimization, video streaming.

---

[1] Work supported in part by Industrial Technology Research Institute.

[2] Trista Pei-chun Chen is now with NVIDIA Corporation, Santa Clara, CA 95050. E-mail: tchen@nvidia.com

[3] Corresponding author: Prof. Tsuhan Chen, Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A. Tel: (412) 268-7536 Fax (412) 268-3890, E-mail: tsuhan@cmu.edu.

I. INTRODUCTION

Due to the rapid growth of wireless communication, video over wireless network has gained a lot of attention [1]-[3]. However, wireless network is hostile for video streaming because of its time-varying error rate and fluctuating bandwidth. Wireless communication often suffers from multi-path fading, inter-symbol interference, and additive white Gaussian noise, etc.; thus, the error rate varies over time. In addition, the bandwidth of the wireless network is also time varying. Therefore, it is important for a video streaming system to address these issues.

Joint source-channel coding techniques [4][5] are often applied to achieve error-resilient video transport with *online coding*. Given the bandwidth requirement, the joint source-channel coder seeks the best allocation of bits for the source and channel coders by varying the coding parameters. However, joint source-channel coding techniques are not suitable for streaming *precoded* video. The precoded video is both source and channel coded prior to transmission. The network conditions are not known at the time of coding. "Rate shaping", which was called dynamic rate shaping (DRS) in [6]-[8], was proposed to solve the bandwidth adaptation problem. DRS "shapes", that is, reduces, the bit rate of the single-layered pre source coded (pre-compressed) video, to meet the real-time bandwidth requirement. DRS adapts the bandwidth by dropping either high frequency coefficients of each block or by dropping several blocks in a frame.

To protect the video from transmission errors, source coded video bitstream is often protected by forward error correction (FEC) codes [9]. Redundant information, known as parity bits, is added to the original source coded bits, assuming systematic codes are adopted. Conventional DRS did not consider shaping for the parity bits in addition to the source coding bits. In our earlier work, we extended rate shaping for streaming the precoded video that is both pre source and channel coded [10]. Such a scheme was called "baseline rate shaping (BRS)". BRS can be applied to precoded video that is

source coded with H.263 [11], MPEG-2 [12], or MPEG-4 [13] scalable coding, and channel coded with Reed-Solomon codes [9] or rate compatible punctured convolutional (RCPC) codes [14]. By means of discrete rate-distortion (R-D) combination, BRS chooses the best state, which corresponds to a certain way to drop part of the precoded video, to satisfy the bandwidth constraint.

The state chosen by BRS however only allows for coarse bandwidth adaptation capability. In this paper, we adopt MPEG-4 fine granularity scalability (FGS) [15] for source coding, and erasure codes [9][16] for FEC coding. Unlike conventional scalability modes such as SNR scalability, MPEG-4 FGS generates a bitstream that is partially decodable over a wide range of bit rates. The more bits the FGS decoder receives, the better the decoded video quality is. On the other hand, it has been known that erasure codes are still decodable if the number of erasures is within the error/loss protection capability of the codes. Therefore, the proposed "fine-grained rate shaping (FGRS)", which is based on the fine granularity property of FGS and erasure codes, allows for fine rate shaping. Moreover, the proposed FGRS optimizes video streaming rather than achieves heuristic objectives such as unequal packet loss protection (UPP). A two-stage rate-distortion (R-D) optimization algorithm is proposed. Note that FGRS focuses on the transport aspect as opposed to the coding aspect of video streaming.

The two-stage R-D optimization is designed to find the solution fast and optimally. In Stage 1, a model-based hyper-surface is trained with a small set of rate and distortion pairs to approximate the relationship between all rate and distortion pairs. The solution of Stage 1 can be found in the intersection in which the hyper-surface meets the bandwidth constraint. In Stage 2, the near-optimal solution from Stage 1 is refined with the hill-climbing based approach. We can see that Stage 1 aims to find the optimal solution globally with the model-based hyper-surface and Stage 2 refines the solution locally.

This paper is organized as follows. In Section II, we introduce baseline rate shaping (BRS) for bandwidth adaptation of the precoded video, which is both scalable and FEC coded. Discrete R-D

combination algorithm is applied to deliver the best video quality. In Section III, fine-grained rate shaping (FGRS) is proposed for streaming the FEC coded FGS bitstream. We first formulate the R-D optimization problem then provide a two-stage R-D optimization algorithm to solve the problem. In Session IV, experiments are carried out to show the superior performance of the proposed FGRS. Concluding remarks are given in Section V.

II. BASELINE RATE SHAPING (BRS)

We propose to use "baseline rate shaping (BRS)" to reduce the bit rate of the precoded video, which is both source and channel coded, given the time-varying error rate and bandwidth. Unlike joint source-channel coding (JSCC) techniques that allocate the bits for the source and channel coders by varying the coding parameters, BRS performs bandwidth adaptation for the *precoded* video at the time of delivery. BRS decision, as to select which part of the precoded video to drop, varies from time to time. There is no need to re-encode as JSCC with different source and channel coder parameters at later time with a different channel condition. Only a different BRS decision needs to be made for the same bitstream. In addition, rate shaping can be applied to adapt to the network condition of each link along the path of transmission from the sender to the receiver. This is in particular suitable for wireless video streaming, since wireless networks are heterogeneous in nature. One single joint source-channel coded bitstream cannot meet the needs of all the links along the path of transmission. Rate shaping can optimize video streaming for each link.

We start by giving the system description of BRS then provide the algorithm for R-D optimization.

A. System Description of Video Streaming with Baseline Rate Shaping (BRS)

Video streaming consists of three stages from the sender to the receiver: (i) precoding, (ii) streaming with rate shaping, and (iii) decoding, as shown in the following from Figure 1 to Figure 3.

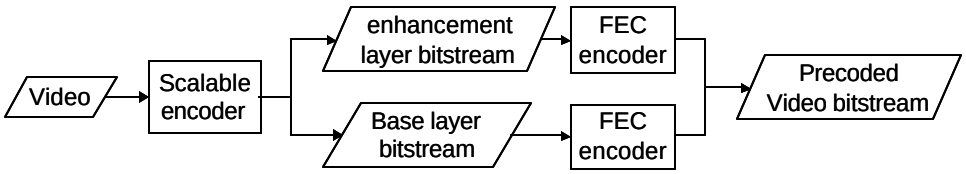


Figure 1. System diagram of the precoding process: scalable encoding followed by FEC encoding

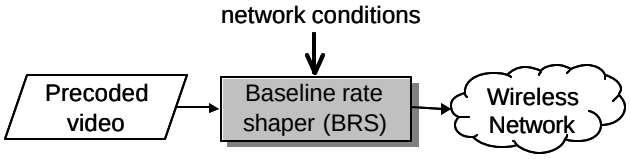


Figure 2. Streaming of the precoded video with BRS

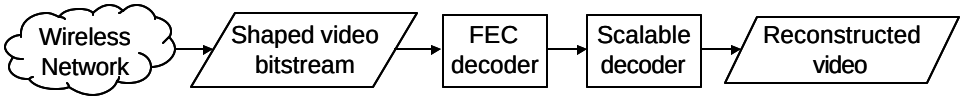


Figure 3. System diagram of the decoding process: FEC decoding followed by scalable decoding

The precoding process (Figure 1) refers to source coding using scalable video coding [11]-[13] and FEC coding. Scalable video coding yields prioritized video bitstream. The concept of rate shaping works for any prioritized video bitstream in general[4]. Without loss of generality, we consider signal-to-noise-ratio (SNR) scalability. Reed-Solomon codes [9] are used as the FEC codes in this paper.

In Figure 2, the pre source and channel coded bitstream is then passed through BRS to adjust its bit rate before being sent to the wireless network. BRS will perform bandwidth adaptation

[4] For example in DRS [6], bits that carries the information of the low frequency DCT coefficients are ranked with high priorities in the video bitstream, as opposed to the ones that carry the information of the high frequency DCT coefficients. By means of data partitioning, the single-layered non-scalable coded bitstream can have different priorities among different segments of the video bitstream.

considering the given packet loss rate in a rate-distortion (R-D) optimized manner. The distortion here is described by mean square error (MSE) of the decoded video. Packet loss rate, instead of bit error rate, is considered since the shaped precoded video will be transmitted in packets.

The decoding process (Figure 3) consists of FEC decoding followed by scalable decoding. The task of rate shaping is performed in the sender and/or midway gateways/routers.

B. Discrete Rate-Distortion (R-D) Optimization Algorithm

BRS reduces the bit rate of each *decision unit* of the precoded video before it sends the precoded video to the wireless network. A decision unit can be a frame, a macroblock, etc., depending on the granularity of the decision. We use a frame as the decision unit herein. BRS performs two kinds of R-D optimizations with: (i) mode decision, and (ii) discrete R-D combination, depending on how much delay the rate shaping decisions can allow. We will discuss both mode decision and discrete R-D combination in the following.

1) BRS by Mode Decision

Let us consider the case in which the video is scalable coded into two layers: one base layer and one enhancement layer. These two layers are FEC coded with unequal packet loss protection (UPP). That is, the base layer is FEC coded with stronger packet loss protection. Therefore, there are four *segments* in the precoded video. The first segment consists of the bits of the base layer video bitstream (upper left segment of Figure 4 (a)). The second segment consists of the bits of the enhancement layer video bitstream (upper right segment of Figure 4 (a)). The third segment consists of the parity bits for the base layer video bitstream (lower left segment of Figure 4 (a)). The fourth segment consists of the parity bits for the enhancement layer video bitstream (lower right segment of Figure 4 (a)). BRS decides a subset of the four segments to send. Note that some constraints need to be imposed for a valid subset. For example, if the segment that consists of the parity bits for the base layer video bitstream is selected, the

segment that consists of the bits of the base layer video bitstream must be selected as well. In the case of two layers of video bitstream, six valid combinations are shown in Figure 4 (b)~(g). We call each valid combination a *state*. Each state is represented by a pair of integers (x, y), where x is the number of segments selected counting from the segment consisting of the bits of the base layer, and y is the number of segments selected counting from the segment consisting of the parity bits for the base layer. Note that x counts from the base layer because the enhancement layer cannot be decoded without the base layer; y counts from the base layer because the base layer needs to be protected by parity bits more than the enhancement layer. The two integers x and y satisfy the relationship of $x \geq y$.

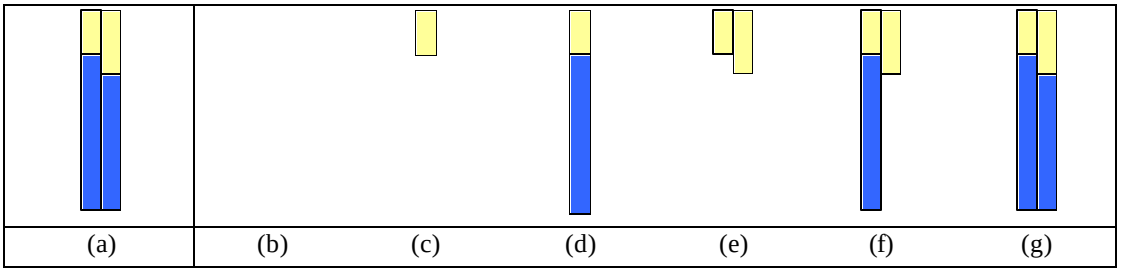


Figure 4. (a) All four segments of the precoded video and (b)~(g) valid states of BRS: (b) state (0,0), (c) state (1,0), (d) state (1,1), (e) state (2,0), (f) state (2,1), and (g) state (2,2)

Each state has its R-D performance represented by a dot in the R-D map, such as the ones shown in Figure 5 (a) and (b). The state constellations are different for different frames because of variations in video content and packet loss rate for different frames. If the bandwidth requirement is "B" for each frame, BRS performs mode decision by selecting the state that has the least distortion. For example in Figure 5, state (1, 1) of Frame 1 and state (2, 0) of Frame 2 are chosen.

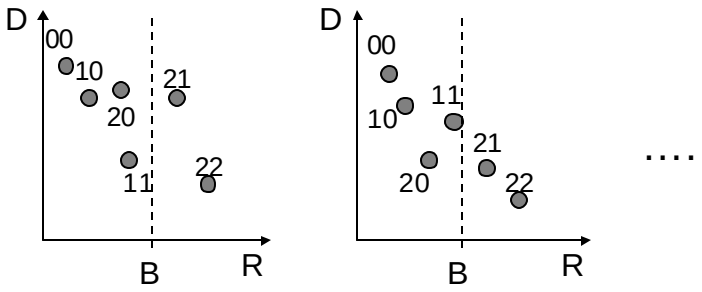


Figure 5. R-D maps of: (a) Frame 1, (b) Frame 2, and so on

2) BRS by Discrete R-D Combination

By allowing some delay in making the rate shaping decision, BRS can optimize video streaming with a better overall quality.By allowing some delay, we mean to accumulate the total bandwidth for a group of pictures (GOP) and to allocate the bandwidth intelligently among frames in a GOP. Video is typically coded with variable bit rate in order to maintain a constant video quality. We want to allocate different numbers of bits for different frames in a GOP to utilize the total bandwidth more efficiently.

Assume that there are F frames in a GOP and the total bandwidth budget for these F frames is C. Let $x(i)$ be the state (represented by a pair of integers mentioned in the last subsection) chosen for frame i, and let $D_{i,x(i)}$ and $R_{i,x(i)}$ be the resulting distortion and rate allocated at frame i respectively. The goal of the rate shaper is to:

$$\text{minimize} \quad \sum_{i=1}^{F} D_{i,x(i)} \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{F} R_{i,x(i)} \le C \tag{2}$$

The discrete R-D combination algorithm [10][19] finds the solution by first eliminating the states that are inside the convex hull (Figure 6 (a) and (b)) for each frame. The algorithm then allocates

the rate step by step to the frame that utilizes the rate more efficiently. That is, among frame m and frame n, if frame m gives a better ratio of distortion decrease over rate increase by moving from the current state $u(m)$ to the next state $u(m)+1$, than frame n, then the rate is allocated to frame m (the next state $u(m)+1$ of frame m is circled in Figure 6 (c)) from the available total bandwidth budget. The allocation process continues until the total bandwidth budget has been consumed completely.

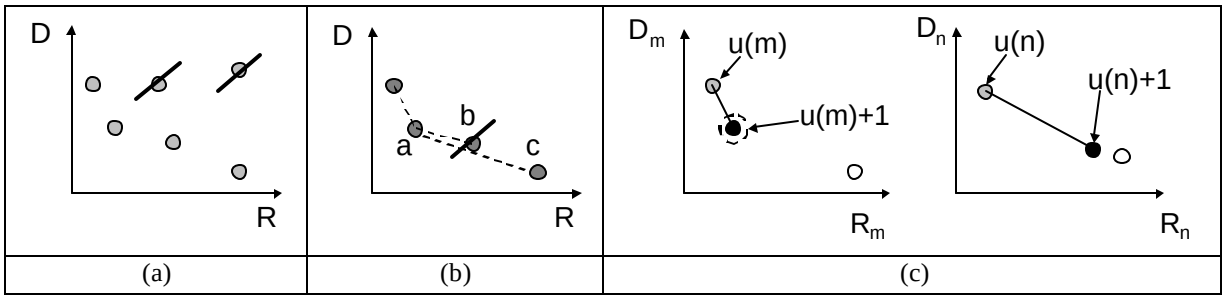


Figure 6. Discrete R-D combination algorithm: (a)(b) elimination of states inside the convex hull of each frame, and (c) allocation of rate to the frame m that utilizes the rate more efficiently

III. FINE-GRAINED RATE SHAPING (FGRS)

As mentioned, BRS performs the bandwidth adaptation for the precoded video by selecting the best state of each frame at any given packet loss rate. Since the packet loss rate and the bandwidth at any given time could lie in any value over a wide range of values, we want to extend the notion of rate shaping to allow for finer grained decisions. There then prompts the need for source and channel coding techniques that offer fine granularities in terms of video quality and packet loss protection, respectively.

Fine granularity scalability (FGS) has been proposed to provide bitstreams that are still decodable when truncated at any byte interval. That is, FGS enhancement layer bitstream is decodable at any rate provided with an intact base layer bitstream. With such a property, FGS was adopted by MPEG-4 for streaming applications [15]. Figure 7 illustrates two layers of video bitstream: the base

layer and the FGS enhancement layer. The base layer is predictive coded while the FGS enhancement layer only uses the corresponding base layer as the reference.

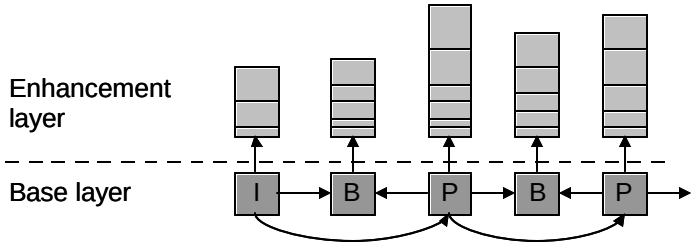


Figure 7. Dependency graph of the base layer and FGS enhancement layer. Base layer has temporal prediction with P and B frames. Enhancement layer is encoded with reference to the base layer only

On the other hand, it has been known that the erasure codes provide "fine-grained" packet loss protection with more and more symbols[5] received at the FEC decoder [9][16]. The "shaped" erasure code is still decodable if the number of erasures/losses from the transmission is no more than $d_{\min} - 1 - (\text{number of unsent symbols})$, where $d_{\min}$ is the minimum distance of the code. An erasure code can successfully decode the message with the number of erasures up to $d_{\min} - 1$, considering both the unsent symbols and the losses taken place in the transmission. Therefore, the more symbols are sent, the better the sent bitstream can cope with the losses. In this paper, we use Reed-Solomon codes as the erasure codes as mentioned in the last section. In Reed-Solomon codes, $d_{\min} - 1$ equals $n - k$, where k is the message size in symbols and n is the code size in symbols. Thus, the partial code with size $r \le n$ is still decodable if the number of losses from the transmission is no more than $r - k$.

A. System Description of Video Streaming with Fine-Grained Rate Shaping (FGRS)

Similar to BRS, there are three stages for transmitting the video from the sender to the receiver: (i) precoding, (ii) streaming with rate shaping, and (iii) decoding, as shown from Figure 8 to Figure 10.

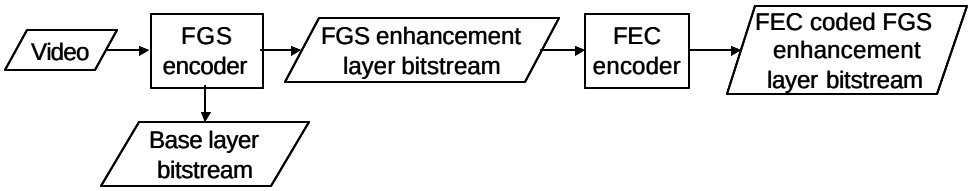


Figure 8. System diagram of the precoding process: FGS encoding followed by FEC encoding

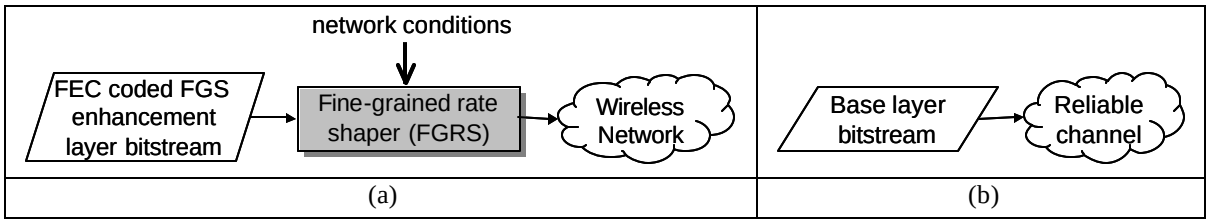


Figure 9. Transport of the precoded bitstreams: (a) transport of the FEC coded FGS enhancement layer bitstream with rate shaper via the wireless network, and (b) transport of the base layer bitstream via the reliable channel

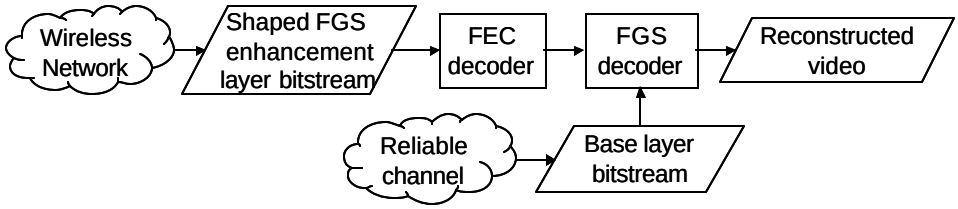


Figure 10. System diagram of the decoding process: FEC decoding followed by FGS decoding

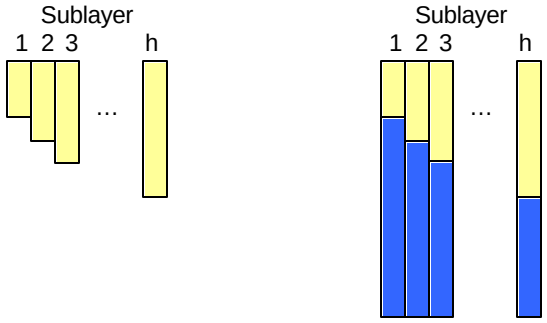
Through MPEG-4 encoding, two layers of bitstream are generated: one base layer and one FGS enhancement layer (Figure 7). We will consider hereafter the bandwidth adaptation and packet loss resilience for the FGS enhancement layer bitstream only, assuming that the base layer bitstream is reliably transmitted as shown in Figure 9 (b) or is considered by approaches outside the scope of this paper. The general rule is to perform enhancement layer bandwidth adaptation after the base layer is

[5] "Symbols" are used instead of "bits" since the FEC codes use a symbol as the encoding/decoding unit. In this paper, we use 14 bits for one symbol. The selection of the symbol size in bits depends on the user.

reliably transmitted. The enhancement layer bitstream will not enhance the quality of the video if its reference base layer is corrupted. Otherwise, a drift prevention remedy is needed.

Recalling that we use a frame as the decision unit, let us look at the FGS enhancement layer bitstream of a frame. FGS enhancement layer bitstream consists of bits of all the bit-planes of this frame. The most significant bit-plane (MSB plane) is coded before the less significant bit-planes until the least significant bit-plane (LSB plane). In addition, since the data in each bit-plane is variable length coded (VLC), if some part of a bit-plane is corrupted (due to packet losses), the remaining part of the bit-plane becomes un-decodable. Bits at the beginning of the enhancement layer bitstream of a frame is more important than the following bits.

Before appending the parity symbols to the FGS enhancement layer bitstream, we first divide all the symbols (in this paper each symbol consists of 14 bits) for this frame into several *sublayers* (Figure 11 (a)). The way to divide the symbols into sublayers is arbitrary except that the later sublayers are longer in length than the previous ones, that is $k_1 \geq k_2 \geq \cdots \geq k_h$, since we want to achieve UPP. A natural way to construct the sublayers is to let Sublayer 1 consist of symbols of the MSB plane, Sublayer 2 consist of symbols of the MSB-1 plane, …, and Sublayer h consist of symbols of the LSB plane. Each sublayer is then FEC encoded with erasure codes to the same length n (Figure 11 (b)). The lower portions of the stripes in Figure 11 (b) consist of the parity symbols. The precoded video is stored and can be used later at the time of delivery.



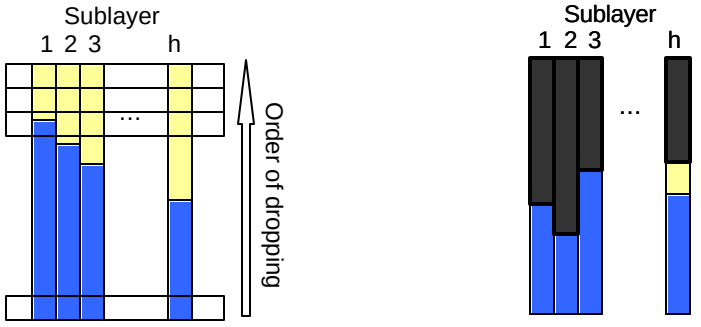
(a) (b)

Figure 11. Precoded video: (a) FGS enhancement layer bitstream in sublayers, and (b) FEC coded FGS enhancement layer bitstream

At the transport stage, FEC coded FGS bitstream is passed through FGRS for bandwidth adaptation given the current packet loss rate. Note that FGRS is different from joint source-channel coding like approaches, which perform FEC encoding for the FGS bitstream at the time of delivery with a bit allocation scheme that achieves certain objectives, as proposed by Radha and van der Schaar [20]-[22] and Yang et al [23]. That is, FGRS focuses on the transport aspect as opposed to the coding aspect. Moreover, FGRS optimizes video streaming rather than achieves certain objectives. We will elaborate on the optimization algorithm taken later.

B. Fine-Grained Rate Shaping (FGRS)

With the precoded video, bandwidth adaptation can be implemented naively by dropping the symbols in the order shown in Figure 12 (a). Given a certain bandwidth requirement for this frame, Sublayer 1 has more parity symbols kept than Sublayer 2 and so on. Shaped bitstream with such a bandwidth adaptation scheme has UPP to the sublayers. We will refer to this method as "UPPRS" herein. However, such UPPRS scheme might not be optimal. We propose FGRS (Figure 12 (b)) for bandwidth adaptation given the current packet loss rate. The darken bars in Figure 12 (b) are selected to be sent by FGRS.



(a) (b)

Figure 12. Bandwidth adaptation with (a) UPPRS and (b) FGRS. The part represented by darken bars are selected to be sent by FGRS

Let us start from the problem formulation. A FGS enhancement layer bitstream provides better and better video quality as more and more sublayers are correctly decoded. In other words, the total distortion is decreased as more sublayers are correctly decoded. With Sublayer 1 correctly decoded, we reduce the total distortion by G_1 (accumulated *gain* is G_1); with Sublayer 2 correctly decoded, we reduce the total distortion further by G_2 (accumulated gain is $G_1 + G_2$); and so on. If Sublayer i is corrupted, the following Sublayers $i+1$, $i+2$, etc., become un-decodable. Note that gain G_i of Sublayer i can either (i) be calculated given the FGS bitstream, after performing partial decoding; or (ii) be embedded in the bitstream as the "meta-data". G_i of Sublayer i is different for every frame.

Since the precoded video is transmitted over error prone wireless networks, sublayers are subject to loss and have certain recovery rates given a particular rate shaping decision. The *expected* accumulated gain is then:

$$G = \sum_{i=1}^{h} \left(G_i \prod_{j=1}^{i} v_j \right) \tag{3}$$

where h is the number of sublayers of this frame, and v_j is the recovery rate of Sublayer j, which is a function of r_j as will be shown later. Sublayer j is recoverable (or successfully decodable) if the number of erasures resulting from the lossy transmission is no more than $r_j - k_j$. k_j is the message (the symbols from the FGS bitstream) size of Sublayer j, and r_j is the number of symbols selected to

be sent for Sublayer j. The recovery rate v_j is the summation of the probabilities that no loss occur, one erasure occurs, and so on until $r_j - k_j$ erasures occur.

$$v_j = \sum_{l=0}^{r_j - k_j} p\{l \text{ erasures occur}\},\ j = 1 \sim h \tag{4}$$

If each erasure occurs as a Bernoulli trial with probability e_m, the probability of having l erasures out of r_j symbols is,

$$p\{l \text{ erasures occur}\} = \binom{r_j}{l} (e_m)^l (1 - e_m)^{r_j - l} \tag{5}$$

The symbol loss rate can be derived from the packet loss rate as $e_m = 1 - (1 - e_p)^{m/s}$, where s is the packet size and m is the symbol size in bits. Depending on the error model (Bernoulli trial, two-state Markov model, etc.), (5) can be replaced with different probability functions.

By choosing different combinations of the number of symbols for each sublayer, the expected accumulated gain will be different. The rate-shaping problem can then be formulated as follows:

$$\text{maximize} \quad G = \sum_{i=1}^{h} \left(G_i \prod_{j=1}^{i} v_j \right) \tag{6}$$

$$\text{subject to} \quad \sum_{i=1}^{h} r_i \le B \tag{7}$$

To solve the problem, an exhausted search on all possible combinations of $\mathbf{r} = [r_1 \quad r_2 \quad \cdots \quad r_h]$ or hill climbing based approaches as described in [24]-[26], where R-D optimization is made for automatic repeat request (ARQ) decisions, can be performed. We propose in this paper a two-stage R-D optimization algorithm. The two-stage R-D optimization algorithm first finds

the *near-optimal* solution fast. The near-optimal solution is then *refined* by the hill climbing approach. The proposed two-stage R-D optimization is different from [24]-[26] in three folds. First, the model-based Stage 1 allows us to examine fewer samples from all operational R-D states. Only a small set of samples are needed to train the model needed for R-D optimization. Second, the proposed distortion measure (or "expected accumulated gain" in the terminology of the paper) accounts for the effects of packet loss as well as the channel codes by means of recovery rates. Finally, the proposed two-stage R-D optimization algorithm can avoid the problem that the solution could be trapped in the local maximum or reach the local maximum too slow. Due to the complexity consideration, Stage 2 can be skipped. Stage 1 does not just serve as a simple initialization stage. It can already find a near-optimal solution.

Packetization is performed after rate shaping. That is, symbols are grouped into packets after the decision of $\mathbf{r} = [r_1 \quad r_2 \quad \cdots \quad r_h]$ has been made. Similar packetization method can be found in [22], while [27] applied bit errors on the bitstream directly. The packets can be sent with "User Datagram Protocol (UDP)" [28]. It is assumed that any error in the packet will result in a packet loss. More considerations on packetization can be found in UDP-Lite [29]. This paper focuses on rate shaping assuming the network condition is provided regardless of which specific packetization method is used.

1) Two-stage R-D Optimization: Stage 1

We can see from (3) and (4) that the expected accumulated gain G is related to $\mathbf{r} = [r_1 \quad r_2 \quad \cdots \quad r_h]$ implicitly through the recovery rates $\mathbf{v} = [v_1 \quad v_2 \quad \cdots \quad v_h]$. We can instead find a model-based hyper-surface that explicitly relates $\mathbf{r}$ and G. The model parameters can be trained from a set of training data $(\mathbf{r}, G)$, where $\mathbf{r}$ values are chosen by the user and G values can be computed from (3) and (4). The optimal solution is in the intersection (Figure 13) in which the model-based hyper-surface meets the

bandwidth constraint. A complex model, with a lot of parameters, can be used to describe as close as possible the true distribution of the R-D states. The solution obtained with this model will be as close to optimal as possible. However, the number of $(\mathbf{r}, G)$ pairs needed to train the model-based hyper-surface increases with the number of parameters.

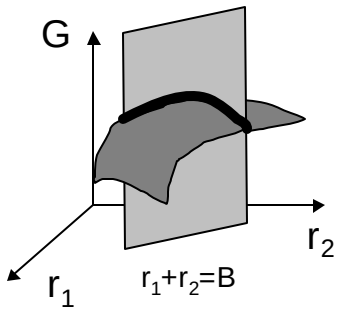


Figure 13. Intersection of the model-based hyper-surface (dark surface) and the bandwidth constraint (gray plane), illustrated with $h = 2$

In this paper, we use a quadratic equation to describe the relation between $\mathbf{r}$ and G as follows:

$$\hat{G} = \sum_{i=1}^{h} a_i r_i^2 + \sum_{i,j=1,i\neq j}^{h} b_{ij} r_i r_j + \sum_{i=1}^{h} c_i r_i + d \tag{8}$$

To distinguish the hyper-surface modeled $\hat{G}$ from the real expected gain G, we denote the former with a "head" sign. The model parameters a_i, b_{ij}, c_i, and d are trained differently for each frame. They can be solved by surface fitting with a set of training data $(\mathbf{r}, G)$ obtained by (3) and (4). For example, the parameters can be computed by:

$$\begin{pmatrix} a_i\text{'s} \\ b_{ij}\text{'s} \\ c_i\text{'s} \\ d \end{pmatrix} = \left(R^T R\right)^{-1} R^T \begin{pmatrix} {}^1G \\ {}^2G \\ \vdots \\ {}^{\Xi}G \end{pmatrix} \tag{9}$$

where the left super index of G is the index of the training data, R is a matrix consisting Ξ rows of $\left(r_i^2\text{'s},\quad r_i r_j\text{'s},\quad r_i\text{'s},\quad 1\right)$. The complexity of computing a_i's, b_{ij}'s, c_i's, and d relates to the number of parameters h^2+h+1 and the number of training data Ξ, using (9). Note that the number of training data Ξ is in general much greater than the number of parameters h^2+h+1. Thus, a more complex model, such as a third-order model with h^3+h^2+h+1 parameters, is not suitable since it requires much more training data than a quadratic model. In addition, second-order Taylor expansion can approximate nicely most functions. (8) can be seen as a second-order approximation to (3). To reduce the computation complexity in reality, we can also choose a smaller h if the precoding process is also under our control (which is outside the scope of the rate shaper).

With (8), the near-optimal solution can be obtained by the use of Lagrange multiplier as follows:

$$J=\left(\sum_{i=1}^{h}a_i r_i^2+\sum_{i,j=1,i\neq j}^{h}b_{ij}r_i r_j+\sum_{i=1}^{h}c_i r_i+d\right)+l\left(\sum_{i=1}^{h}r_i-B\right) \tag{10}$$

By $\frac{\partial J}{\partial r_i}=0$, we get:

$$r_i=\frac{-1}{2a_i}\left(\sum_{j=1,\,j\neq i}^{h}b_{ij}r_j+c_i+l\right) \tag{11}$$

where l is:

$$l=\frac{2B+\sum_{i=1}^{h}\frac{1}{a_i}\left(\sum_{j=1,\,j\neq i}^{h}b_{ij}r_j+c_i\right)}{-\sum_{i=1}^{h}\frac{1}{a_i}} \tag{12}$$

The near-optimal solution can be solved recursively using (11) and (12), starting from the initial condition that all sublayers are allocated with equal number of symbols, $r_1 = r_2 = \cdots = r_h = \frac{B}{h}$.

2) Two-stage R-D Optimization: Stage 2

Stage 1 of the two-stage R-D optimization gives a near-optimal solution. The solution can be refined by a hill-climbing based approach (Figure 14). The solution from Stage 1 is perturbed in Stage 2 in order to yield a larger expected accumulated gain. The process can be iterated until the solution reaches a stopping criterion such as the convergence.

```
While (stop == false)
    zi = ri for all i=1~h
    For (j=1; j<=h; j++)
        For (k=1; k<=h; k++)
            zk = zk + delta for k==j //Increase sublayer j
            zk = zk - delta/(h-1) for k!=j //Decrease others
        End-for
        Evaluate Gj
    End-for
    Find the j* with the largest Gj*.
    For (i=1; i<=h; i++)
        ri = ri + delta for i==j*
        ri = ri - delta/(h-1) for i!=j*
    End-for
    Calculate the stop criterion.
End-while
```

Figure 14. Pseudocodes of hill-climbing algorithm

The idea of allocating bandwidth optimally for sublayers can be extended to a higher level to allocate bandwidth efficiently among frames in a GOP. The problem formulation is slightly different from the original (6)-(7) as follows:

$$\text{maximize} \qquad G = \sum_{m=1}^{F}\left[\sum_{i=1}^{h}\left(G_{mi}\prod_{j=1}^{i}v_{mj}\right)\right] \tag{13}$$

$$\text{subject to} \quad \sum_{m=1}^{F}\sum_{i=1}^{h} r_{mi} \le C \tag{14}$$

where F is the number of frames in a GOP. FGRS will incur delay with duration of F frames if it allows for optimization among frames in a GOP.

To summarize, the proposed FGRS achieves the best streaming performance for FEC coded FGS bitstream with the two-stage R-D optimization. The two-stage R-D optimization obtains the optimal solution by first finding the near-optimal solution, then refining the solution with a hill-climbing based approach.

IV. EXPERIMENT

We start by describing the wireless network simulation for the experiment. We then compare the proposed FGRS with the naive UPPRS described in Figure 12 (a).

A. Experiment Setup

Wireless networks are generally with time-varying packet loss rate and fluctuating bandwidth. The packet loss rate and bandwidth vary at each time interval. We simulate random bandwidth fluctuation according to an autoregressive (AR) process [30], and use a two-state Markov model [17][18] to simulate the bursty bit errors. The two-state Markov model is also adopted by [31][32]. "Good" and "Bad" in Figure 15 correspond to error free and erroneous states of a bit respectively. The bit error rate e_b is related to the transition probabilities p and q by $e_b = p/(p+q)$.

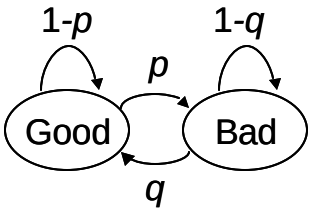


Figure 15. Two-state Markov chain for bit error simulation

Since the coded bitstream is transmitted in packets, let us look at how the packet loss rate e_p relates to the transition probability p and the bit error rate e_b. With bit error rate e_b, transition probability p, and packet size s, the packet loss rate of the s-bit packet is,

$$e_p = 1-(1-e_b)(1-p)^{s-1} \tag{15}$$

We observe two properties from (15) given the same bit error rate e_b: (i) the smaller the transition probability p, the smaller the packet loss rate e_p, and (ii) the smaller the packet size s, the smaller the packet loss rate e_p. These two properties are shown in Figure 16 with $e_b = 10^{-4}$.

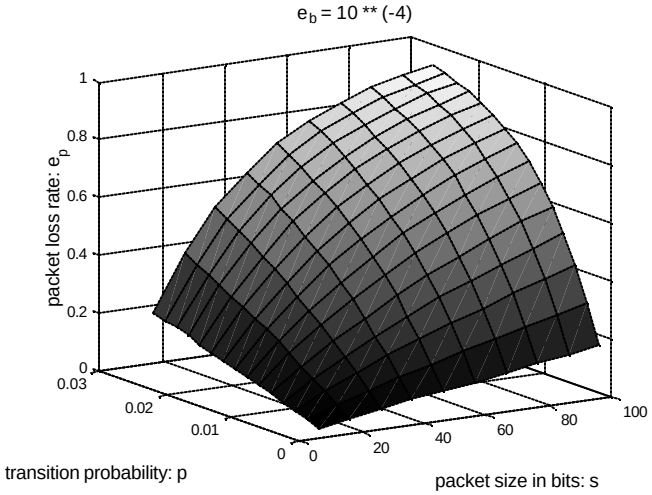


Figure 16. Packet loss rate as a function of the transition probability and the packet size

Besides the two properties we have just seen, it is also known that to detect the loss of packets, some information as the packet number has to be added to each packet. The smaller the packet is, the heavier the overhead is. Therefore, it is a trade-off between the selection of the packet size and the resulting packet loss rate. We use $s = 280$ (bits) in this paper. Users can select the packet size s according to real system consideration using (15).

The time varying bandwidth is simulated pseudo-randomly according to an auto-regressive (AR) process. The bandwidth available at current time t is fed to FGRS optimization of time $t+1$, in order to simulate the delay nature of the network feedback. Such delay in feedback will not affect too much the performance since the bandwidth requirements of the two consecutive frames are closely related given the AR assumption. Example traces of simulated packet loss rate and bandwidth observed at the rate shaper are shown below. The packet loss rate is plotted using the line and the bandwidth is illustrated using the vertical bars. Each interval in the axis of time index represents 0.33 sec.

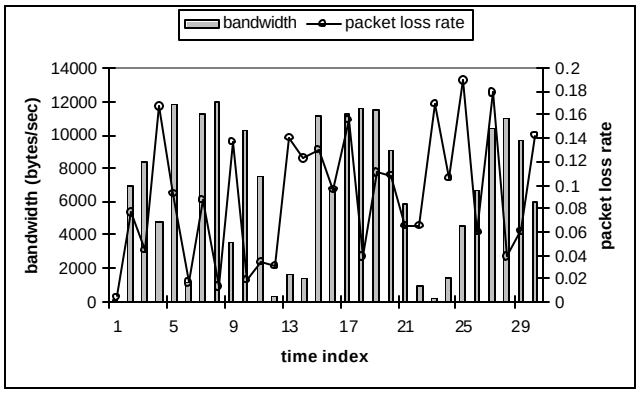


Figure 17. Network conditions: bandwidth and packet loss rate fluctuations

The test video sequences are “akiyo”, “foreman”, and “stefan” in common intermediate format (CIF) (Figure 18 (a)-(c)). The frame rate is three frames/sec.

(a) (b) (c)

Figure 18. Test video sequences in CIF: (a) akiyo, (b) foreman, and (c) stefan

B. Experiment Result

Results for Sequence "akiyo" is shown in Figure 20 and Figure 21. Results for Sequence "foreman" is shown in Figure 22 and Figure 23. Results for Sequence "stefan" is shown in Figure 24 and Figure 25. The overall PSNR performance for all three test sequences are listed in Figure 26 and Table 1. Results for different wireless channel conditions are shown in Figure 27.

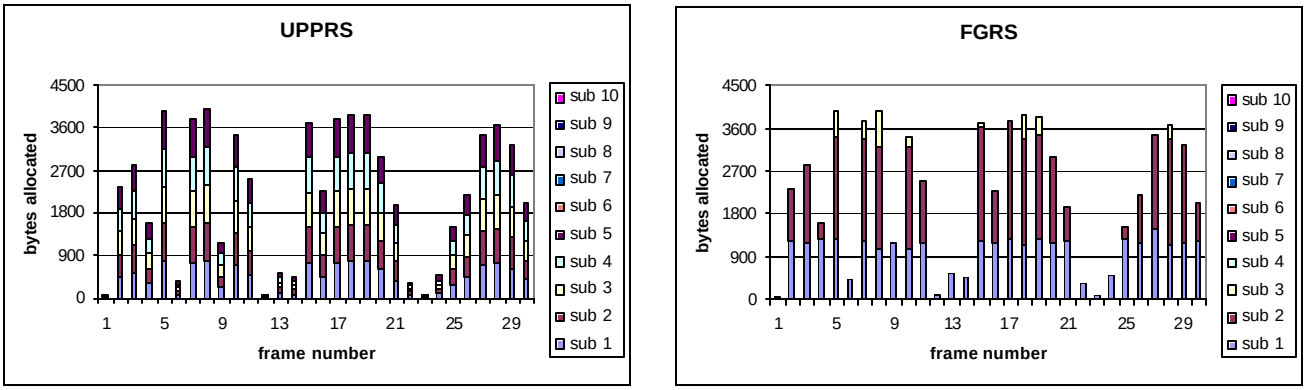
Figure 20, Figure 22, and Figure 24 show how bit-allocation with "UPPRS" and "FGRS" is done in bytes (converted from number of symbols) for each sublayer. After bit-allocation, the number of symbols to send is constrained to be at least k_i for each sublayer (that is, to satisfy $r_i \geq k_i$) by moving the number of symbols allocated for the higher sublayers to the lower layers that does not satisfy $r_i \geq k_i$ as shown in Figure 19.

```
For (i=1; i<=h; i++)
    If r_i < k_i
        a = k_i – r_i
        //the difference needed to satisfy r_i> k_i

        b = c = 0
        For (j=h; j>=1 & c<a; j--)
            b  = r_j > a ? a : r_j
            //the symbols got from Sublayer j

            c  += b
            r_j -= b
        End-for
        r_i = k_i
    End-if
End-for
```

Figure 19. Pseudocodes to satisfy $r_i \geq k_i$ after bit-allocation

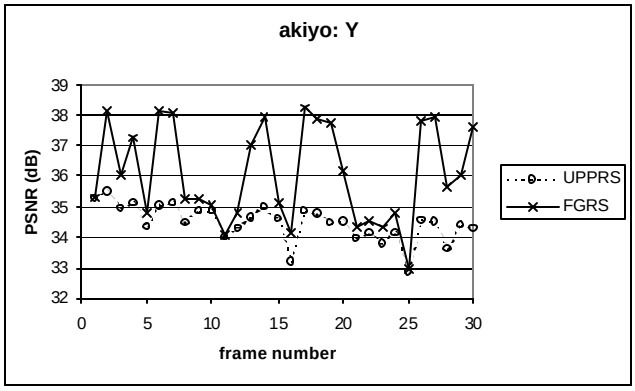
With limited bandwidth, FGRS allocates enough bytes to Sublayer 1 (indicated as sub 1 in the figures) first, than to Sublayer 2, and so on. Allocating enough bytes to a sublayer means providing enough packet loss protection, but not allocating too many bytes as to include too much redundancy. The bit allocation process happens automatically by the proposed two-stage R-D optimization considering the current packet loss rate and the bandwidth requirement.

From the frame-by-frame PSNR performance in Figure 21, Figure 23 and Figure 25, we see that the proposed FGRS provides superior results to UPPRS. Comparing performance with different sequences, the PSNR improvement of FGRS over UPPRS is the most significant in Sequence “akiyo”, followed by Sequence “foreman” and “stefan”. Sequence “stefan” is the most challenging one with the most complex scene and the highest motion. The source-coding rates of the FGS enhancement layer bitstream of “akiyo”, “foreman”, and “stefan” are 354.69 kbits/sec, 747.74 kbits/sec, and 975.70 kbits/sec. Hence, given the same amount of bits allocated by FGRS, the PSNR of Sequence “stefan” is the smallest among the three. Considering the gain in the Y component, FGRS yields 0.76 dB to 1.38 dB improvement compared to UPPRS as shown in Table 1.

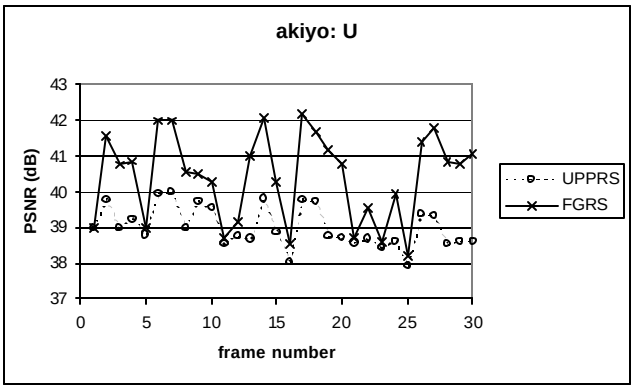


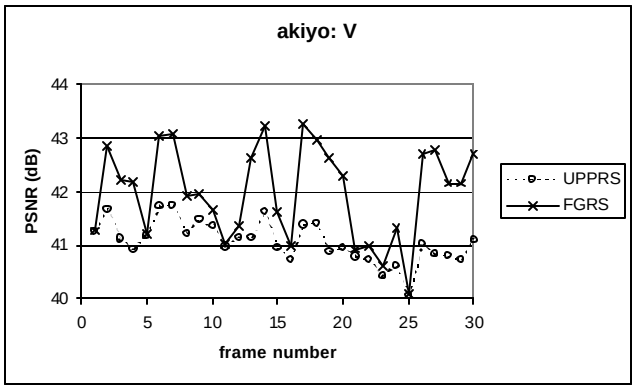
(a) (b)

Figure 20. Sublayer bit allocations with Sequence "akiyo" by (a) UPPRS and (b) FGRS.



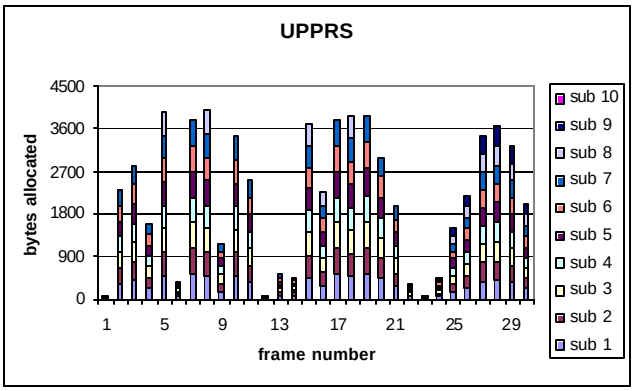
(a)



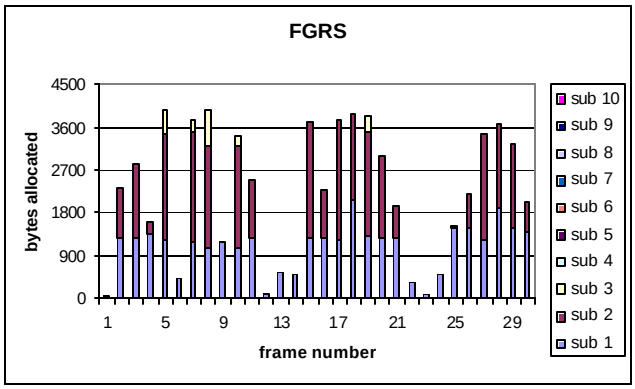


(b) (c)

Figure 21. Frame by frame PSNR of UPPRS and FGRS with Sequence "akiyo": (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component

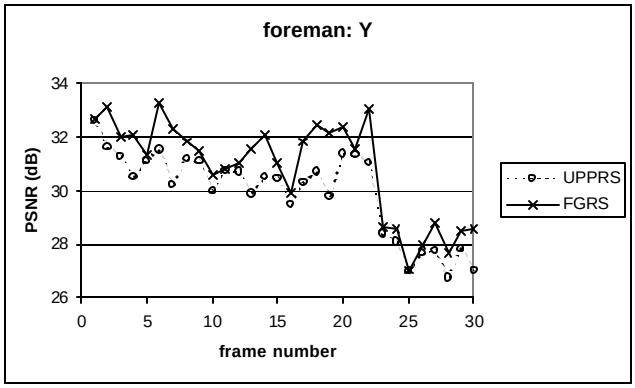


(a)

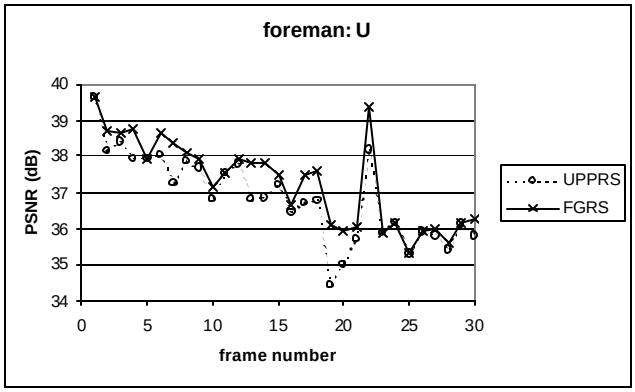


(b)

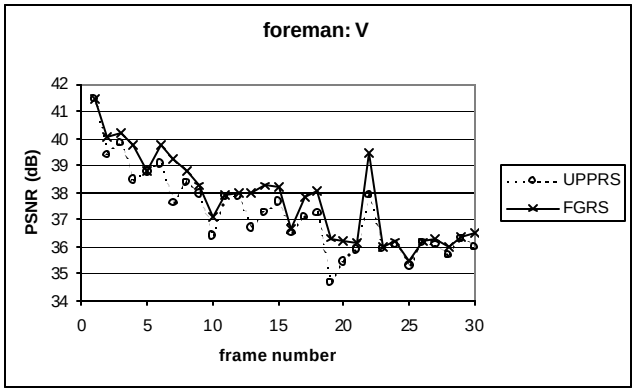
Figure 22. Sublayer bit allocations with Sequence "foreman" by (a) UPPRS and (b) FGRS



(a)

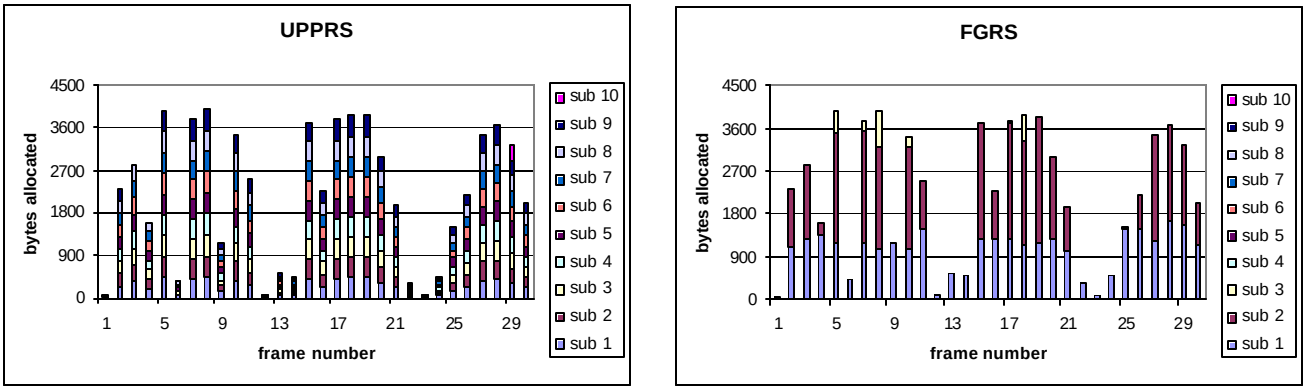


(b)



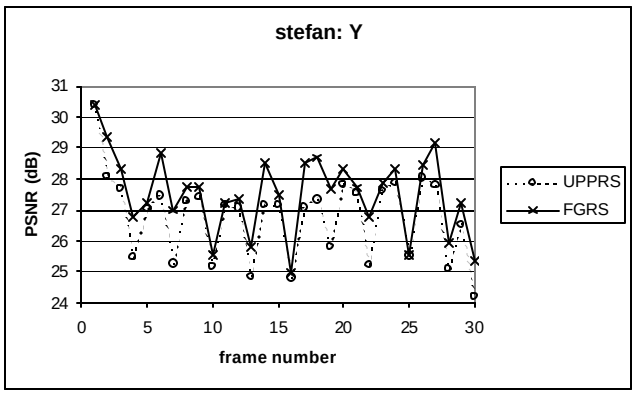
(c)

Figure 23. Frame by frame PSNR of UPPRS and FGRS with Sequence "foreman": (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component

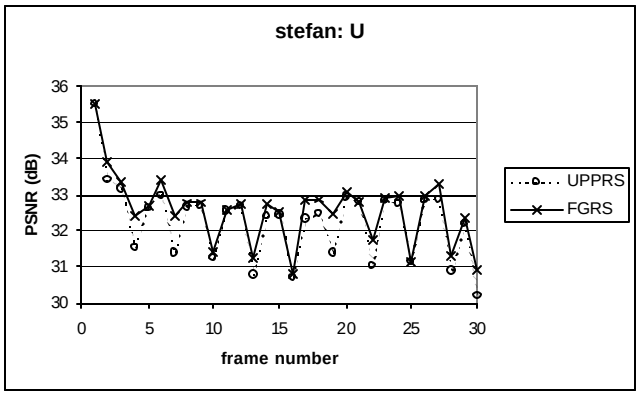


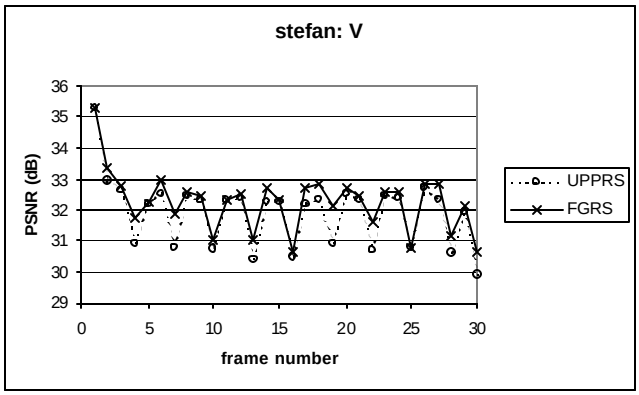
(a) (b)

Figure 24. Sublayer bit allocations with Sequence "stefan" by (a) UPPRS and (b) FGRS



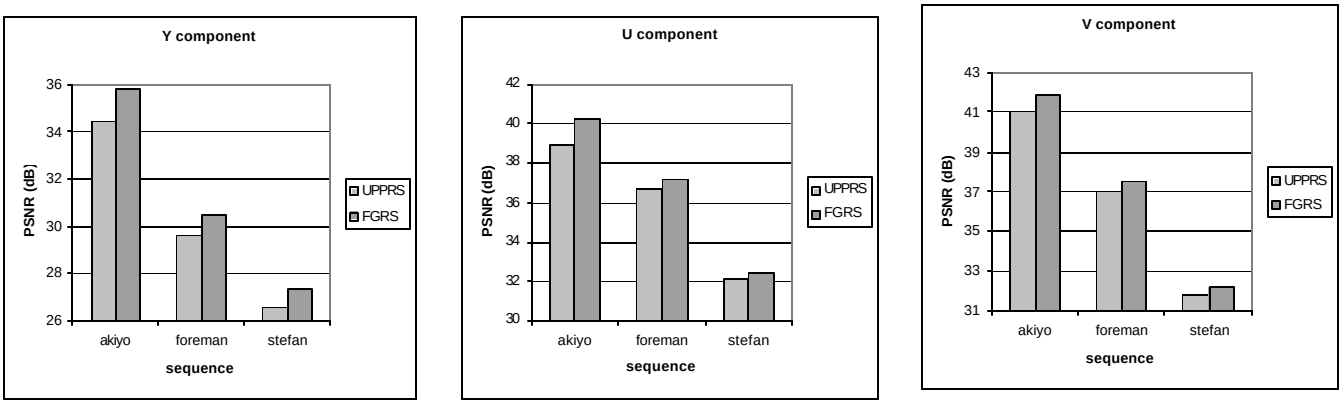
(a)





(b) (c)

Figure 25. Frame by frame PSNR of UPPRS and FGRS with Sequence "stefan": (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component



(a) (b) (c)

Figure 26. Overall PSNR of UPPRS and FGRS with sequences "akiyo", "foreman", and "stefan": (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component

Table 1. PSNR gains in Y, U, and V components with sequences "akiyo", "foreman", and "stefan"

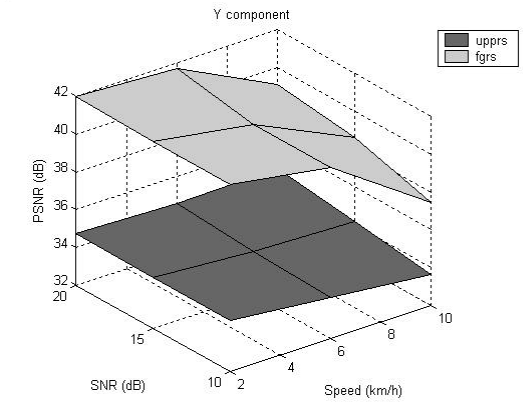
PSNR gain (dB)	Y component	U component	V component
akiyo	1.38	1.28	0.87

foreman	0.86	0.44	0.52
stefan	0.76	0.34	0.38

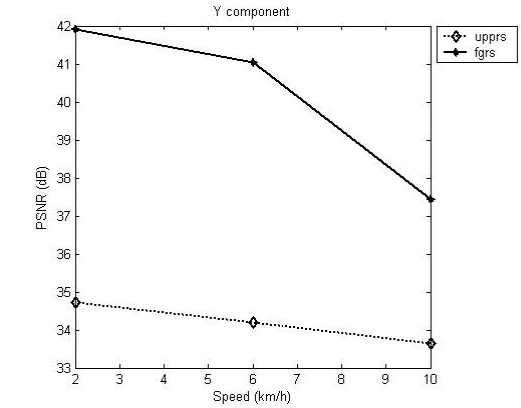
To validate the performance of the proposed algorithm, the performance in terms of the overall PSNR of the Y components at various wireless channel conditions is shown in Figure 27, where we consider a two-state Markov model at various speeds and SNR [17]. Figure 27 (a) shows the 3-D plots of the overall PSNR. At all wireless channel conditions, "fgrs" outperforms "upprs".

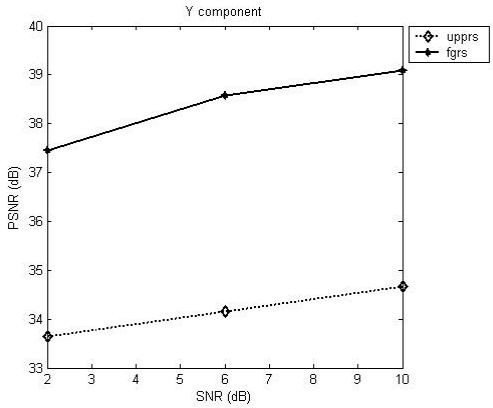
Figure 27 (b) shows the overall PSNR at various speeds at $\mathrm{SNR} = 10\,\mathrm{dB}$. Fixed SNR value gives the same bit error rate (BER) of the wireless channel. The higher the speed is, the more bursty the bit error of the wireless channel is. In other words, the larger the transition probability is. From the results, we see that the PSNR drops as the speed increases. The higher the transition probability is, the higher the packet-loss rate is, given the same bit error rate. Higher packet-loss rate has the effect of requiring more parity bits in the shaped bitstream, and higher probability of corrupting the packets that carries the shaped bitstream, thus, the PSNR value is lower.

Figure 27 (c) shows the overall PSNR at various SNR at $\mathrm{speed} = 10\,\mathrm{km/h}$. Fixed speed gives the same burstiness of the bit errors of the wireless channel. The larger the SNR is, the smaller the BER is. We see from the results that the PSNR value increases with SNR. Smaller packet-loss rate then leads to a higher PSNR.



(a)





(b) (c)

Figure 27. Performance (PSNR of the Y component) of all methods at various wireless channel conditions for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) PSNR at various speeds; (c) PSNR at various SNR

Optimization for video streaming needs to be real time. As mentioned, in the training process for the model-based hyper-surface, only a few number of operational R-D states need to be examined, which saves the time. Thus, the two-stage R-D optimization is preferred over the hill-climbing based approach. In addition, as mentioned in Section III.B, Step 2 can be skipped without too much performance degradation.

V. CONCLUSION

We proposed in this paper a novel fine-grained rate shaping (FGRS) approach to perform bandwidth adaptation for the precoded video, which is both FGS coded and FEC coded. FGRS utilizes the fine granularity property of FGS and FEC. Moreover, FGRS optimizes video streaming rather than achieves heuristic objectives. A two-stage rate-distortion (R-D) optimization algorithm is used. The two-stage R-D optimization algorithm finds the solution efficiently. The proposed FGRS outperforms UPPRS.

The novelty of the paper lies in three aspects. Although FGS has been proposed to provide fine granularity for pre-compressed video, none prior work has shown how to adapt the rate of the FGS bitstream that is protected by the FEC codes. Note that related work performs FEC encoding for the FGS bitstream at the time of delivery. Secondly, we formulate the FGRS problem as a R-D optimization problem, while the work by van der Schaar and Radha [22] is not optimized but is to achieve a certain target recovery rate. In addition, the distortion measure, which is called "gain" in the paper, is derived from the current packet loss rate in addition to the video characteristics. The gain is defined as the expected gain given the current packet loss rate. Prior work of DRS defines the distortion measure solely from the video characteristics. Thirdly, the R-D optimization problem is solved by the proposed two-stage R-D optimization algorithm, which can achieve the optimal solution fast. It is crucial that optimization for video streaming is done in real time.

Future work includes considering the smoothness criterion in FGRS optimization as [33] to smooth the fluctuating PSNR resulted from the time-varying network conditions. Such fluctuation is not inherent from the FGRS algorithm. We can also investigate more the effect of outdated network information on FGRS, in addition to the simulation done in this paper by delaying the network bandwidth feedback. Moreover, deploying FGRS in a large network system, such as the "End System Multicast (ESM)" [34] system, can be an exciting future research direction.

VI. ACKNOWLEDGEMENT

The authors would like to acknowledge the suggestions of Prof. Mihaela van der Schaar of U.C. Davis, Prof. Jose Moura and Prof. Rohit Negi of CMU, Prof. Alex Eleftheriadis and Prof. Shih-Fu Chang of Columbia University, Prof. Antonio Ortega of USC, and the reviewers of the paper.

VII. REFERENCE

[1] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error Resilient Video Coding Techniques", *IEEE Signal Processing Magazine*, pp. 61-82, July 2000.

[2] J. Cabrera, A. Ortega, and J.I. Ronda, "Stochastic Rate-Control of Video Coders for Wireless Channels", *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6), pp. 496-510, June 2002.

[3] Z. He, J. Cai, C. W. Chen, " Joint Source Channel Rate-Distortion Analysis for Adaptive Mode Selection and Rate Control in Wireless Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6), pp. 511-523, June 2002.

[4] G. Cheung and A. Zakhor, "Bit Allocation for Joint Source/Channel Coding of Scalable Video", *IEEE Transactions on Image Processing*, 9(3), March 2000.

[5] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint Source-Channel Coding for Motion-Compensated DCT-based SNR Scalable Video", *IEEE Transactions on Image Processing*, 11(9), September 2002.

[6] A. Eleftheriadis and D. Anastassiou, "Meeting Arbitrary QoS Constraints using Dynamic Rate Shaping of Coded Digital Video", *NOSSDAV* 1995, pp. 96-106, Durham, New Hampshire, April 1995.

[7] W. Zeng and B. Liu, "Rate Shaping by Block Dropping for Transmission of MPEG-precoded Video over Channels of Dynamic Bandwidth", *ACM Multimedia 96*, Boston, MA, U.S.A, 1996.

[8] S. Jacobs and A. Eleftheriadis, "Streaming Video Using Dynamic Rate Shaping and TCP Congestion Control", *Journal of Visual Communication and Image Representation*, 9(3), 211-222, 1998.

[9] S. Wicker, Error Control Systems for Digital Communication and Storage, Prentice-Hall, 1995.

[10] T. P.-C. Chen and T. Chen, "Adaptive Joint Source-Channel Coding using Rate Shaping", *ICASSP 2002*, Orlando, FL, U.S.A., May 2002.

[11] D. S. Turaga and T. Chen, "Fundamentals of Video Compression: H.263 as an Example", in Compressed Video over Networks, edited by M.-T. Sun and A. R. Reibman, Marcel Dekker, Inc., 2001.

[12] B. G. Haskell, A. Puri, and A. N. Netravali, Digital Video: An Introduction to MPEG-2, Chapman & Hall 1997.

[13] Motion Pictures Experts Group, "Overview of the MPEG-4 Standard", ISO/IEC JTC1/SC29/WG11 N2459, 1998.

[14] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and Their Applications," *IEEE Transactions on Communications*, vol. 36, pp. 389–399, April 1998.

[15] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Coding" Standard", *IEEE Transactions on CSVT*, 11(3), pp. 301-317, March 2001.

[16] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", *Computer Communication Review*, April 1997.

[17] J.-P. Ebert and A. Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation", *TKN Technical Reports Series of Technical University Berlin*, March 1999.

[18] M. Yajnik, S. Moon, J. Kurose, D. Towsley, "Measurement and Modeling of the Temporal Dependence in Packet Loss," *INFOCOM 1999*, pp. 345-52, March 1999.

[19] A. Ortega and K. Ramchandran, "Rate-Distortion Methods for Image and Video Compression". *IEEE Signal Processing Magazine*, 15(6), 23-50, Nov. 1998.

[20] 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, March 2001.

[21] M. van der Schaar and H. Radha, "A Hybrid Temporal-SNR Fine-Granular Scalability for Internet Video", *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 318-331, March 2001.

[22] M. van der Schaar and H. Radha, "Unequal Packet Loss Resilience for Fine-Granular-Scalability Video", *IEEE Transactions on Multimedia*, 3(4), pp 381-394, December 2001.

[23] X. K. Yang, C. Zhu, Z. G. Li, G. N. Feng, S. Wu, and N. Ling, "A Degressive Error Protection Algorithm for MPEG-4 FGS Video Streaming", *ICIP 2002*, pp. 737-740, September 2002.

[24] A. E. Mohr, E. A. Riskin and R. E. Ladner, "Unequal Loss Protection: Graceful Degradation of Image Quality over Packet Erasure Channels through Forward Error Correction," *IEEE J. Select. Areas Comm.*, 18(6), pp. 819-828, June 2000.

[25] P. A. Chou and Z. Miao, "Rate-Distortion Optimized Streaming of Packetized Media", submitted to *IEEE Transactions on Multimedia*, February 2001.

[26] J. Chakareski, P. A. Chou, and B. Aazhang, "Computing Rate-Distortion Optimized Policies for Streaming Media to Wireless Clients", *IEEE Data Compression Conference*, Snowbird, UT, April 2002.

[27] G. Cote, S. Shirani, and F. Kossentini, "Optimal Mode Selection and Synchronization for Robust Video Communications over Error-Prone Networks", *IEEE J. Select. Areas Comm.*, 18(6), June 2000.

[28] J. Postel, "User Datagram Protocol", RFC 768, Aug 1980, http://www.ietf.org/rfc/rfc768.txt .

[29] L.-A. Larzon, M. Degermark, and S. Pink, "Efficient Use of Wireless Bandwidth for Multimedia Applications", *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*, San Diego, pp. 187-193, November 1999.

[30] A. J. Ganesh, "Estimating effective bandwidths from traffic data", Globecom 1996, London, U.K., pp. 654-658, November 1996.

[31] S. Khayam, S. Karande, M. Krappel, and H. Radha, "Cross-Layer Protocol Design for Real-Time Multimedia Applications over 802.11B Networks", *ICME 2003*, Baltimore, pp. 425-428, July 2003.

[32] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "An End-to-End TCP-Friendly Streaming Protocol for Multimedia over Wireless Internet", *ICME 2003*, Baltimore, pp. 429-432, July 2003.

[33] X. M. Zhang, A. Vetro, Y. Q. Shi, and H. Sun, "Constant Quality Constrained Rate Allocation for FGS-Coded Video", IEEE Transactions on Circuits and Systems for Video Technology, 13(2), pp. 121-130, February 2003.

[34] Y.-H. Chu, S. G. Rao, S. Seshan and H. Zhang , "A Case for End System Multicast", *IEEE Journal on Selected Areas in Communication*, Special Issue on Networking Support for Multicast, 20(8), October 2002, pp. 1456-1471.