CARNEGIE MELLON UNIVERSITY

ERROR-RESILIENT RATE SHAPING FOR VIDEO STREAMING OVER PACKET-LOSS NETWORKS

PH.D. DISSERTATION

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND COMPUTER ENGINEERING

BY

TRISTA PEI-CHUN CHEN

PITTSBURGH, PENNSYLVANIA
MAY 2003

Abstract

Video streaming over packet-loss networks faces the challenges that the networks are error-prone, transmission bandwidth is limited and fluctuating, the user device capabilities vary, and networks are heterogeneous. These challenges necessitate the need for smart adaptation of the precoded video. The focus of the thesis is *error-resilient rate shaping* for streaming precoded video over packet-loss networks. Given the packet-loss characteristic of the networks, the precoded video consists of channel-coded as well as source-coded bits. Error-resilient rate shaping is a filtering process that adapts the bit rates of the precoded video, in order to deliver the best video quality given the network condition at the time of delivery. We first illustrate "baseline rate shaping (BRS)" of the proposed error-resilient rate shaping as a baseline. Having introduced BRS with coarse decisions in rate adaptation, more sophisticated error-resilient rate shaping is proposed for layer-coded videos, namely, the enhancement layer video and the base layer video. "Fine-grained rate shaping (FGRS)" is proposed for streaming the enhancement layer video, and "error-concealment aware rate shaping (ECARS)" is proposed for streaming the base layer video. FGRS and ECARS are formulated as rate-distortion (R-D) optimization problems. A two-stage R-D optimization approach is proposed to solve the R-D optimization problem in a fast and accurate manner. FGRS makes use of the fine granularity property of the MPEG-4 fine-granularity-scalability bitstream and outperforms ad-hoc unequal packet-loss protection methods. ECARS takes into account error concealment (EC) performed at the receiver to deliver the part of precoded video that cannot be EC-reconstructed well. Frame dependency due to predictive coding and/or temporal EC is also considered in ECARS by means of feedback from the receiver. Experiments are conducted under various channel conditions and for various types of the video to demonstrate the effectiveness of the proposed scheme. Finally, we see that network conditions are needed in optimizing the streaming performance. In the last part of the thesis, we focus on modeling the video traffic so that we may use the syntactic traffic to probe the network to determine the network condition and optimize the proposed error-resilient rate shaping accordingly.

Acknowledgements

I am extremely grateful to my advisor Prof. Tsuhan Chen, for being a source of inspiration, and being a great mentor during my Ph.D. study. Through his guidance and challenge, I learned to define my research direction and incorporated innovative research ideas. I have also been benefited by his encouragements to interact with researchers in the field, to present research ideas effectively, and to contribute to the societies. I cannot but be deeply thankful to have him as my advisor.

I would like to thank Prof. Jose Moura, Prof. Rohit Negi, and Dr. Mihaela van der Schaar, for being my thesis committee members. Their insightful feedback has been very precious to the improvement of the work. I am grateful to be part of the Electrical and Computer Engineering department, to be stimulated and surrounded by excellent faculty members and motivated students. I would like to thank Prof. B.V.K. Vijaya Kumar for being a great mentor.

I am thankful to all my group mates and friends during my study at Carnegie Mellon University. My group mates, Ta-chien Lin, Deepak Turaga, Howard Leung, Fu Jie Huang, Xiaoming Liu, Cha Zhang, Fang Fang, Ed Lin, Jessie Hsu, Sam Chen, Wende Zhang, Simon Lucey, Kate Shim, Jack Yu, Avinash Baliga, and Michael Kaye, have been spending time and provided valuable suggestions to my work. In addition, they are the best friends. My friends Senan E Guran, Pinky Pongpaibool, and Yu-nu Hsu have always been there for me. Thanks to Elie Shammas, Frances Ning, Iuliana Tanase, and James Bruce for bringing the happy moments.

Finally, I would like to express my earnest gratitude to my parents and brothers for their love and support. In particular, I am deeply indebted to my dear Mom. I would not have reached the accomplishment of the current stage without their strong support. The thesis is dedicated to my parents and brothers.

Table of Contents

List of Tables

List of Illustrations

1. Introduction

Video streaming over packet-loss networks faces the challenges that the networks are error-prone, transmission bandwidth is limited and fluctuating, the user device capabilities vary, and networks are heterogeneous. These challenges necessitate the need for smart adaptation of the precoded video. We propose in the thesis study an error-resilient rate shaping framework (Figure 1) for video streaming over packet-loss networks.

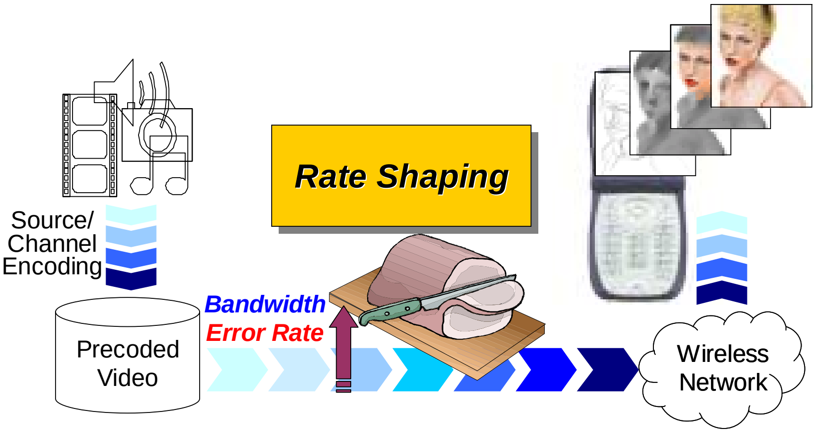


Figure 1. Rate shaping for error-resilient video streaming

Rate shaping is a technique to selectively drop part of the pre (*source*- and/or *channel*-) coded bitstream before the bitstream is sent to the network. To ensure that the *shaped* bitstream can best survive the hostile network condition, rate shaping takes into account the network information as the channel error rate and the available bandwidth, as well as the video source statistics. In addition to rate shaping at the sender, post-processing error concealment can be performed at the receiver to recover the decoded video quality. Furthermore, knowing the receiver can replenish the video data by error concealment, error concealment aware rate shaping can take place at the sender to deliver better quality videos than non- error concealment aware rate shaping.

The proposed error-resilient rate shaping has many advantages over other error-resilient video transport mechanisms [59], namely, error-resilient video coding and joint source-channel coding.

- *Error-resilient rate shaping vs. error-resilient video coding*

  In many situations, the video encoder and decoder are fixed and allow for no modifications to include error-resilient video coding features, such as reversible variable length coding and independent segment prediction, etc. The proposed error-resilient rate shaping does not need to alter the original video encoder and decoder, thus can be adopted by systems, for example the commercial video-on-demand system, in which tremendous amount of work to modify the video coders is needed.

- *Error-resilient rate shaping vs. joint source-channel coding*

  By varying the source and channel encoder parameters, joint source-channel coding [10][29] [55][63] allocates the bits for source and channel encoders to achieve the best video quality given the current network condition. Joint source-channel coding techniques are limited by only providing end-to-end optimization at the time of encoding and are not suitable for streaming the precoded video. The encoded bitstream may not be optimal for transmission along a different path or along the same path at later time. Moreover, rate adaptation for each link might be needed in a heterogeneous network. Rate shaping can optimize the video streaming performance for each link.

1.1. Error-Resilient Rate Shaping: Baseline Rate Shaping (BRS)

Dynamic rate shaping (DRS) [16][17][27][65][66] was first introduced in 1995 to adapt the rates of the pre source-coded (pre-compressed) video to the dynamically varying bandwidth constraints. On the other hand, to protect the video from losses in the packet-loss networks, source-coded video bitstream is often protected by forward error correction (FEC) codes [46][61]. Redundant information, known as parity bits, is added to the original source-coded bits. Conventional DRS does not consider shaping for the parity bits in addition to the source-coding bits, that is, *pre source- and channel- coded* bitstream.

We propose a new framework of rate shaping: error-resilient rate shaping, which adapts the rates of the pre source- and channel- coded bitstream to the conditions of the packet-loss networks. The baseline approach of the proposed error-resilient rate shaping is first introduced, which we call “baseline rate shaping (BRS)”. BRS optimizes the video streaming performance in a rate-distortion (R-D) sense. Two R-D optimization algorithms: BRS by mode decision and BRS by discrete R-D combination are presented.

1.2. Rate Shaping for Enhancement Layer Video: Fine-Grained Rate Shaping (FGRS)

The baseline approach BRS of the proposed error-resilient rate shaping makes decisions on a coarse level. To incorporate a finer granularity, we propose "fine-grained rate shaping (FGRS)" for streaming the enhancement layer video and "error-concealment aware rate shaping (ECARS)" for streaming the base layer video.

We adopt MPEG-4 fine granularity scalability (FGS) [32] for source coding, and erasure codes [46][61] for FEC coding. Unlike conventional scalability techniques such as SNR scalability, MPEG-4 FGS video bitstream is partially decodable over a wide range of bit rates. The more bits of the FGS bitstream is received, the better the video quality is. In addition, it has been known that partial FEC coded bitstream is still decodable within the error correction capability if erasure codes are used. Thus, both FGS and erasure codes provide fine-granularity properties in video quality and in packet-loss protection. Given the FEC coded FGS bitstream as the precoded video, FGRS adapts the rates of the precoded video considering the current packet-loss rate. There are conceptually infinitely many possible combinations of dropping portion of the FGS bitstream and portion of the FEC codes. FGRS seeks the optimal solution in the R-D sense. A new two-stage R-D optimization approach is proposed to select part of the precoded video to drop.

1.3. Rate Shaping for Base Layer Video: Error Concealment Aware Rate Shaping (ECARS)

To have a finer-granular decision instead of the coarse decision made by BRS, "error-concealment aware rate shaping (ECARS)" for streaming the base layer video is proposed in addition to FGRS for streaming the enhancement layer video. Taking into account that the receiver may perform error concealment (EC) if any video data is lost during the transmission, ECARS makes rate-shaping decisions accordingly.

Frame dependency is usually inherent in video bitstream due to predictive coding. In addition, temporal EC might introduce extra frame dependency. Feedback from the receiver to the sender might be helpful in addressing the frame dependency problem in rate shaping. We then introduce two types of ECARS algorithms: without feedback and with feedback. Both evaluate the *gains* of sending some parts of the precoded video as opposed to not sending them. The gain metrics are then included in the R-D optimization formulation. Finally, the two-stage R-D optimization approach is adopted to solve for the R-D optimization problem.

In the case of no feedback, ECARS evaluates the gain considering a particular EC method used at the receiver. In order to incorporate the frame dependency into the rate shaping process, we propose to send the location (and mean) of the corrupted macroblock back to the sender, and use such feedback information to determine the gain in the R-D optimized ECARS.

1.4. Modeling of Video Traffic

To both the video service providers and the network designers, it is important to have a good model for the video traffic. A good model for video traffic allows for better admission control, scheduling, network resource allocation policies, etc., that guarantee a desired quality of service (QoS) as well as a better utilization of the network resources. A good model captures essential characteristics of the real video traffic. The synthetic trace generated by such a model can be used to test the network. In this thesis study, video traffic modeling is useful in helping to gather network conditions.

We present a new stochastic process called the *punctured* autoregressive (AR) process, and use it to model the video traffic. To model the video traffic, we propose to use punctured autoregressive processes modulated by a doubly Markov process. The doubly Markov process models the state of a video frame while the autoregressive process describes the number of bits of a frame at one particular state. The punctured autoregressive process considers the timing information between frames of the same state and thus gives better modeling performance. The model captures the long-range dependency (LRD) characteristics as well as the short-range dependency (SRD) characteristics of the video traffic. Queuing behavior of the punctured autoregressive process is also closer to the real video traffic than the conventional autoregressive process.

1.5. Organization of Thesis

This thesis is organized as follows. Chapter 2 describes the fundamentals of the proposed error-resilient rate shaping. Conventional rate shaping is first introduced followed by description of the characteristics of the packet-loss networks. The baseline approach of the proposed error-resilient rate shaping is then presented.

Chapter 3 describes one of the main ideas of the proposed error-resilient rate shaping: “fine-grained rate shaping (FGRS)” for streaming the enhancement layer video. The proposed two-stage R-D optimization approach will be detailed as well.

Chapter 4 describes the other main idea of the proposed error-resilient rate shaping: “error concealment aware rate shaping (ECARS)” for streaming the base layer video. The focus in this chapter is to determine the gain used for R-D optimization. Two cases of ECARS are discussed: ECARS without feedback and ECARS with feedback.

Chapter 5 discusses about the use of “punctured AR processes” for video traffic modeling. The synthetic traffic generated can be used to probe the network conditions that are eventually fed to the error-resilient rate shaping system.

We summarize the study in Chapter 6 with conclusion, contribution of the thesis, and future directions.

2. Rate Shaping for Error-Resilient Video Streaming

Error-resilient rate shaping is a filtering process that, given a precoded bitstream and the network condition, generates an alterative bitstream that adapts to the network condition. Considering rate shaping for video streaming over the packet-loss networks, the precoded bitstream should be both *source- and channel- coded* to be error resilient. We will illustrate the baseline approach of the proposed error-resilient rate shaping as fundamentals for more advanced rate shaping in the later chapters. In this chapter, we first introduce the conventional rate shaping where the rate adaptation is performed on pre source-coded video only. We then brief some characteristics of packet-loss networks, given that we aim to solve the problem of error-resilient video streaming. Finally, we introduce the baseline approach of our proposed error-resilient rate shaping, which is called baseline rate shaping (BRS).

2.1. Conventional Rate Shaping

Rate shaping is a filtering process that, given a pre source-coded bitstream and the target bandwidth, generates an alterative bitstream that satisfies such a bandwidth constraint. If the bandwidth constraint varies over time, it is called dynamic rate shaping (DRS) [16][17][27][65][66]. Without rate shaping, the bitstream that exceeds the bandwidth constraint will be discarded indiscriminately by the network. The resulting video quality will be degraded unexpectedly.

In a *wide sense*, the format of the video bitstream, e.g. from MPEG-1 to MPEG-4, the resolution of the video, the frame rate of the video, may all be manipulated to achieve the target bit rates. Such kind of rate shaping is usually called *transcoding* [48][49][56]. The other kind of rate shaping is done by means of *re-quantization* of discrete cosine transform (DCT) coefficients. In re-quantization [4][41][60], the entire set of already quantized DCT is mapped to new values at a coarser level of quantization thus resulting in a rate reduction. In [41], local and global activity criteria are used to determine the re-quantization step size. In [60], optimal selection of the re-quantization step size is examined for Intra-coded frames. In [4], optimal selection of the re-quantization step size is analyzed for all frame types.

The focus in this section will be rate shaping in a *strict sense*. That is, rate shaping drops part of the pre source coded video that is considered less critical to the quality of the decoded video, without changing the content of the bitstream. In [16][17][27], selective transmission of transform (DCT) coefficients is presented. In [65][66], rate shaping is achieved by block dropping and the additional error concealment at the receiver. We will describe these two types of strict-sense rate shaping in the following.

2.1.1. Rate Shaping by Selective Transmission of Transform Coefficients

DRS presented in [16][17][27] proposed to selectively send some of the transform coefficients to satisfy the bandwidth constraints. There are two cases of DRS, *constrained* and *general* (or *unconstrained*). In constrained DRS, the number of DCT run-length codes within each block that will be kept is called the “breakpoint”. All DCT coefficients that are above the breakpoint are to be eliminated from the bitstream. In general DRS, the breakpoint becomes a 64-element binary vector, indicating which coefficients within each block will be kept.

1) Constrained DRS of Intra-Coded Pictures

If the video is Intra-coded, there is no dependency between frames. The rate shaping errors in the current frame will not propagate to the next frames. The problem formulation is as follows:

$$\text{minimize} \quad \sum_{i=1}^{N} D_i(b_i), \text{ where } D_i(b_i) = \sum_{k \geq b_i} E_i(k)^2 \tag{2.1}$$

$$\text{subject to} \quad \sum_{i=1}^{N} R_i(b_i) \leq B \tag{2.2}$$

where $b_i \in \{1,2,\cdots,64\}$ is the breakpoint for block i, N is the number of blocks considered, $E_i(k)$ is the DCT coefficient at the kth position, and $R_i(b_i)$ is the rate required to send block i till the breakpoint.

This problem can be converted to a linear programming/integer programming problem with Lagrange multipliers as follows:

$$\min\left\{\sum_{i=1}^{N} D_i(b_i) + \lambda \sum_{i=1}^{N} R_i(b_i)\right\} \tag{2.3}$$

The problem can be solved by an iterative bisection algorithm [19][42][43].

2) Constrained DRS of Intra-Coded Pictures

If the video is Inter-coded with I, P, and B pictures, the dependency between frames adds complexity to the problem. The rate shaping errors are the summation of the accumulated errors (motion compensation of the accumulated errors from the previous frames) and the errors from the current frame (errors from shaping of DCT coefficients of the current frame). The problem formulation is as follows:

$$\text{minimize} \quad \begin{array}{l} \sum_{i=1}^{N}\hat{D}_i(b_i), \text{ where} \\ \hat{D}_i(b_i) = \sum_{k} A_i(k)^2 + \sum_{k \geq b_i}\left[2A_i(\xi(k))E_i(k) + E_i(k)^2\right] \end{array} \tag{2.4}$$

$$\text{subject to} \quad \sum_{i=1}^{N} R_i(b_i) \leq B \tag{2.5}$$

where $A_i(k)$ represents the accumulated error and $\xi(k)$ maps the run-length position to zig-zag scan position. The problem can be solved with the same algorithms as the case of constrained DRS of Intra-coded pictures with the new definition of distortion $\hat{D}_i(b_i)$.

3) Unconstrained DRS

In unconstrained DRS, the breakpoint is denoted as a vector with binary elements, $\mathbf{b}_i = \{b_i^k\}$, where $i \in \{1,2,\cdots,N\}$ and $k \in \{1,2,\cdots,K\}$. The problem formulation for Intra-coded pictures is:

$$\text{minimize} \quad \sum_{i=1}^{N} D_i(\mathbf{b}_i), \text{ where } D_i(\mathbf{b}_i) = \sum_{k} \bar{b}_i^k E_i(k)^2 \tag{2.6}$$

$$\text{subject to} \quad \sum_{i=1}^{N} R_i(\mathbf{b}_i) \leq B \tag{2.7}$$

And the problem formulation for Inter-coded pictures is:

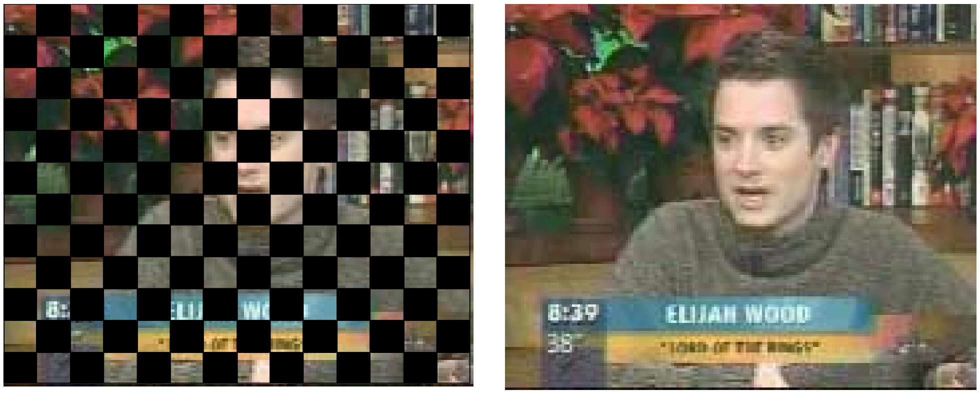
$$\text{minimize} \quad \begin{array}{l} \sum_{i=1}^{N} D_i(\mathbf{b}_i), \text{ where} \\ D_i(\mathbf{b}_i) = \sum_{k} A_i(k)^2 + \sum_{k} 2\bar{b}_i^k A_i(\xi_i(k))E_i(k) + \sum_{k} \bar{b}_i^k E_i(k)^2 \end{array} \tag{2.8}$$

subject to $$\sum_{i=1}^{N} R_i(\mathbf{b}_i) \le B \qquad (2.9)$$

The problems can be solved by Lagrange multiplier with bisection-based algorithm [19] or descent-based algorithm [20].

2.1.2. Rate Shaping by Block Dropping

Instead of dropping the DCT coefficients, rate shaping can also be achieved by block dropping and the additional error concealment at the receiver [65][66]. Rate shaping makes a decision on which DCT blocks to drop depending how much the distortion is if these blocks are to be reconstructed with error concealment ([64] and Figure 2) at the receiver.



(a) (b)

Figure 2. Geometric-structure-based error concealment for 50% block loss: (a) without error concealment; (b) with error concealment

The problem formulation is of Intra-coded pictures is:

minimize $$D(X, \overline{X}) \qquad (2.10)$$

subject to $$R(\overline{X}) \le B \qquad (2.11)$$

where X is the original image, $\hat{X}$ is the MPEG-quantized image with bit rate $R(\hat{X})$, and $\overline{X}$ is the shaped-followed-by-concealed version of $\hat{X}$, with bit rate $R(\overline{X})$. The problem can be solved by tree pruning algorithm [11]. Similar to 2.1.1, the i th reconstructed frame $\overline{X}_i$ consists of the motion compensated result from the previous frame $M_i\left(\overline{X}_{i-1}\right)$, where $M_i(\cdot)$ denotes the motion compensation, and the reconstructed coefficients from the current frame $\overline{e}_i$.

$$\overline{X}_i = M_i\left(\overline{X}_{i-1}\right) + \overline{e}_i \tag{2.12}$$

After the modification of (2.12), the rest rate-distortion optimization follows.

The above two types of conventional rate shaping select to drop either some of the transform coefficients or the blocks in order to adapt the bit rate of the pre source-coded (no channel-coded) video. The conventional rate shaping however is not suitable for video streaming over packet-loss networks, where the precoded video is both source- and channel- coded. We will describe in the next section some characteristics of packet-loss networks.

2.2. Video Transport over Packet-Loss Networks

Packet-loss networks are generally with time-varying packet loss rate and fluctuating bandwidth. Bandwidth can be the bandwidth of circuit-switched networks, or is provided by the network management layer. It can also be the estimated effective bandwidth. In the study, we regard the bandwidth constraints as the target bit rates the output bitstream is trying to satisfy. Packet losses are due to two reasons: the packets never arrive (or arrive too late over a certain threshold) and the arrived packets contain bit errors. In the study, we focus on the second packet-loss scenario since the first one usually results from router queue overflows, packet reordering, etc., that are beyond the scope of forward error correction (FEC) codes.

The sources of bit errors in a wireless channel are noise, shadowing, fading, intersymbol interference, etc. We adopt a simple finite-state Markov chain for wireless channel bit error simulation (detailed in Appendix B). To derive some interesting results about how the size of the packet s and the transition probability (equivalently the burstiness) affect the packet loss rate e_p, we simplify the model to make $e_G = e_1 = 0$, $e_B = e_0 = 1$, $t_{1,0} = p$, and $t_{0,1} = q$. The mean bit error rate (BER) e_b is related to the transition probabilities p and q by $e_b = p/(p+q)$. With bit error rate e_b, transition probability p, and packet size s, the packet loss rate of the s-bit packet is,

$$e_p = 1 - (1 - e_b)(1 - p)^{s-1} \tag{2.13}$$

We observe two properties from (2.13) 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 3 with $e_b = 10^{-4}$. We will see the use of these properties in the later chapters.

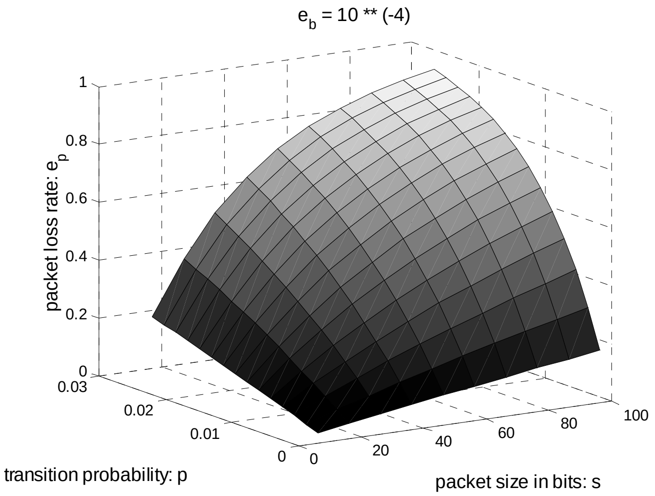


Figure 3. Packet loss rate as a function of the transition probability and the packet size

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) herein. Users can select the packet size s according to real system consideration.

2.3. Error-Resilient Rate Shaping: Baseline Rate Shaping (BRS)

To protect the video from transmission errors, source-coded video bitstream is often protected by forward error correction (FEC) codes [46][61]. Redundant information, known as parity bits, is added to the original source-coded bits. Parity bits are included in the precoded video because FEC encoding at the time of transmission may not be feasible given the capability of the node that is transporting the video. On the other hand, this node should be able to perform rate shaping for both the source- and channel- coded bitstream since rate shaping has less complexity than full decoding. This node is able to perform full decoding if it wants to view the content of the video.

Error-resilient rate shaping is in need to adapt the bit rates of the *pre source- and channel- coded* video. The adaptation of bit rates is natural for wireless transmission in the wireless LAN etc., given the fluctuating characteristic of the channel rates. It is also known that the devices used as clients of streaming applications vary a lot in their computation powers, connection bandwidths, etc.

The proposed error-resilient rate shaping can be performed either at the source (the video server), at the application-aware network node (the proxy), or at the receiver, as shown in Figure 4. It is worth noted that, unlike joint source-channel techniques that allocate the bits for the source and channel coders to achieve the best video quality, the proposed error-resilient rate shaping performs the rate adaptation for the precoded video at the time of delivery. The decision, as to select which part of the precoded video to drop, varies from time to time. There is no need to reassign bits to the source and channel coders as proposed by the joint source-channel techniques. In addition, rate shaping can be applied to adapt to the network condition of each link along the path of transmission. This is in particular suitable for wireless video transport, since wireless networks are heterogeneous in nature. One single joint source-channel coded bitstream cannot be optimal for all links along the path of transmission from the sender to the receiver. Rate shaping on the other hand can optimize the video transport of each link.

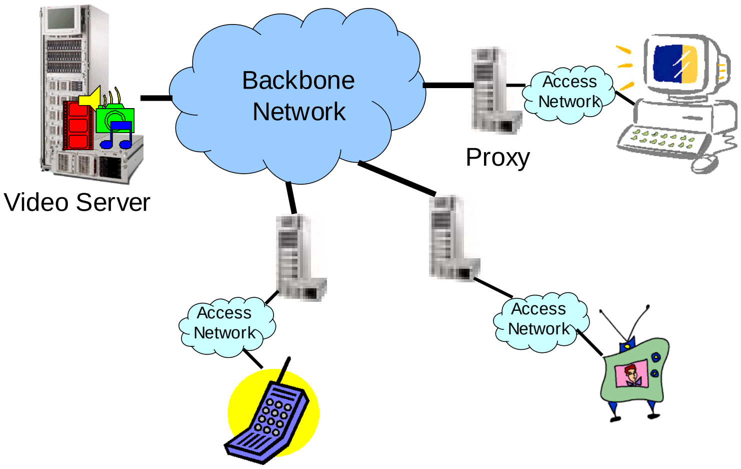


Figure 4. A general video transport system

We start introducing the error-resilient rate shaping with the baseline approach, "baseline rate shaping (BRS)". More advanced error-resilient rate shaping will be introduced in the following chapters. BRS system will be illustrated first followed by the rate-distortion (R-D) optimization algorithm for BRS.

2.3.1. *System Description of Video Transport with 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 5 to Figure 7.

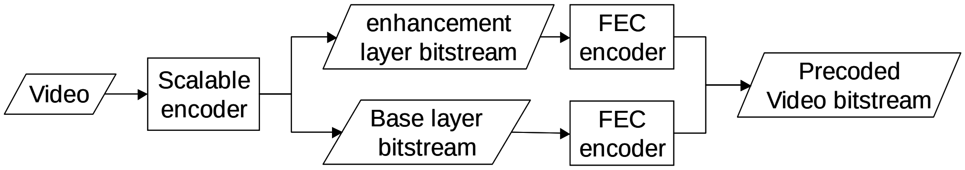


Figure 5. System diagram of the precoding process: scalable encoding followed by FEC encoding

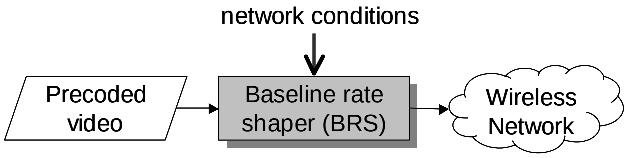


Figure 6. Transport of the precoded video with BRS

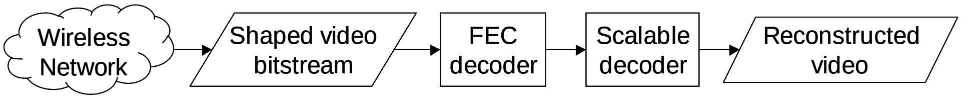


Figure 7. System diagram of the decoding process: FEC decoding followed by scalable decoding

The precoding process (Figure 5) consists of source coding using scalable video coding [22][40][52] and FEC coding. Scalable video coding provides the prioritized bitstream for rate shaping. The concept of rate shaping works for any prioritized video bitstream in general[1]. Without loss of generality, we consider using signal-to-noise-ratio (SNR) scalability. We use Reed-Solomon codes [61] as the FEC codes.

In Figure 6, the pre source and channel coded bitstream is then passed through BRS to adjust the bit rate before being sent to the wireless networks. BRS seeks to perform the best bandwidth adaptation at the given packet loss rate. The distortion here is described by the distortion in peak-signal-to-noise-ratio (PSNR). Packet loss rate, instead of bit error rate, is

[1] For example in DRS, prioritized video bitstream from high to low priorities, is offered by low to high frequency DCT coefficients. Data partitioning for the single-layered non-scalable coded bitstream can also give the prioritized bitstream.

considered since the shaped precoded video will be transmitted in packets. In summary, considering the packet loss rate and the bandwidth, BRS reduces the bit rate of the precoded video in the R-D optimized manner (will be elaborated later).

The decoding process (Figure 7) consists of FEC decoding followed by scalable decoding.

2.3.2. Algorithms for BRS

Rate-distortion (R-D) optimization algorithms are taken by BRS to deliver the best video quality. We will describe in the following two R-D optimization algorithms: BRS by mode decision and BRS by discrete R-D combination, depending on how much delay rate shaping allows.

2.3.2.1. BRS by Mode Decision

Let us consider the case in which the video sequence 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) capabilities. 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 8 (a)). The second segment consists of the bits of the enhancement layer video bitstream (upper right segment of Figure 8 (a)). The third segment consists of the parity bits for the base layer video bitstream (lower left segment of Figure 8 (a)). The fourth segment consists of the parity bits for the enhancement layer video bitstream (lower right segment of Figure 8 (a)). BRS decides a subset of the four segments to send. There is some constraint to yield a valid combination. 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 this case with two layers of video bitstream, there are six valid combinations shown in Figure 8 (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. The two integers x and y satisfy the relationship of $x \geq y$.

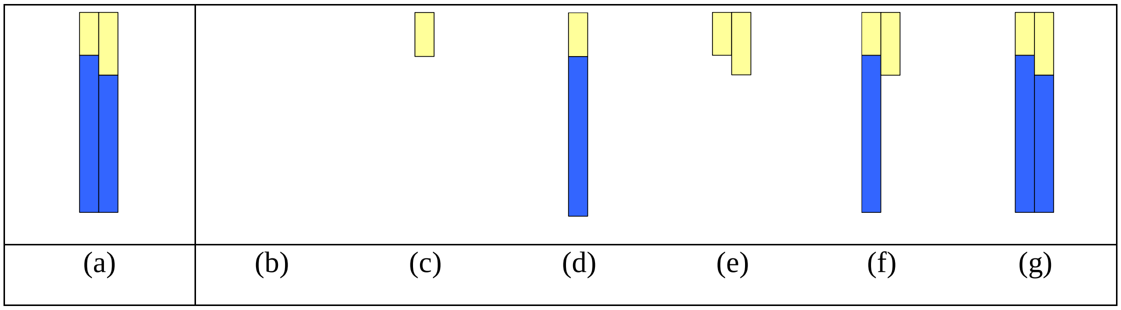


Figure 8. (a) All four segments of the precoded video and (b)~(g) available states for BRS: (b) state (0,0), (c) state (1,0), (d) state (1,1), (e) state (2,0), (f) state (2,1), and (g) state (2,2)

Each state of a frame has its R-D performance represented by a dot in the R-D map shown in Figure 9 (a) or (b), where B represents the bandwidth constraint. The constellations of state R-D performances of different frames are different because of variations of the video source and the packet loss rate. If the bandwidth requirement “B” of each frame is given and used, BRS performs mode decision by selecting the state that gives the least distortion. For example in Figure 9, state (1, 1) of Frame 1 and state (2, 0) of Frame 2 are chosen.

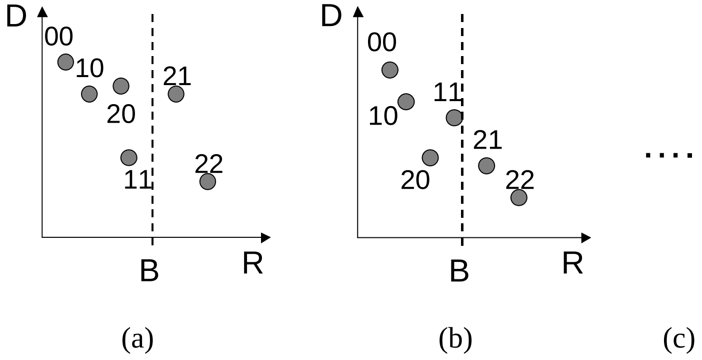


Figure 9. R-D maps of: (a) Frame 1, (b) Frame 2, and so on

2.3.2.2. BRS by Discrete R-D Combination

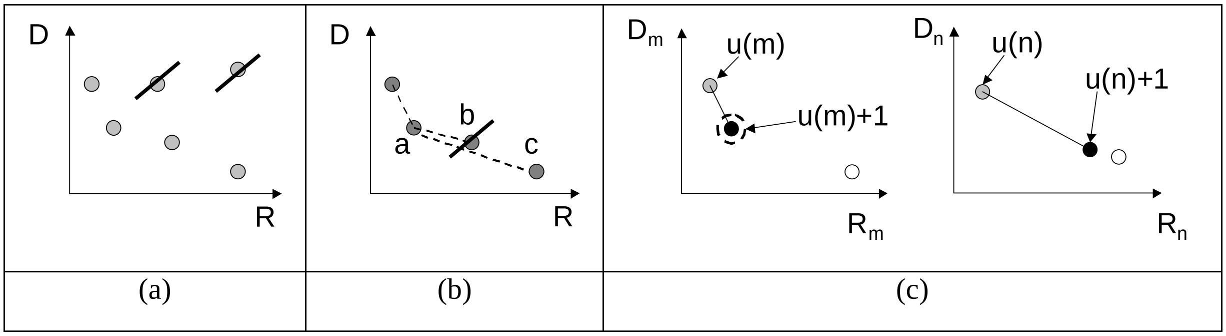
By allowing for some delay in making the rate shaping decision, BRS can deliver the precoded video with a better quality. By allowing for delay, we mean to accumulate the total bandwidth budget for a group of pictures (GOP) and to allocate the bandwidth intelligently among frames in a GOP. Video bitstream is typically coded with variable bit rate in order to maintain a constant video quality. Therefore, we want to allocate different number of bits to 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:

$$\text{minimize} \qquad \sum_{i=1}^{F} D_{i,x(i)} \tag{2.14}$$

$$\text{subject to} \qquad \sum_{i=1}^{F} R_{i,x(i)} \le C \tag{2.15}$$

The discrete R-D combination algorithm [7][43] finds the solution by first eliminating the states that are inside the convex hull of states (Figure 10 (a) and (b)) for each frame. The algorithm then allocates the rate step by step to the frame that can utilize 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 10 (c)) from the available total bandwidth budget. The allocation process continues until the total bandwidth budget has been consumed completely.



(a) (b) (c)

Figure 10. 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

To summarize for this chapter, we introduce the conventional rate shaping that is applied to the pre source-coded video, brief the characteristics of packet-loss networks, and provide the baseline approach, BRS, of the proposed error-resilient rate shaping.

3. Rate Shaping for Enhancement Layer Video

The baseline approach BRS of the proposed error-resilient rate shaping makes decisions on a coarse level. One out of six states, is selected by BRS for streaming the precoded video. To incorporate a finer granularity, we propose “fine-grained rate shaping (FGRS)” for streaming the enhancement layer video and “error-concealment aware rate shaping (ECARS)” for streaming the base layer video. We will talk about FGRS in this chapter and ECARS in the next chapter.

We adopt MPEG-4 fine granularity scalability (FGS) [32] for source coding, and erasure codes [46][61] for FEC coding. Unlike conventional scalability techniques such as SNR scalability, MPEG-4 FGS provides the video bitstream that is partially decodable over a wide range of bit rates. The more bits of the FGS bitstream is received, the better the video quality is. In addition, it has been known that partial FEC coded bitstream is still decodable within the error correction capability if erasure codes are used. Thus, both FGS and erasure codes provide fine-granularity properties in video quality and in packet-loss protection. Given the FEC coded FGS bitstream as the precoded video, “fine-grained rate shaping (FGRS)” is proposed for bandwidth adaptation considering the current packet-loss rate. There are conceptually infinitely many possible combinations of dropping portion of the FGS bitstream and portion of the FEC codes. FGRS seeks the optimal solution in the R-D sense. A new two-stage R-D optimization is proposed to select part of the precoded video to drop.

The proposed “two-stage R-D optimization” aims for both efficiency and optimality by using model-based hyper-surface and hill-climbing based refinement. In Stage 1, a model-based hyper-surface is first trained with a set of rate and gain pairs. We then find the solution that sits in the intersection of the hyper-surface and the bandwidth constraint. In Stage 2, the near-optimal solution from Stage 1 (because the model can only approximate the true relationship between rate and gain) is then 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 chapter is organized as follows. We first introduce the system of FGRS. Background materials as MPEG-4 FGS and Reed-Solomon codes will also be mentioned. We then elaborate on algorithms for FGRS, with the R-D problem formulation followed by the two-stage R-D optimization. Experiments are carried out to show the superior performance of the proposed FGRS to naïve unequally packet-loss protection methods. Finally, concluding remarks are given.

3.1. Rate Shaping for Enhancement Layer Video: 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 the 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 would like to extend the notion of BRS to allow for finer-grained decisions. There 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. That is, FGS enhancement layer bitstream is decodable at any bit rate over a wide range of values. With such a property, FGS was adopted by MPEG-4 for streaming applications [32]. Through FGS encoding, two layers of bitstream are generated: one base layer and one enhancement layer (Figure 11). The base layer is predictive coded while the enhancement layer only uses the corresponding base layer as the reference.

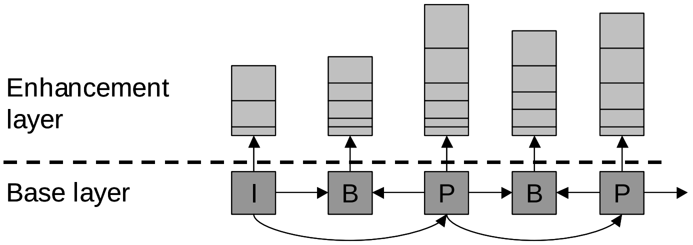


Figure 11. Dependency graph of the FGS base layer and enhancement layer. Base layer allows for temporal prediction with P and B frames. Enhancement layer is encoded with reference to the base layer only

On the other hand, it has also been known that the erasure codes provide "fine-grained" packet-loss protection with more and more symbols[2] received at the FEC decoder [46][61]. The "shaped"

[2] "Symbols" are used instead of "bits" since the Reed-Solomon codes used in the study use a symbol as the encoding/decoding unit. We use 14 bits to form one symbol. The selection of symbol size in bits is up to the user.

erasure code is still decodable if the number of erasures/losses from the transmission is no more than $d_{\min} - 1 - (\text{number of unsent symbols})$. 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.

We use Reed-Solomon codes as the erasure codes. 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 of size $r \le n$ is still decodable if the number of losses from the transmission is no more than $r - k$.

After understanding the background materials of MPEG-4 FGS and Reed-Solomon codes, let us introduce the system for streaming the precoded video. As 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 12 to Figure 14.

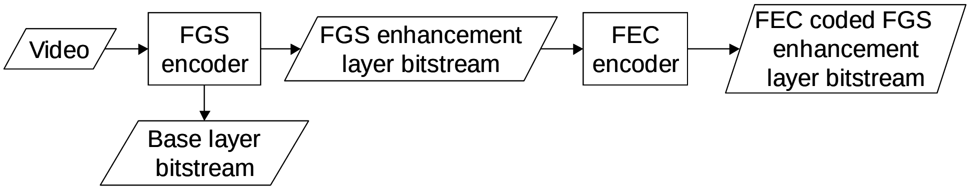

Video
FGS encoder
FGS enhancement layer bitstream
FEC encoder
FEC coded FGS enhancement layer bitstream
Base layer bitstream

Figure 12. System diagram of the precoding process: FGS encoding followed by FEC encoding

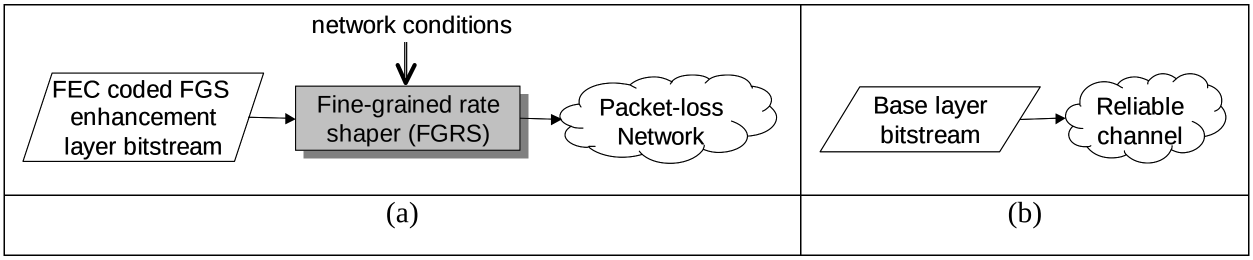


Figure 13. Transport of the precoded bitstreams: (a) transport of the FEC coded FGS enhancement layer bitstream with rate shaper via the wireless network, and (b) transport of the base layer bitstream via the secure channel

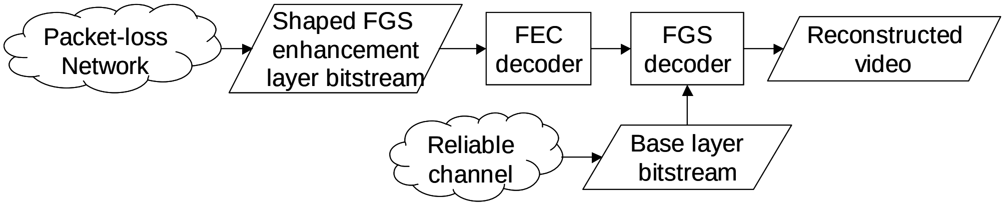


Figure 14. System diagram of the decoding process: FEC decoding followed by FGS decoding

Through FGS encoding, two layers of bitstream are generated: one base layer and one enhancement layer (Figure 11). 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 13 (b).

Let us look at the FGS enhancement layer bitstream for a frame. FGS enhancement layer bitstream consists of bits of all the bit-planes for 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 the bit-plane is corrupted (due to packet losses), the remaining part of the bit-plane becomes un-decodable. The importance of the bits of the enhancement layer decreases from the beginning to the end.

Before appending the parity symbols to the FGS enhancement layer bitstream, we first divide all the symbols for this frame into several sublayers (Figure 15 (a)). The way to divide the symbols into sublayers is arbitrary except that the later sublayers are longer than the previous ones, $k_1 \geq k_2 \geq \cdots \geq k_h$, since we want to achieve unequal packet loss protection (UPP). A natural way of division 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 15 (b)). The precoded video is stored and can be used at the time of delivery.

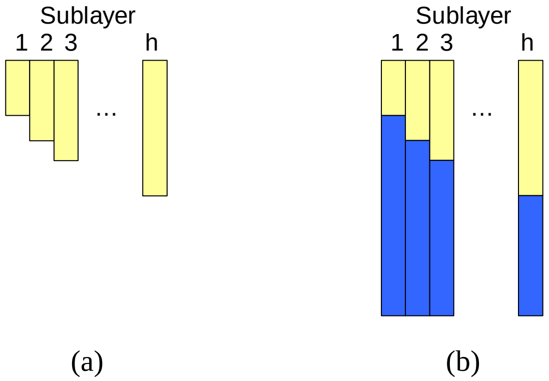


Figure 15. Precoded video: (a) FGS enhancement layer bitstream in sublayers, and (b) FEC coded FGS enhancement layer bitstream

At the transport stage, FEC coded FGS bitstream is passed through FGRS for bandwidth adaptation under the current packet loss rate. Note again that FGRS is different from joint source-channel coding based 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 van der Schaar and Radha [55] and Yang et al. [63]. Packetization is performed after error-resilient rate shaping.

3.2. Algorithms for FGRS

With the precoded video, bandwidth adaptation can be achieved by methods shown in Figure 16. The dark bars in Figure 16 (a) and Figure 16 (d) are selected to be sent. Figure 16 (a) shows how to adapt the bandwidth by randomly dropping part of the precoded video (or randomly keeping part of the precoded video). Bandwidth adaptation can also be achieved by naïvely dropping the symbols in the order shown in Figure 16 (b). 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 naive bandwidth adaptation scheme has UPP to the sublayers. We will refer to this method as “UPPRS1” hereafter. In addition, bandwidth adaptation can be achieved by first dropping the symbols from higher sublayers as shown in Figure 16 (c), which we refer to as “UPPRS2” hereafter. However, none of the above methods are optimal. We propose FGRS (Figure 16 (d)) for bandwidth adaptation given the current network condition.

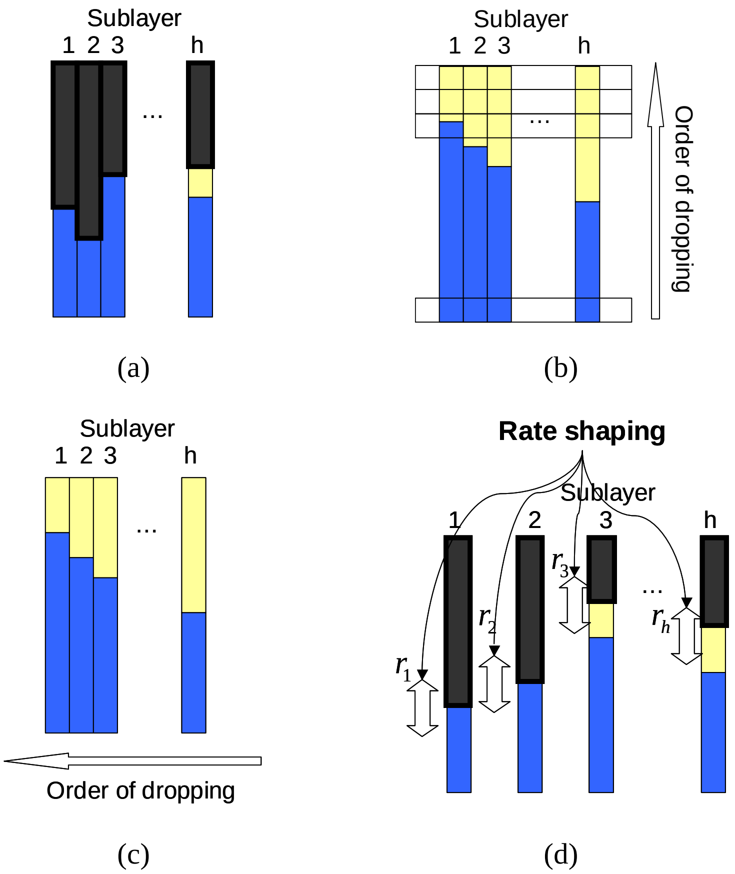


Figure 16. Bandwidth adaptation with (a) random dropping; (b) UPPRS1; (c) UPPRS2; and (d) FGRS

Let us start from the problem formulation and continue with the two-stage R-D optimization to solve for the FGRS problem.

3.2.1. 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 G_i of Sublayer i can either (1) be calculated given the FGS bitstream, after performing partial decoding in order to get the values of gain; or (2) 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:

$$G = \sum_{i=1}^{h} \left(G_i \prod_{j=1}^{i} v_j \right) \tag{3.1}$$

where h is the number of sublayers of this frame, and v_j is the recovery rate of Sublayer j that is a function of r_j as 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 in Sublayer j, and r_j is the number of symbols selected to be sent in Sublayer j. With Reed-Solomon codes used, $r_i \geq k_i$ with the exception of the last sublayer (not necessary the Sublayer h, can be the sublayer before that); and the whole sublayer is considered lost if the number of erasures is beyond the error-correction capability $r_i - k_i$.

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.

$$v_j = \sum_{l=0}^{r_j - k_j} p\{l \text{ erasures occur}\}, \; j = 1 \sim h \tag{3.2}$$

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

$$p\{l \text{ erasures occur}\} = \binom{r_j}{l} (e_m)^l (1 - e_m)^{r_j - l} \tag{3.3}$$

The symbol loss rate can be derived from the packet loss rate as $e_m = 1 - \left(1 - e_p\right)^{m/s}$, where s is the packet size and m is the symbol size in bits.

If the erasures come from a finite-state Markov model, for example, a two-state Markov model with symbol loss rates $e_{m,1}$ and $e_{m,2}$, the probability of having l erasures out of r_j symbols is:

$$p\{l \text{ erasures occur}\} = \sum_{l_1=0}^{l} \left[\binom{r_{j,1}}{l_1} \left(e_{m,1}\right)^{l_1} \left(1 - e_{m,1}\right)^{r_{j,1} - l_1} \right] \left[\binom{r_j - r_{j,1}}{l - l_1} \left(e_{m,2}\right)^{l_2} \left(1 - e_{m,2}\right)^{r_{j,2} - l_2} \right] \quad (3.4)$$

where $r_{j,1}$ is the number of symbols out of r_j symbols that originates from State 1 of the two-state Markov model, and $r_{j,2}$ is the number of symbols out of r_j symbols that originates from State 2 of the two-state Markov model. We can see that (3.4) is the convolution of two binomial distributions.

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

$$\text{maximize} \qquad G = \sum_{i=1}^{h} \left(G_i \prod_{j=1}^{i} v_j \right) \quad (3.5)$$

$$\text{subject to} \qquad \sum_{i=1}^{h} r_i \le B \quad (3.6)$$

where B is the bandwidth constraint of this frame.

To solve this problem, we propose a new two-stage R-D optimization approach. The two-stage R-D optimization first finds the near-optimal solution globally. The near-optimal global solution is then refined by a hill climbing approach. Prior work on R-D optimization includes [12][43][45][50]. The proposed two-stage R-D optimization is different from [12][43][45][50] in three folds. First, the model-based Stage 1 allows us to examine fewer samples from all the operational R-D states. Second, the proposed distortion measure (or “expected accumulated gain” in the terminology of the thesis) 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 approach can avoid the potential problem that the solution could be trapped in the local maximum or reach the local maximum very slowly.

Packetization is performed after error-resilient 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. Small packet is desirable to make use of the fine-grained decision resulted from FGRS. For example, a big packet that contains all the symbols from a frame could be unrecoverable if it is decided to be dropped by the lower layers (for example, the link layer detects a CRC check error for this big packet).

3.2.2. Two-Stage R-D Optimization: Stage 1

We can see from (3.1) to (3.4) 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 (3.1) to (3.4). The optimal solution is the feasible solution within the intersection of the hyper-surface and the bandwidth constraint as illustrated in Figure 17. 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 from the intersection 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.

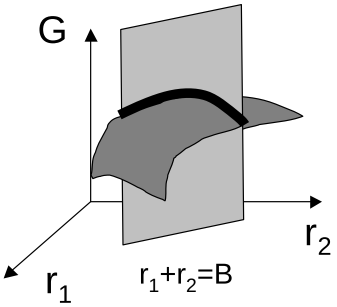


Figure 17. Intersection of the model-based hyper-surface (dark surface) and the bandwidth constraint (gray plane), illustrated with $h = 2$

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

$$\hat{G} = \sum_{i=1}^{h} a_i r_i^2 + \sum_{i,j=1, i \neq j}^{h} b_{ij} r_i r_j + \sum_{i=1}^{h} c_i r_i + d \tag{3.7}$$

To distinguish the hyper-surface model $\hat{G}$ 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 (3.1)-(3.4). For example, the parameters can be computed by:

$$\begin{pmatrix} a_i\text{'s} \\ b_{ij}\text{'s} \\ c_i\text{'s} \\ d \end{pmatrix} = \left(R^T R\right)^{-1} R^T \begin{pmatrix} {}^1G \\ {}^2G \\ \vdots \\ {}^{\Xi}G \end{pmatrix} \tag{3.8}$$

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 (3.8). 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, will not be suitable since it requires much more training data. In addition, Second-order Taylor expansion can approximate nicely in general every function. (3.7) can be seen as a second-order approximation to (3.1). To reduce the computation complexity in reality, we can also choose a smaller h.

With (3.7), the near-optimal solution can be obtained by the use of Lagrange multiplier as follows.

$$J = \left(\sum_{i=1}^{h} a_i r_i^2 + \sum_{i,j=1, i \neq j}^{h} b_{ij} r_i r_j + \sum_{i=1}^{h} c_i r_i + d \right) + \lambda \left(\sum_{i=1}^{h} r_i - B \right) \tag{3.9}$$

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

$$r_i = \frac{-1}{2a_i} \left(\sum_{j=1, j \neq i}^{h} b_{ij} r_j + c_i + \lambda \right) \tag{3.10}$$

where λ is:

$$\lambda = \frac{2B + \sum_{i=1}^{h} \frac{1}{a_i}\left(\sum_{j=1, j\neq i}^{h} b_{ij} r_j + c_i \right)}{-\sum_{i=1}^{h} \frac{1}{a_i}} \tag{3.11}$$

The near-optimal solution can be solved recursively starting from the initial condition that all sublayers are allocated with equal number of symbols, $r_1 = r_2 = \cdots = r_h = \frac{B}{h}$ using (3.10) and (3.11).

3.2.3. 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 18). The solution from Stage 1 is perturbed 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)
    $z_i$ = $r_i$ for all i=1~h
    For (j=1; j<=h; j++)
        For (k=1; k<=h; k++)
            $z_k$ = $z_k$ + delta for k==j //Increase sublayer j
            $z_k$ = $z_k$ - delta/(h-1) for k!=j //Decrease others
        End-for
        Evaluate $G_j$
    End-for
    Find the j* with the largest $G_{j*}$.
    For (i=1; i<=h; i++)
        $r_i$ = $r_i$ + delta for i==j*
        $r_i$ = $r_i$ - delta/(h-1) for i!=j*
    End-for
    Calculate the stop criterion.
End-while
```

Figure 18. Pseudo-codes of the 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 (3.5)-(3.6) as follows:

$$\text{maximize} \qquad G = \sum_{m=1}^{F}\left[\sum_{i=1}^{h}\left(G_{mi} \prod_{j=1}^{i} v_{mj} \right)\right] \tag{3.12}$$

subject to $$\sum_{m=1}^{F}\sum_{i=1}^{h} r_{mi} \le C \tag{3.13}$$

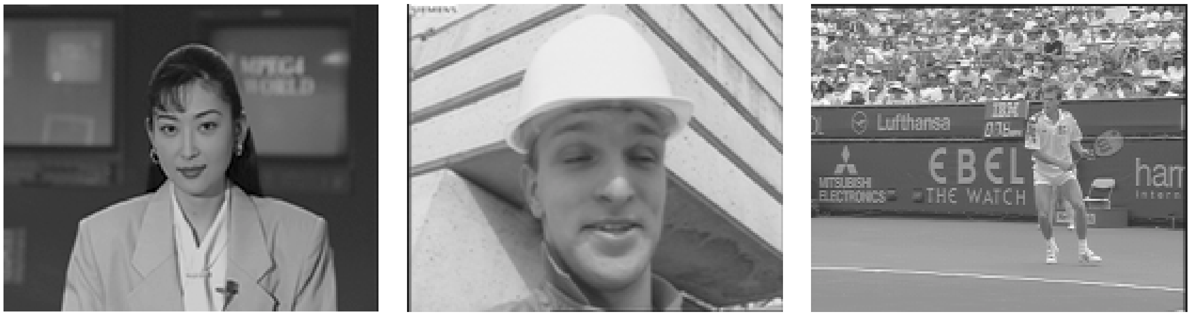
where F is the number of frames in a GOP. FGRS will incur delay with duration of F frames if it allows for optimization among frames in a GOP.

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 globally, then refining the solution with the hill-climbing based approach.

3.3. Experiment

We will show in this section the effectiveness of FGRS in streaming the precoded video over packet-loss networks. Four methods (mentioned in Figure 16) will be compared side-by-side: random dropping (with legend “rand”), UPPRS1 (with legend “upprs1”), UPPRS2 (with legend “upprs2”), and FGRS (with legend “fgrs”).

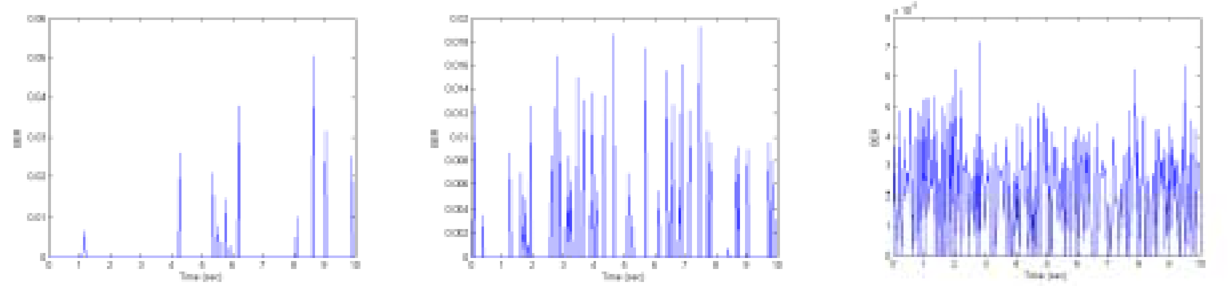
The test video sequences are “akiyo”, “foreman”, and “stefan” in common intermediate format (CIF) (Figure 19 (a)-(c)). Sequence “akiyo” represents a video sequence with lower bit rate due to simpler texture and less motion. Sequence “foreman” represents a video sequence with medium bit rate with regular texture and motion. Sequence “stefan” represents a video sequence with higher bit rate with complex texture and faster motion. The frame rate of MPEG-4 FGS coding is three frames/sec. The source-coding rates of the FGS enhancement layer bitstream of the three sequences are 354.69 kbits/sec, 747.74 kbits/sec, and 975.70 kbits/sec. The FEC coded bitstreams (before being shaped) of these three sequences have rates 3440.5 kbits/sec, 5275.5 kbits/sec, 6147.1 kbits/sec, respectively.



(a) (b) (c)

Figure 19. Test video sequences in CIF: (a) akiyo, (b) foreman, and (c) stefan

The bandwidth of the simulated networks fluctuates between 200 kbits/sec and 1100 kbits/sec. The bit error rate (BER) of the channel also fluctuates according to the two-state Markov chain model detailed in Appendix B. The wireless channel simulation parameters can be found in B.2. Some of the BER traces are shown in Figure 20. Under the same network condition (the same BER trace and the same bandwidth trace), the results shown in the following are tested for 10 different seeds for pseudo-random simulations. That is, the “overall PSNR” result shown is the average of 10 different tests. The frame-by-frame PSNR result is an instance out of of the 10 tests.



(a) (b) (c)

Figure 20. Sample BER traces of the wireless channel: (a) mobile unit at 2 km/h; (a) mobile unit at 6 km/h; (a) mobile unit at 10 km/h

Given the *gain* embedded in the bitstream, FGRS consumes on the average <0.01% (the denominator is the bit rates of the source-coded bitstream) of the original precoded video to carry the sublayer gain information (“meta-data”). The performance improvement of FGRS in PSNR over non- rate shaping based methods is on the average 8 dB. On the other hand, if the gain is not embedded in the bitstream for rate shaping, no extra bits are needed to carry the sublayer gain information. Partial decoding to obtain the sublayer gain information is required.

In the following experiment results, we first show an example of how each method allocates the rates among sublayers (Figure 21). We then show the performance in terms of the overall PSNR of different sequences (from Figure 22 to Figure 24), and the performance in terms of the overall PSNR at various wireless channel conditions (from Figure 25 to Figure 27). Finally, we show the performance in terms of the frame-by-frame PSNR for sequence “foreman” (from Figure 28 to Figure 30).

Figure 21 shows that with the bandwidth constraint specified, Method “rand” allocates the rates randomly among the nine sublayers; Method “upprs1” allocates the rates equally among the nine sublayers; Method “upprs2” allocates the rates all to the first sublayer; and Method “fgrs” allocates the rates smartly among the nine sublayers (some sublayers are even not allocated with rates). The bit allocation process of FGRS happens automatically by the proposed two-stage R-D optimization considering the current network condition.

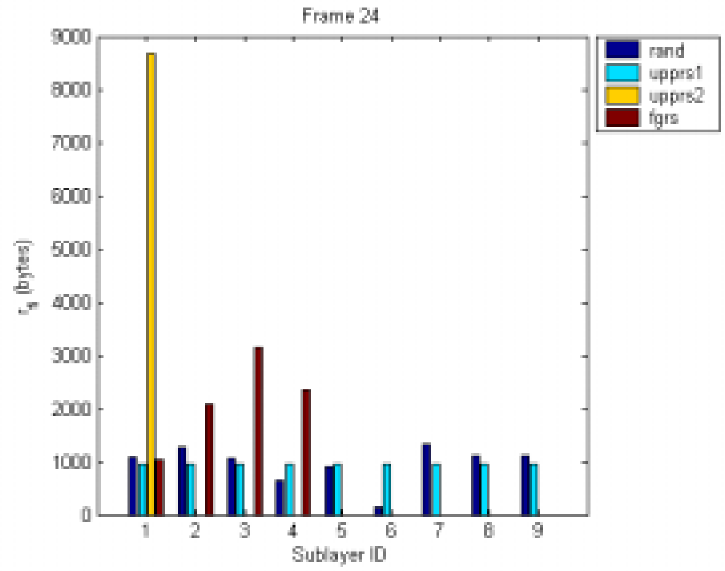


Figure 21. Sublayer bit allocations of all methods at 10 km/h and SNR=20 dB for Sequence “foreman”

From Figure 22 to Figure 24, the performance in terms of the overall PSNR of the Y, U, and V components, of different sequences is shown. We can see that for each sequence, for all Y, U, and V components, “fgrs” performs the best among all four methods. Given the same network condition, Sequence “akiyo” has higher PSNR than “foreman”; and Sequence “foreman” has higher PSNR than “stefan”. The sequence with texture that is more complex and faster motion, such as “stefan”, gives smaller PSNR value given the same bandwidth budget. Results are consistent for Y, U, and V, components.

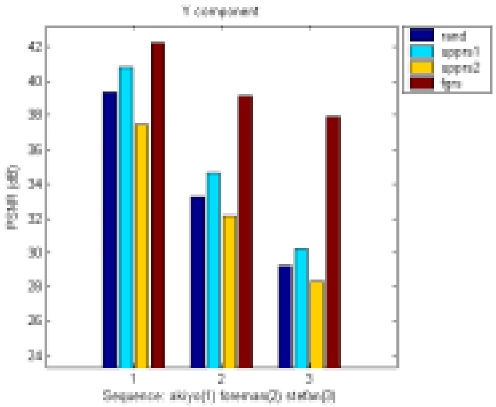


Figure 22. Performance (PSNR of the Y component) of all methods at 10 km/h and SNR=20 dB for Sequences "akiyo", "foreman", and "stefan"

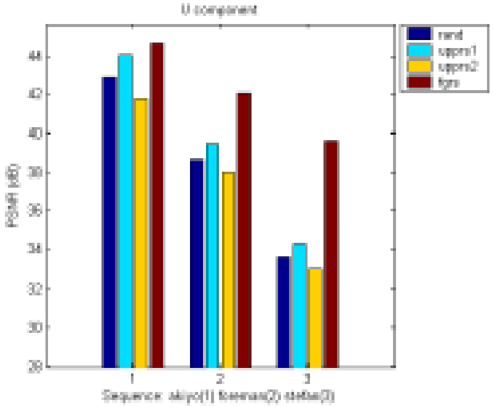


Figure 23. Performance (PSNR of the U component) of all methods at 10 km/h and SNR=20 dB for Sequences "akiyo", "foreman", and "stefan"

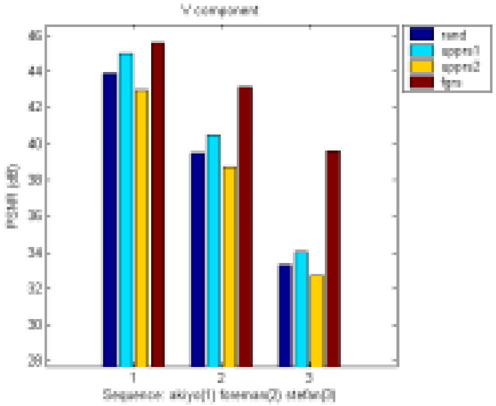


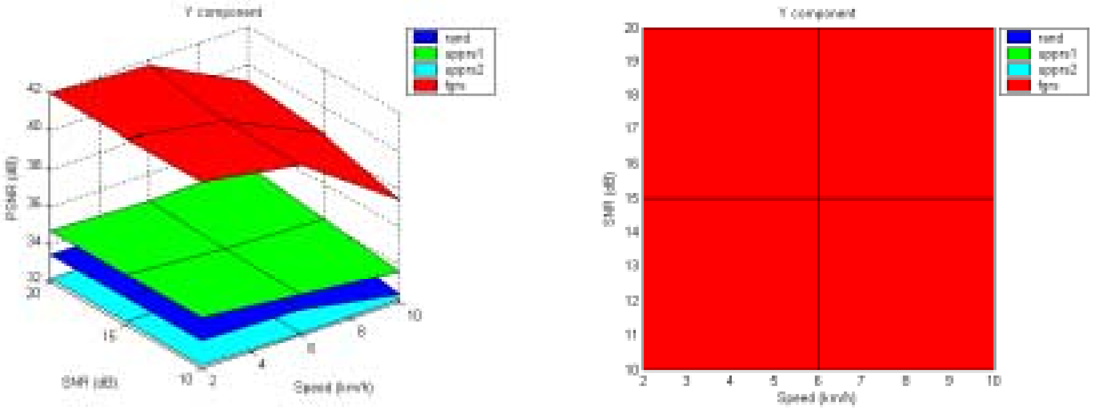
Figure 24. Performance (PSNR of the V component) of all methods at 10 km/h and SNR=20 dB for Sequences "akiyo", "foreman", and "stefan"

The performance in terms of the overall PSNR of the Y, U, and V components at various wireless channel conditions is shown from Figure 25 to Figure 27. Figure 25 (a), Figure 26 (a), and Figure 27 (a) show the 3-D plots of the overall PSNR. Figure 25 (b), Figure 26 (b), and Figure 27 (b) show the top views (seen from the top of the z-axis) of the 3-D plots. The color shown in the top view represents the color of the method that outperforms the others. At all wireless channel conditions, “fgrs” outperforms all other methods.

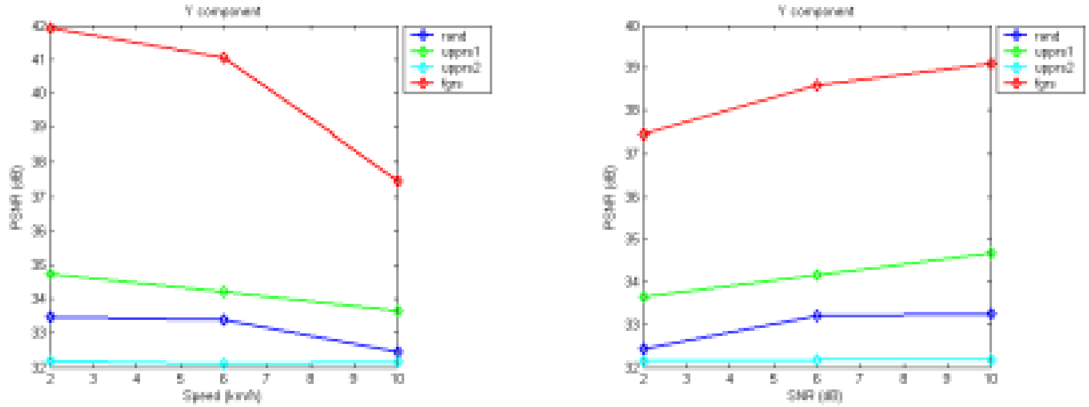
Figure 25 (c), Figure 26 (c), and Figure 27 (c) show the overall PSNR at various speeds at $\mathrm{SNR} = 10\,\mathrm{dB}$. Fixed SNR value gives the same bit error rate (BER) of the wireless channel. The higher the speed is, the more bursty the bit error of the wireless channel is. In other words, the larger the transition probability is. From the results, we see that the PSNR drops as the speed increases. This matches with what we have mentioned in Section 2.2 that 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 25 (d), Figure 26 (d), and Figure 27 (d) show 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. Also from Section 2.2, we know that the smaller the BER is, the smaller the packet-loss rate is, given the same burstiness. Smaller packet-loss rate then leads to a higher PSNR.

Results are consistent for Y, U, and V, components for all the figures shown from Figure 25 to Figure 27.

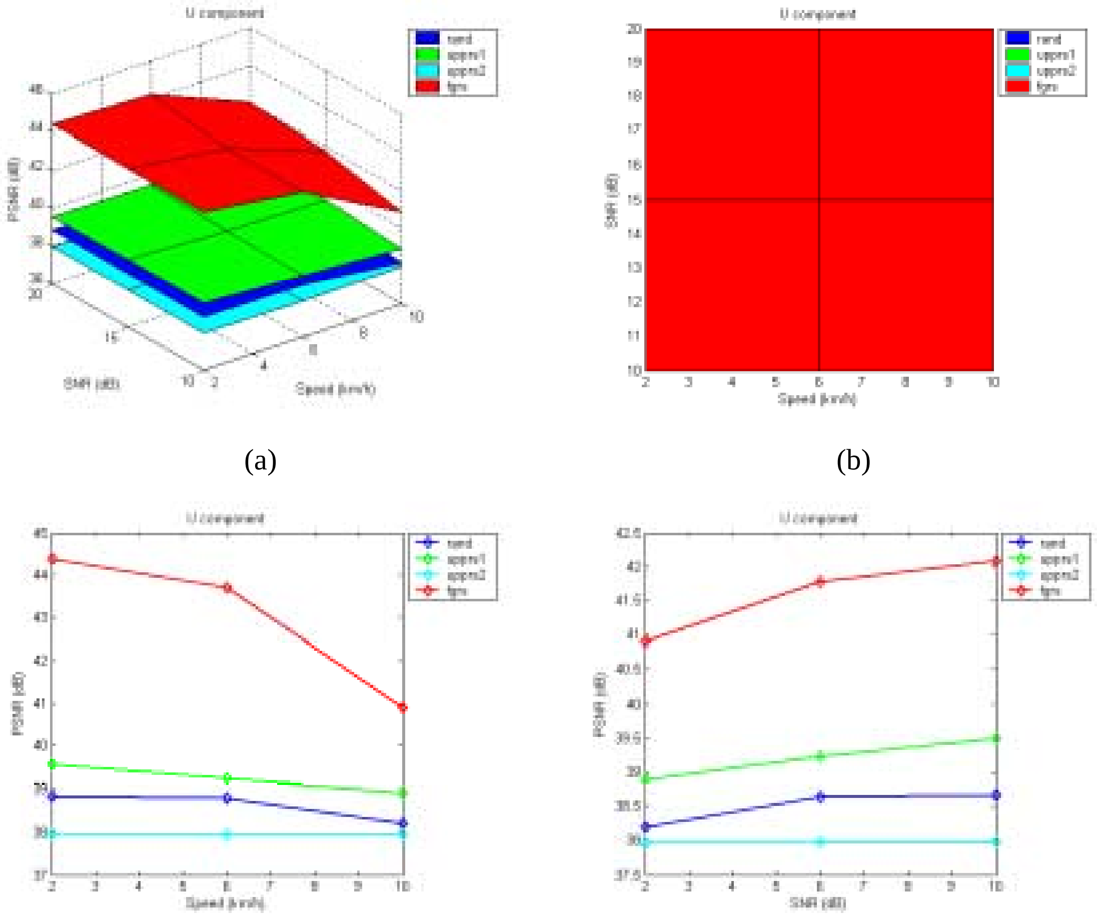


(a) (b)



(c) (d)

Figure 25. 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) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR



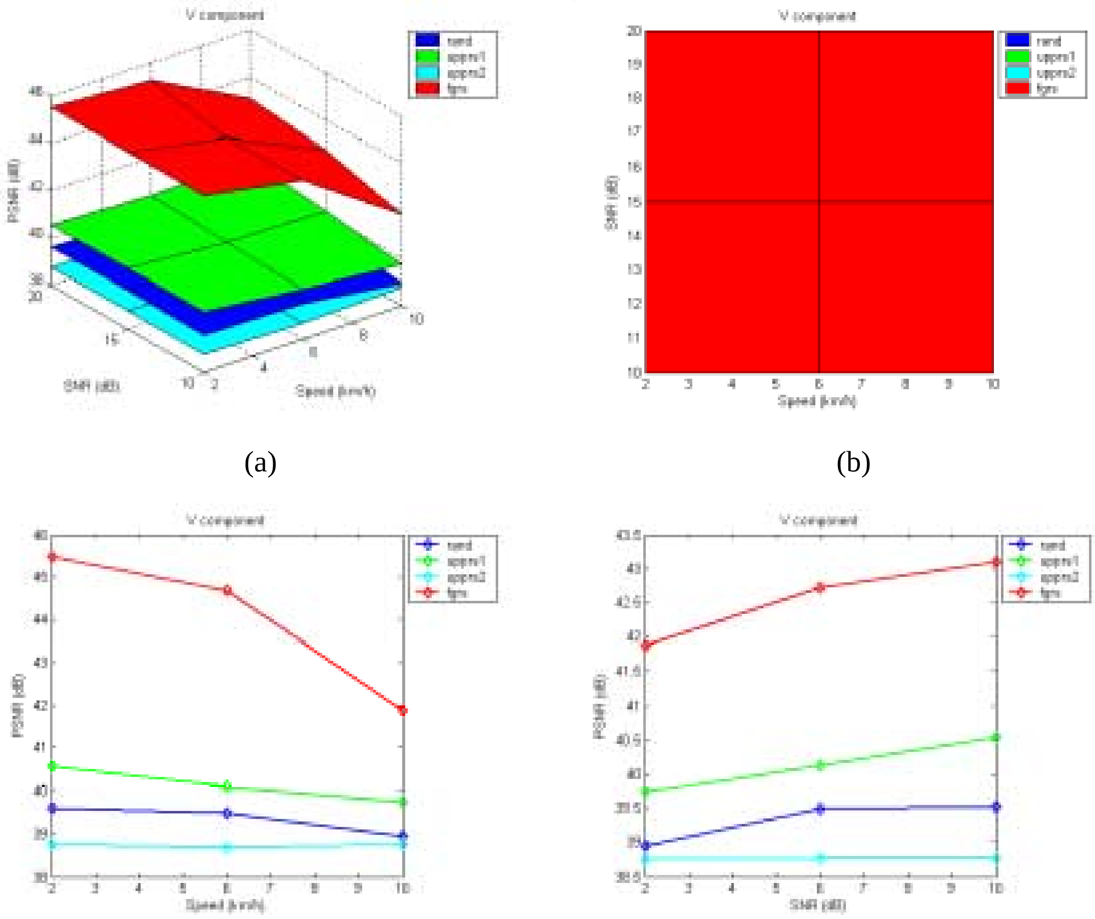
(a)

(b)

(c)

(d)

Figure 26. Performance (PSNR of the U component) of all methods at various wireless channel conditions for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR



(a)

(b)

(c)

(d)

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

Finally, we show the performance in terms of the frame-by-frame PSNR of the Y, U, and V components for sequence “foreman” (from Figure 28 to Figure 30). We see that “fgrs” performs the best among all. Sample frames of Method “upprs1” and “fgrs” are also shown in Figure 31 to demonstrate visually the merit of fine-grained rate shaping.

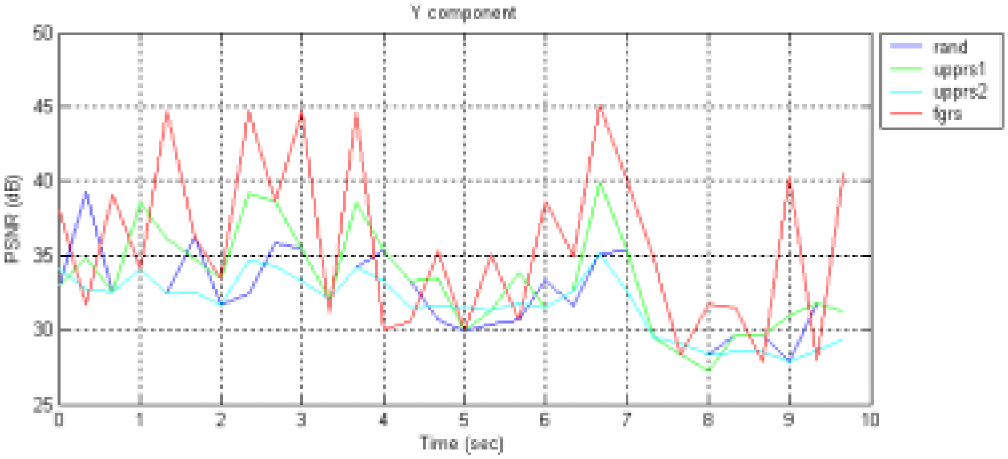


Figure 28. Frame-by-frame PSNR of the Y component of all methods at 10 km/h and SNR=20 dB for Sequence "foreman"

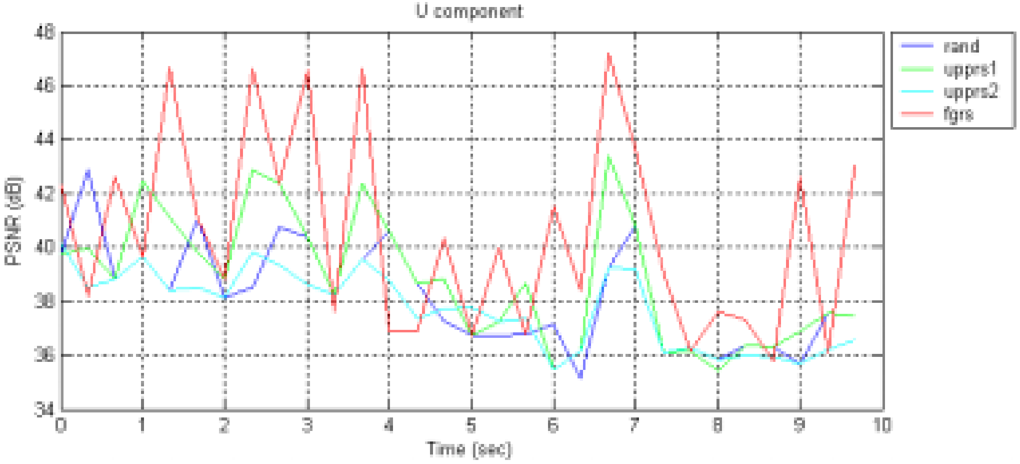


Figure 29. Frame-by-frame PSNR of the U component of all methods at 10 km/h and SNR=20 dB for Sequence "foreman"

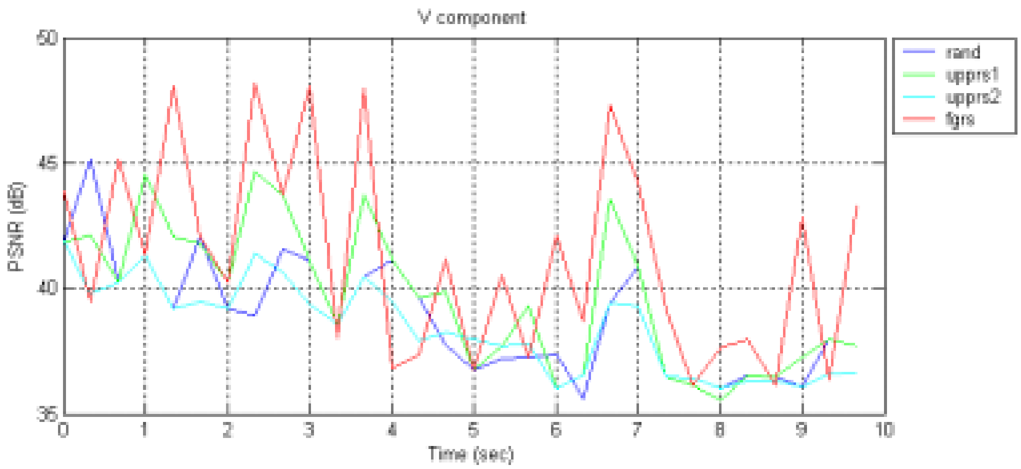
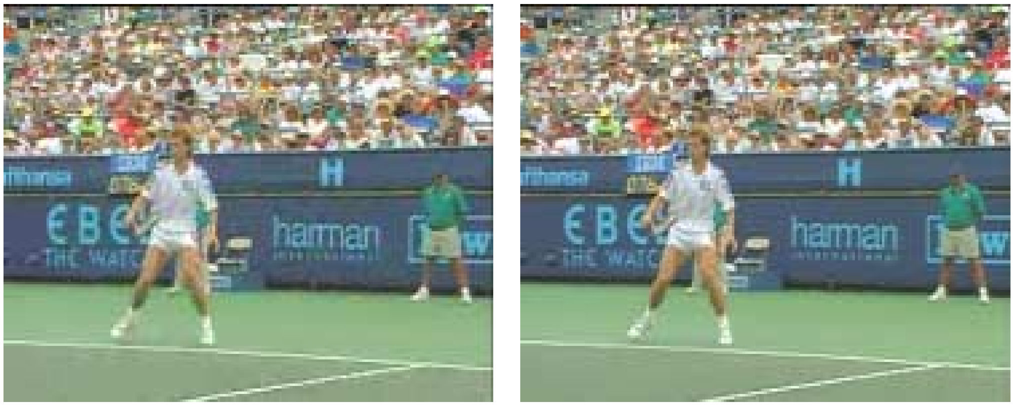


Figure 30. Frame-by-frame PSNR of the V component of all methods at 10 km/h and SNR=20 dB for Sequence "foreman"



(a)

(b)

Figure 31. A sample frame of (a) "upprs1" and (b) "fgrs" at 10 km/h and SNR=20 dB for Sequence "stefan"

3.4. Conclusion

To incorporate finer scalability to error-resilient rate shaping, we proposed fine-grained rate shaping (FGRS) for streaming the enhancement layer video, given that the base layer video is reliably transmitted. FGRS uses the proposed two-stage R-D optimization approach to adapt the rates of the FEC coded FGS enhancement layer bitstream, given the network condition. The two-stage R-D optimization first obtains the near-optimal solution that sits in intersection of the model-based hyper-surface and the bandwidth constraint. The near-optimal solution is then refined by a hill-climbing based approach. The two-stage R-D optimization aims for both the efficiency and the optimality. The proposed FGRS outperforms the other naïve methods.

4. Rate Shaping for Base Layer Video

To have a finer-granular decision instead of the coarse decision made by BRS, we propose "error-concealment aware rate shaping (ECARS)" for streaming the base layer video in addition to FGRS for streaming the enhancement layer video, illustrated in the last chapter. Taking into account that the receiver may perform error concealment (EC) if any video data is lost during the transmission, ECARS makes rate shaping decisions accordingly. Related work that utilized EC information for rate shaping on pre source- coded bitstream only can be found in [66].

Frame dependency is usually inherent in video bitstream due to predictive coding. In addition, temporal EC might introduce extra frame dependency. Feedback from the receiver to the sender might be helpful in addressing the frame dependency problem in rate shaping. We then introduce two types of ECARS algorithms: without feedback and with feedback. Both evaluate the *gains* of sending some parts of the precoded video as opposed to not sending them. The gain metrics are then included in the R-D optimization formulation. Finally, the two-stage R-D optimization approach is adopted to solve for the R-D optimization problem. In the case of no feedback, ECARS evaluates the gains considering a particular EC method used at the receiver. In order to incorporate the frame dependency into the rate shaping process, we propose to send the location (and mean) of the corrupted macroblock back to the sender, and use such feedback information to determine the gains used in the R-D optimized ECARS.

This chapter is organized as follows. We first introduce the system of ECARS. Background materials as EC methods and timely feedback follow. We then elaborate on algorithms for both types of ECARS: without feedback and with feedback. Experiments are carried out to show the performance of the proposed ECARS. Finally, concluding remarks are given.

4.1. Rate Shaping for Base Layer Video: Error Concealment Aware Rate Shaping (ECARS)

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 32 to Figure 34.

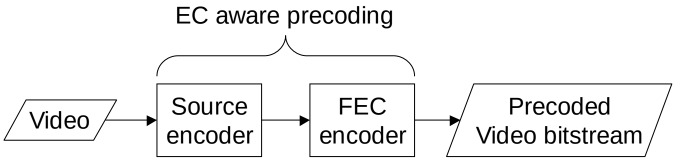


Figure 32. System diagram of the precoding process: source encoding (which can be EC aware) followed by FEC encoding

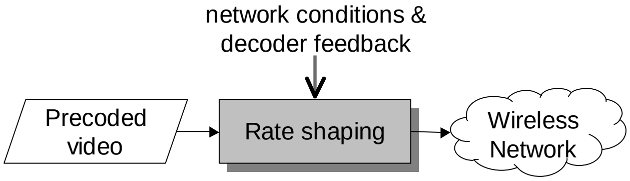


Figure 33. Transport of the precoded video with ECARS

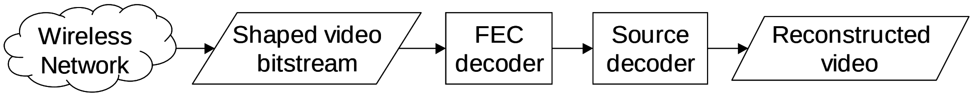


Figure 34. System diagram of the decoding process: FEC decoding followed by source decoding

In the precoding process (shown in Figure 32), video is encoded by both the source encoder and the FEC encoder. The precoding process is done before the time of delivery. The precoding process may be aware of the EC method used at the receiver, which we will describe later. In the streaming stage (shown in Figure 33), ECARS takes the network conditions as bandwidth and packet-loss rate, and possibly the feedback from the receiver, into account. The decoding process (shown in Figure 34) consists of FEC decoding followed by scalable decoding.

4.2. Background for ECARS

We will describe briefly on error concealment (EC) methods, EC aware precoding, and timely feedback in this section.

4.2.1. Error Concealment

EC relies on some *a priori* knowledge to reconstruct the lost video content. Such *a prior* can come from spatial or temporal neighbors. For example, we can assume that the pixel values are smooth across the boundary of the lost and retained regions. To recover lost data with the smoothness assumption, interpolation or optimization based on certain objective functions are often used. Figure 35 and Figure 36 show corrupted frames and the corresponding reconstructed frames. The black regions in Figure 35 (a) and Figure 36 (a) indicate losses of the video data. Figure 35 shows an EC method using spatial interpolation from the neighboring pixels. Figure 36 shows an EC method using temporal interpolation. That is, if some pixel values are lost, the decoder copies the pixel values from the previous frame at the corresponding locations to the current frame. The EC method using temporal interpolation can be extended to copying the pixel values from the previous frame at the motion-compensated locations. The motion vectors used for motion compensation either are assumed error-free or can be estimated at the decoder [3][31].

We use the simple temporal interpolation method in the study. Future extension includes using motion-compensated temporal interpolation, or more sophisticated EC methods as mentioned in [8][9].

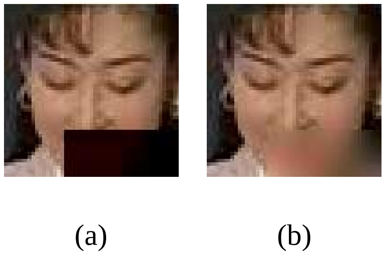


Figure 35. EC example by spatial interpolation: (a) the corrupted frame without EC, and (b) the reconstructed frame with EC

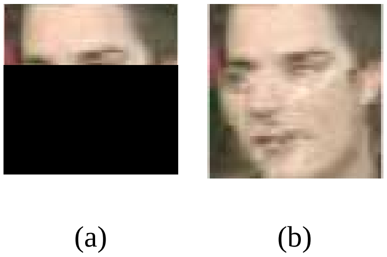


Figure 36. EC example by temporal interpolation: (a) the corrupted frame without EC, and (b) the reconstructed frame with EC

4.2.2. Error Concealment Aware Precoding

In addition to ECARS, the precoding process can be EC aware to prioritize the precoded video based on the *gain*. We present an example EC aware precoding process by means of macroblock

(MB) prioritization. A MB in a frame is ranked according to its gain, which depends on how well this MB can be reconstructed by the EC method used at the receiver. The gain of sending a MB is large if the EC method used at the receiver cannot reconstruct this MB very well.

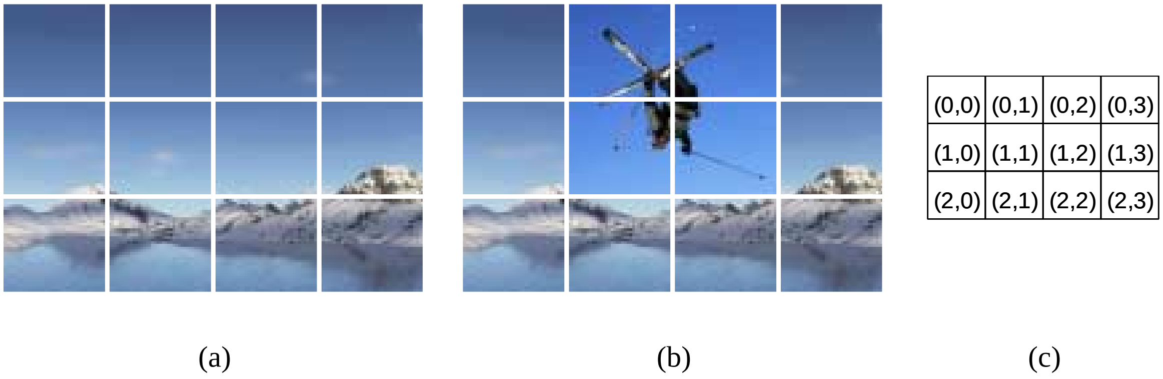
Let us consider that a simple temporal interpolation based EC method is adopted. Figure 37 provides us with an illustration of EC aware MB prioritization. If MB $(1,1)$ is lost in Frame n, it cannot be well reconstructed by MB $(1,1)$ from Frame $n-1$. On the other hand, if MB $(0,3)$ is lost in Frame n, it can be well reconstructed by MB $(0,3)$ from Frame $n-1$. Therefore, we should rank MB $(1,1)$ with higher priority than MB $(0,3)$.

We can use square sum of the pixel differences between the original MB and the EC-reconstructed MB as the measure for priority. The larger the square sum is, the larger the gain for this MB is, thus, the higher the priority of this MB is. Assuming that the neighboring MB of the MB considered are decoded without errors, the MB gain g_j is defined as follows:

$$g_j = \sum_{u=0}^{255} \left(c_{ju} - p_{ju} - s_{ju}\right)^2, \; j = 1 \sim \text{number of MB in a frame} \tag{4.1}$$

where u[3] is the coefficient index in a MB, c_{ju} is the coefficient of the EC-reconstructed MB, p_{ju} is the prediction value of this MB, and s_{ju} is the residue value of this MB. $p_{ju} + s_{ju}$ is the ideal value without any transmission error or rate adaptation by rate shaping. $c_{ju} - \left(p_{ju} + s_{ju}\right)$ is to see how far the EC value is from the ideal value. The assumption that the neighboring MB are decoded without errors is valid if the packet losses are not too bursty.

[3] We consider only the Y components in the MB without loss of generality. Thus, there are four 8×8 blocks or 256 coefficients inside.



(a) (b) (c)

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)

Figure 37. (a) Frame $n-1$, (b) Frame n, and (c) MB indices. EC aware MB prioritization— MB (1,1) has higher priority than MB (0, 3)

An observation to make is that the conventional video coding can be considered as a special case of the proposed EC aware MB prioritization. Let us consider the case where no motion vector is used in video coding. The MB with large residues is encoded and transmitted, while the MB with small residues does not need to be transmitted since the small residues will become zero after quantization. This case translates to the case of EC aware MB prioritization using temporal interpolation with zero motion vectors. Let us consider another case where motion vectors are included in video coding. This then translates to the case of EC aware MB prioritization using temporal interpolation with motion vectors. We can see that the proposed EC aware MB prioritization is more general since it is not limited to any specific error concealment method.

The source-coded bitstream with EC aware MB prioritization can be appended with parity bits from the FEC coding. First, the bits of the highest priority MB is placed followed by the bits of the second highest priority MB and so on, as shown in Figure 38 (a). To label the MB after the MB are ordered by their priorities, 446 bytes of complementary information of the MB labels are needed if the video is in common intermediate format (CIF). The bits are then divided into sublayers as shown in Figure 38 (b). Sublayer $i+1$ has more bits than Sublayer i since we want to achieve UPP for the sublayers when appended with the parity bits. For example, we can let Sublayer 1 consists of bits from the first 10 highest priority MB, Sublayer 2 consists of bits from the following 20 highest priority MB, and so on. Each sublayer is then appended with parity bits from the FEC coding as shown in Figure 38 (c).

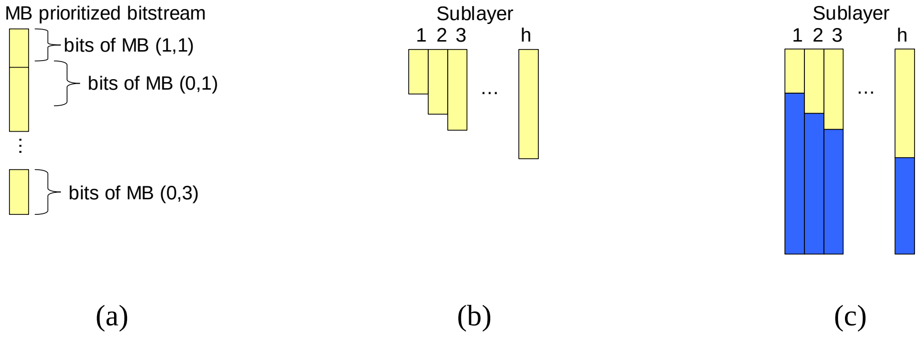


Figure 38. Precoded video: (a) MB prioritized bitstream, (b) MB prioritized bitstream in sublayers, and (c) FEC coded MB prioritized bitstream

Also, with the MB gain defined, we can define the sublayer gain correspondingly as:

$$G_i = \sum_{j \in \{\text{indices of MB that belong to Sublayer } i\}} g_j \quad , \; i = 1 \sim \text{number of sublayers in a frame} \qquad (4.2)$$

Note again that ECARS can perform rate adaptation with or without EC aware precoding as long as the precoded video is provided with sublayer gains. The sublayer gain will be used later in the R-D optimized ECARS.

4.2.3. Timely Feedback

In the system of ECARS with feedback, feedback information from the receiver is utilized. Feedback can be carried out by RTCP message [44]. To understand whether feedback information is timely, i.e. within one frame interval[4], let us examine the one-way transmission time [25].

[4] The ITU-T recommends the following limits for one-way transmission time according to ITU-T Recommendation G. 131 [26]. However, the limit in the rate shaping system with feedback is a frame interval since the rate shaping decision is made on each frame interval.

- 0 to 150 ms: Acceptable for most user applications.
- 150 to 400 ms: Acceptable provided that administrations are aware of the transmission time impact on the transmission quality of user applications.
- above 400 ms: Unacceptable for general network planning purposes; however, it is recognized that in some exceptional cases this limit will be exceeded.

The transmission time is the aggregate of several components, e.g. group delay in cables and equipment processing times, etc. In addition, one-way transmission time of the national extension circuits and the international circuits must be taken into account.

The transmission time for the national extension circuits can be estimated as follows:

a) In purely analogue networks, the transmission time will probably not exceed:

$$12 + (0.004 \times \text{distance in kilometers})\ \text{ms} \tag{4.3}$$

Here the factor 0.004 is based on the assumption that national trunk circuits will be routed over high-velocity plant (250 km/ms). The 12 ms constant term makes allowance for terminal equipment and for the probable presence in the national network of a certain quantity of loaded cables (e.g. three pairs of channel translating equipments plus about 160 km of H 88/36 loaded cables). For an average size country, the one-way propagation time will be less than 18 ms.

b) In mixed analogue/digital networks, the transmission time can generally be estimated by the equation given for purely analogue networks. However, under certain unfavorable conditions, increased delay may occur compared with the purely analogue case. This occurs in particular when digital exchanges are connected with analogue transmission systems through PCM/FDM equipment in tandem, or trans-multiplexers. With the growing degree of digitalization, the transmission time will gradually approach the condition of purely digital networks.

c) In purely digital networks between local exchanges, based on optical fiber systems (e.g. an IDN), the transmission time will probably not exceed:

$$3 + (0.005 \times \text{distance in kilometers})\ \text{ms} \tag{4.4}$$

The 3 ms constant term makes allowance for one pair of PCM coder and decoder and for five digitally switched exchanges. The value 0.005 is a mean value for optical fiber systems; for coaxial cable systems and radio-relay systems 0.004 is to be used.

d) In purely digital networks between subscribers (e.g. an ISDN), the delay of c) above has to be increased by up to 3.6 ms if burst-mode (time compression multiplexing) transmission is used on 2-W local subscriber lines.

The transmission time for the international circuits can use values of Table 1 below.

Table 1. One-way transmission time

Transmission or processing system	Contribution to one-way transmission time	Remarks
Terrestrial coaxial cable or radio-relay system: FDM and digital transmission	4 µs/km	
Optical fiber cable system, digital transmission	5 µs/km (Note 1)	Allows for delay in repeaters and regenerators
Submarine coaxial cable system	6 µs/km	
Submarine optical fiber system: – transmit terminal – receive terminal	13 ms 10 ms	Worst case
Satellite system: – 400 km altitude – 14 000 km altitude – 36 000 km altitude	12 ms 110 ms 260 ms	Propagation through space only (between earth stations)
FDM channel modulator or demodulator	0.75 ms (Note 2)	
PLMS (Public Land Mobile System) – objective 40 ms	80-110 ms	
H.260-series video coders and decoders	Further study (Note 3)	
DCME per pair: for speech, VBD, and non-remodulated fax	30 ms	Half the sum of transmission times in both directions of transmission
DCME per pair: for speech, VBD, and non-remodulated fax	30 ms	
DCME in conjunction with ITU-T Rec. G.763 or ITU-T Rec. G.767) per pair: for remodulated fax	200 ms	
PCME per pair: – with speech and non-remodulated VBD – with remodulated VBD	35 ms 70 ms	
Transmultiplexer	1.5 ms (Note 4)	
Digital transit exchange, digital-digital	0.45 ms (Note 5)	
Digital local exchange, analogue-analogue	1.5 ms (Note 5)	
Digital local exchange, analogue subscriber line-digital junction	0.975 ms (Note 5)	
Digital local exchange, digital subscriber line-digital junction	0.825 ms (Note 5)	
Echo cancellers	0.5 ms (Note 6)	
ATM (CBR using AAL1)	6.0 ms (Note 7)	

NOTE 1 – This value is provisional and is under study.

NOTE 2 – These values allow for group-delay distortion around frequencies of peak speech energy and for delay of intermediate higher order multiplex and through-connecting equipment.

NOTE 3 – Further study required. Delay for these devices is usually non-constant, and the range varies by implementation. Current implementations are on the order of several hundred milliseconds and considerable delay is added to audio channels to achieve lip-synchronization. Manufacturers are encouraged to reduce their contribution to transmission time, in accordance with this ITU-T Recommendation.

NOTE 4 – For satellite digital communications where the transmultiplexer is located at the earth station, this value may be increased to 3.3 ms.

NOTE 5 – These are mean values: depending on traffic loading, higher values can be encountered, e.g. 0.75 ms (1.950 ms, 1.350 ms, or 1.250 ms) with 0.95 probability of not exceeding.

NOTE 6 – This is averaged for both directions of transmission.

NOTE 7 – This is the cell formation delay of 64 kbits/s stream when it completely fills the cell (one voice channel per VC). In practical applications, additional delay will result, e.g. from cell loss detection and buffering. Other delays may be applicable to other AALs and cell mapping arrangements, and are for further study.

Since rate shaping is performed at each link, the transmission time from one hop to the other is considered. We can see from (4.3), (4.4), and Table 1 that the feedback consumes in general less than 33 ms (assuming the video frame rate is 30 frames/sec) to get back to the sender.

4.3. Algorithms for ECARS

To explain the algorithms for ECARS, let us start from a simple example as an extension to BRS. Let us consider that the precoded video consists of two layers of video bitstream, namely, the base layer and the enhancement layer. Each layer is protected by some parity bits from the FEC coding. The setting is shown earlier in Figure 8 (a). The rate shaper is extended to give a finer decision on how many symbols to send (or how many symbols to drop) for each layer, instead of deciding which segment(s) to drop. Since the rate shaper is aware of the EC method used at the receiver, it can evaluate how much distortion it will result in if the rate shaper decides to send a certain amount of symbols for each layer. In other words, the rate shaper can evaluate how much gain it will get if it decides to send this certain amount of symbols for each layer. In general, the base layer can be reconstructed well with EC since the base layer consists of coarse information of the video that can be easily reconstructed. On the other hand, the enhancement layer, which consists of fine details of the video, cannot be easily reconstructed. The EC aware rate shaper may assign a higher gain on sending symbols in the enhancement layer than the symbols in the base layer. Notice that the example given here is just for understanding. We are not going to prioritize the bitstream in terms of base and enhancement layers. Instead, all the discussions hereafter will occur in the base layer as said in the very beginning of the chapter. The prioritization takes place when we order the macroblocks (MB) by their MB gains.

4.3.1. ECARS without Feedback

Suppose ECARS is given the precoded video with *sublayers*. Each sublayer consists of symbols from source coding, which is shown as the upper portion of each stripe in Figure 39 (a), and symbols from channel coding, which is shown as the lower portion of each stripe in Figure 39 (a). The darken bars in Figure 39 (b) represent the symbols to be sent by ECARS.

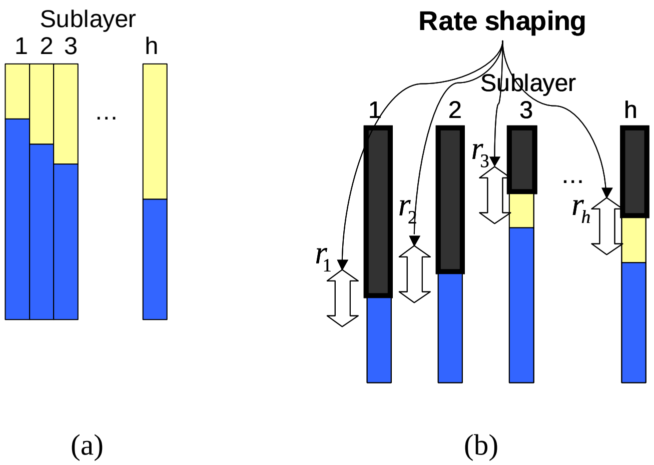


Figure 39. (a) Precoded video in sublayers and (b) ECARS decision on which symbols to send

The problem formulation for ECARS is as follows. The total *gain* is increased (or the total distortion is decreased) as more sublayers are correctly decoded. With Sublayer 1 correctly decoded, the total gain is increased by G_1 (accumulated gain is G_1); with Sublayer 2 correctly decoded, the total gain is increased further by G_2 (accumulated gain is $G_1 + G_2$); and so on. Note that G_i of Sublayer i can either (1) be calculated given the bitstream and the EC method used by the receiver, after performing partial decoding in order to get the values of gain; or (2) be embedded in the bitstream as the "meta-data". ECARS is EC aware because the gain G_i is dependent on the EC method used by the receiver. G_i of Sublayer i is different for every frame. The expected accumulated gain is then:

$$G = \sum_{i=1}^{h} G_i v_i \tag{4.5}$$

where h is the number of sublayers of this frame, and v_i is the recovery rate of Sublayer i. The definition of v_i and the rest of the R-D optimization follow what is stated and proposed in Section 3.2 except the change of definition in the expected accumulated gain shown above in (4.5).

4.3.2. ECARS with Feedback

As we have discussed, the goal of rate shaping is to achieve the maximal expected accumulated gain. We should consider the frame dependency in the process of R-D optimization, since the reconstructed result of the previous frame will affect the following frames if the video is predictive coded, and/or the EC method performed at the receiver utilizes the temporal information. We propose to use feedback information from the receiver to carry information about the previous reconstructed frame for the use of the current frame in rate shaping.

Notice that in our setup, the forward and feedback transmissions share the same amount of bandwidth. In addition, the use of feedback in ECARS does not suggest an increase in channel capacity. The total bandwidth stays the same. Feedback is to inform the rate shaper for transmission of "more useful" data (with a larger gain) rather than "less useful" (with a smaller gain).

If the EC method used at the receiver is precisely known by the rate shaper, with the information where the macroblock is corrupted being sent back, the rate shaper can imitate what the decoder gets. Knowing what the decoder gets, the rate shaper can calculate the MB gain for R-D optimization. In the later experiments, we will use "ecars-ideal" to represent that the location of the corrupted macroblock is sent back and the EC method used at the receiver is known.

If the EC method used at the receiver is not precisely known by the rate shaper, we can try to approximate what the gain should be by sending back information as (i) the location of the corrupted macroblock; or (ii) the mean of the corrupted macroblock in addition to the location of the corrupted macroblock. In the later experiments, we will use "ecars-nf" to represent no feedback is used and the gain information is embedded in the bitstream, "ecars-loc" to represent feedback with the location of the corrupted macroblock is used, and "ecars-mean" to represent feedback with the location and mean of the corrupted macroblock is used. Note that none of the rate shaping methods, "ecars-nf", "ecars-loc", "ecars-mean" know the precise EC method at the sender.

For "ecars-loc", we will explain how to determine the gain g'_i of each MB with respect to three cases where frame dependency can occur. The three cases are denoted as (0,1), (1,2), and (1,1). An "1" in the first field represents Inter-coding, and a "0" in the first field represents Intra-coding. An "1" in the second field represents an EC method that utilizes the temporal information, and "2" in the second field represents an EC method that utilizes only the spatial information from the neighbors. Likewise, we will explain how to determine the gain with respect to (0,1), (1,2), and (1,1) for "ecars-mean".

In general, the MB gain g'_j of each MB remains the same as g_j if the corresponding MB of the previous frame is successfully decoded. We then see how to determine the MB gain g'_j of each MB if the corresponding MB of the previous frame is corrupted.

4.3.2.1. ECARS with Feedback: ECARS-LOC

- (0,1): If a MB of the previous frame is corrupted, we want to increase the MB gain of the corresponding MB of the current frame. Temporal EC will use information from the corrupted MB of the previous frame if the corresponding MB of the current frame cannot be decoded successfully. Thus, we want to make sure that the MB of the current frame is sent with good protection. A natural way is to double the value of MB gain g_j as:

$$g'_j = 2g_j\,,\ j = 1 \sim \text{number of MB in a frame} \tag{4.6}$$

- (1,2) and (1,1): If a MB of the previous frame is corrupted, we want to decrease the MB gain of the corresponding MB of the current frame. We do not want to use the corrupted MB of the previous frame for predictive coding. Sending the residues of the MB of the current frame is useless if the prediction to this MB is already erroneous. A natural way is to set the MB gain to zero as:

$$g'_j = 0\,,\ j = 1 \sim \text{number of MB in a frame} \tag{4.7}$$

Having determined the MB gains, they are grouped together to form the sublayer gain G_i, where $i = 1 \sim \text{number of sublayers in a frame}$. Again, the rest of the R-D optimization follows what is stated and proposed in Section 3.2 except the change of definition in the expected accumulated gain.

4.3.2.2. ECARS with Feedback: ECARS-MEAN

The method proposed in Section 4.3.2.1 is heuristic. We know that the MB gain g'_j remains the same as g_j if the corresponding MB of the previous frame is successfully decoded. If the corresponding MB of the previous frame is corrupted, we can determine the MB gain g'_j from g_j by examining "how different" the corrupted corresponding MB of the previous frame is from the successfully decoded counterpart.

The further apart the corrupted MB is from its successfully decoded counterpart, the more we should change the gain of the MB of the current frame. The corrupted MB of the previous frame will affect either the prediction of the corresponding MB of the current frame, the EC reconstruction of the corresponding MB of the current frame, or both. We propose to send the mean of the prediction of the MB for the current frame p'_j and the mean of the EC reconstruction of the MB for the current frame c'_j, both affected by the corrupted MB of the previous frame, back to the sender for calculation of g'_j.

Recall that the original definition of the MB gain g_j of the current frame is $g_j = \sum_{u=0}^{255} \left(c_{ju} - p_{ju} - s_{ju}\right)^2$ from (4.1). Now the MB of the previous frame is corrupted, the MB gain g'_j of the current frame should be:

$$\begin{aligned} g'_j &= \sum_{u=0}^{255} \left[\left(c'_{ju} - p_{ju} - s_{ju}\right)^2 - \left(p'_{ju} + s_{ju} - \left(p_{ju} + s_{ju}\right)\right)^2\right] \\ &= \sum_{u=0}^{255} \left[\left(c'_{ju} - p_{ju} - s_{ju}\right)^2 - \left(p'_{ju} - p_{ju}\right)^2\right] \end{aligned}, \qquad (4.8)$$

$$j = 1 \sim \text{number of MB in a frame}$$

since the gain is defined as the distortion decrease comparing sending the MB of the current frame with not sending but reconstructing the MB by EC. With the EC method known and the location where the MB is corrupted, the rate shaper can calculate the exact values of p'_{ju} and c'_{ju}, where $u = 1 \sim 256$. Those values p'_{ju} and c'_{ju} are to be used by (4.8). However, if the EC method is not precisely known at the rate shaper (which is usually the case since the receiver might not want to disclose its own EC method), we can instead consider only the means p'_j and c'_j back to the sender to approximate g'_j as follows:

$$g'_j = \frac{\left(c'_j - p_j - s_j\right)^2 - \left(p'_j - p_j\right)^2}{\left(c_j - p_j - s_j\right)^2} g_j , \qquad (4.9)$$

$$j = 1 \sim \text{number of MB in a frame}$$

Without subscript u, the above values in the numerator and the denominator represent means. Similarly, having determined the MB gains, they are grouped together to form the sublayer gain G_i, where $i = 1 \sim \text{number of sublayers in a frame}$. The rest of the R-D optimization follows what is stated and proposed in Section 3.2 except the change of definition in the expected accumulated gain.

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 then:

$$\text{maximize} \qquad G = \sum_{m=1}^{F} \left[\sum_{i=1}^{h} \left(G_{mi} v_{mi} \right) \right] \qquad (4.10)$$

$$\text{subject to} \qquad \sum_{m}^{F} \sum_{i=1}^{h} r_{mi} \le C \qquad (4.11)$$

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

Note again the packetization is performed after error-resilient rate shaping. That is, symbols are grouped into packets after the decision of $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \cdots & r_h \end{bmatrix}$ has been made. Small packet is desirable to make use of the fine-grained decision resulted from ECARS. For example, a big packet that contains all the symbols from a frame could be unrecoverable if it is decided to be dropped by the lower layers (for example, the link layer detects a CRC check error for this big packet).

4.4. Experiment

We will show in this section the effectiveness of ECARS in streaming the precoded video over packet-loss networks. Seven methods will be compared side-by-side: random dropping (with legend “rand”), UPPRS1 (with legend “upprs1”), UPPRS2 (with legend “upprs2”), and non-ECARS (with legend “n-ecars”), ECARS without feedback (with legend “ecars-nf”), ECARS with location feedback (with legend “ecars-loc”), and ECARS with mean and location feedback (with legend “ecars-mean”). One ideal method (as the performance bound) where EC method at the receiver is precisely known will be shown as well.

The test video sequences are “akiyo”, “foreman”, and “stefan” in CIF format. Sequence “akiyo” represents a video sequence with lower bit rate due to simpler texture and less motion. Sequence “foreman” represents a video sequence with medium bit rate with regular texture and motion. Sequence “stefan” represents a video sequence with higher bit rate with complex texture and faster motion. The frame rate is 30 frames/sec.

The bandwidth of the simulated networks fluctuates between 2 Mbits/sec and 11 Mbits/sec. The bandwidth of the forward channel is subtracted by the amount of bits the feedback channel requires if there is some feedback sent back from the receiver. The bit error rate (BER) of the channel also fluctuates according to the two-state Markov chain model detailed in Appendix B. The wireless channel simulation parameters can be found in B.2. Under the same network condition (the same BER trace and the same bandwidth trace), the results shown in the following are tested for 10 different seeds for pseudo-random simulations. That is, the “overall PSNR” result shown is the average of 10 different tests. The frame-by-frame PSNR result is an instance of the 10 tests.

Given the *gain* embedded in the bitstream, ECARS consumes on the average <1% (the denominator is the bit rates of the source-coded bitstream) of the original precoded video to carry the gain information (“meta-data”). The performance improvement of ECARS in PSNR over non- rate shaping based methods is on the average 8 dB.

In the following, we will present the results of:

- Rate shaping vs. non- rate shaping (i.e., rate shaping vs. UPPRS)
- ECARS vs. Non-ECARS
- ECARS with feedback vs. ECARS without feedback
- ECARS with EC method known vs. ECARS without EC method known
- All seven methods

The reference that the results of all methods compared to in computing the PSNR, is the result of a video bitstream that is transmitted with no packet loss and with unlimited bandwidth.

4.4.1. Rate Shaping vs. UPPRS

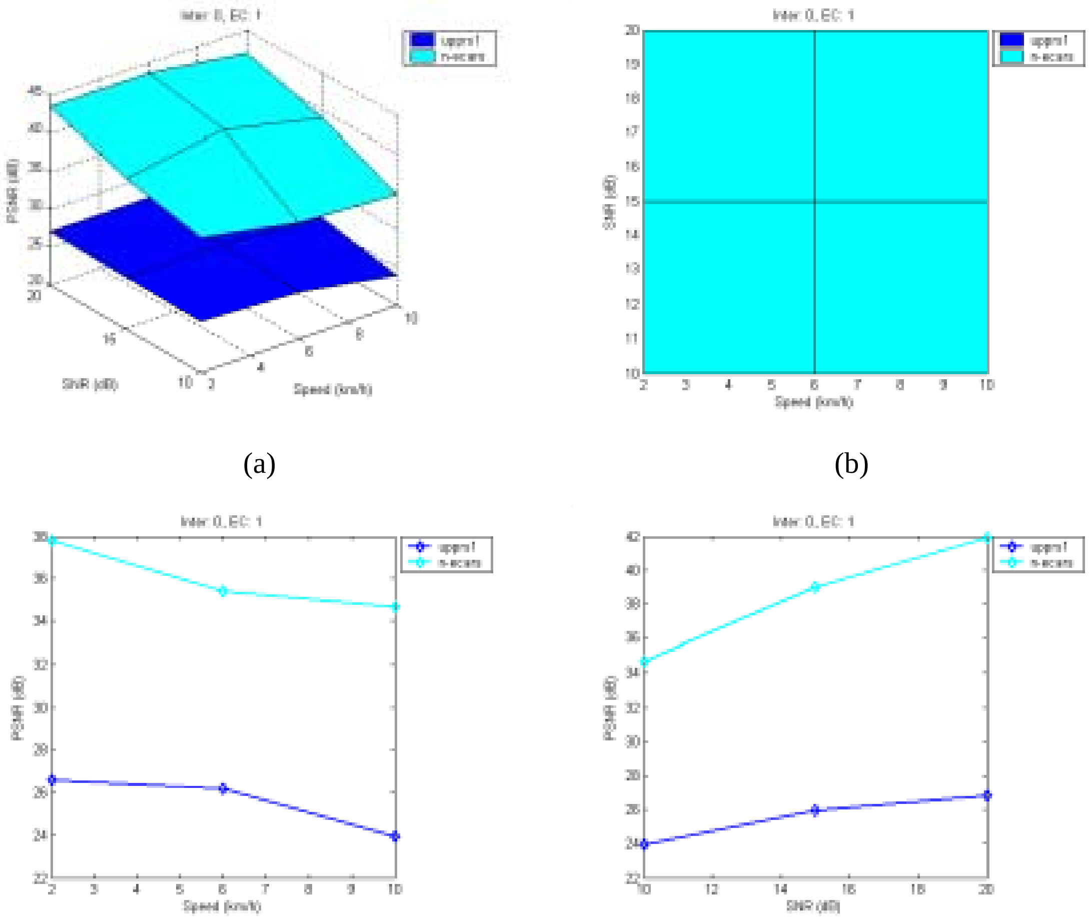
We will compare the results of rate shaping based method “n-ecars” with the non- rate shaping based method “upprs1” here. Results of Case (0, 1) are shown in Figure 40 and Figure 41; results of Case (1, 2) are shown in Figure 42 and Figure 43; and results of Case (0, 1) are shown in Figure 44 and Figure 45.

The performance in terms of the overall PSNR at various wireless channel conditions is shown in Figure 40, Figure 42, and Figure 44. Figure 40 (a), Figure 42 (a), and Figure 44 (a) show the 3-D plots of the overall PSNR. Figure 40 (b), Figure 42 (b), and Figure 44 (b) show the top views (seen from the top of the z-axis) of the 3-D plots. The color shown in the top view represents the color of the method that outperforms the others. At all wireless channel conditions, “n-ecars” outperforms “upprs1”. Even though “n-ecars” is not EC aware, it still performs rate shaping by R-D optimization.

Figure 40 (c), Figure 42 (c), and Figure 44 (c) show 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. The higher the transition probability is, the higher the packet-loss rate is, given the same BER. On the other hand, the EC performance degrades as the error becomes more bursty because EC relies on spatial or temporal neighbors. Neighbors are usually corrupted if the error is bursty. Therefore, from the results, we do not see the correlation between the overall PSNR and the speed.

Figure 40 (d), Figure 42 (d), and Figure 44 (d) show 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. Also from Section 2.2, we know that the smaller the BER is, the smaller the packet-loss rate is, given the same burstiness. Smaller packet-loss rate then leads to a higher PSNR.

Frame-by-frame PSNR performance is shown in Figure 41, Figure 43, and Figure 45. We also see that “n-ecars” outperforms “upprs1”.



(a)

(b)

(c)

(d)

Figure 40. Performance of Methods "upprs1" and "n-ecars" at various wireless channel conditions with Case (0, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

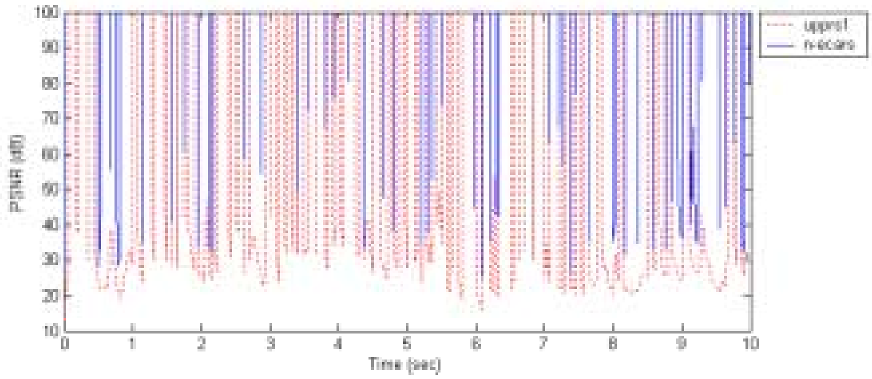
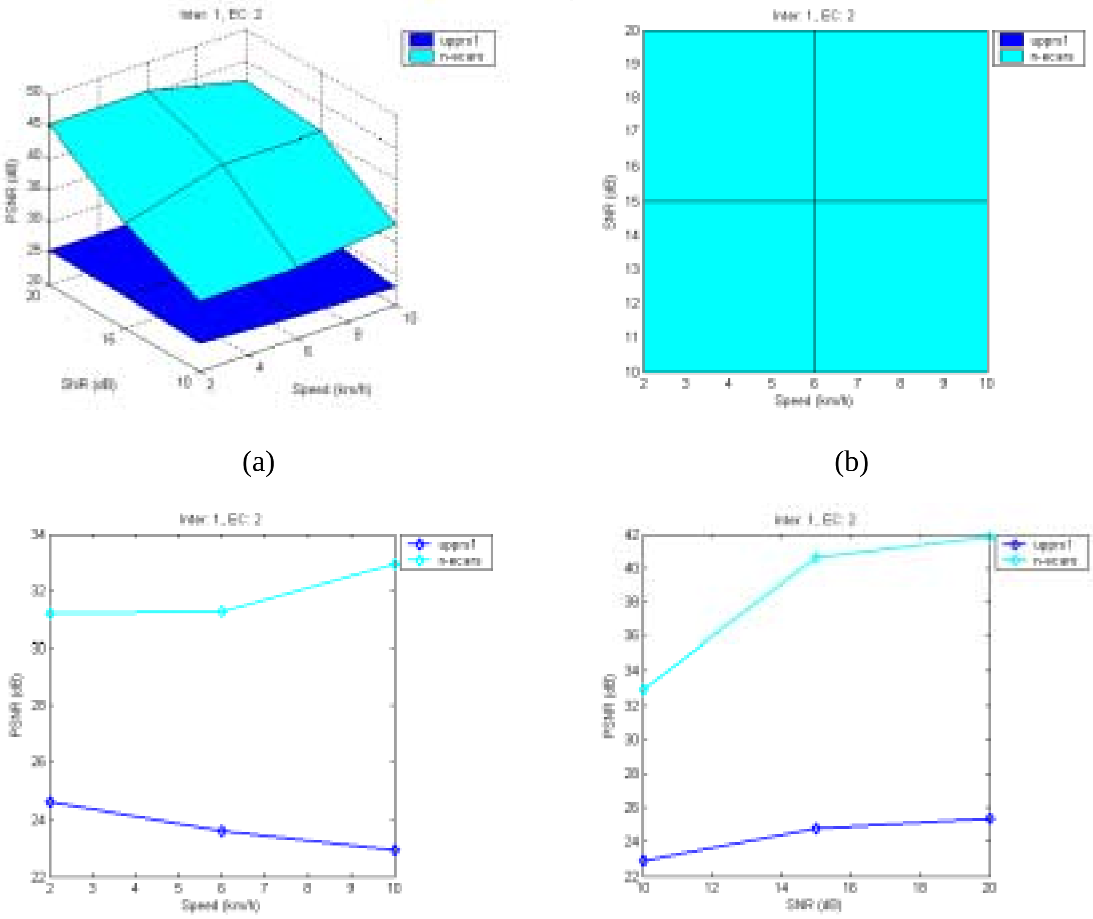


Figure 41. Frame-by-frame PSNR of Methods "upprs1" and "n-ecars" at 10 km/h and SNR=20 dB with Case (0, 1) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 42. Performance of Methods "upprs1" and "n-ecars" at various wireless channel conditions with Case (1, 2) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

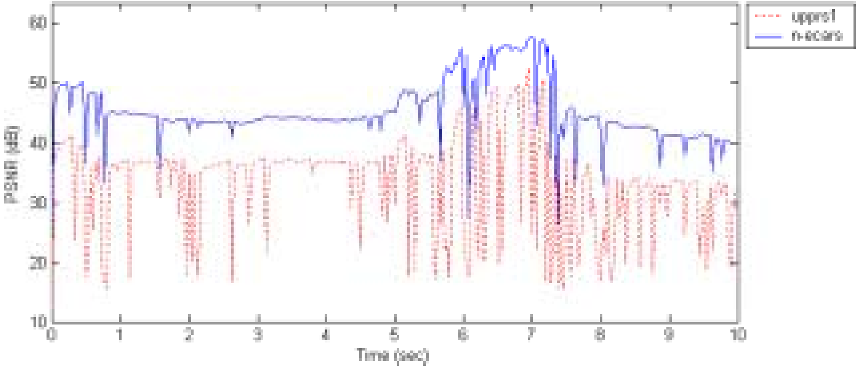
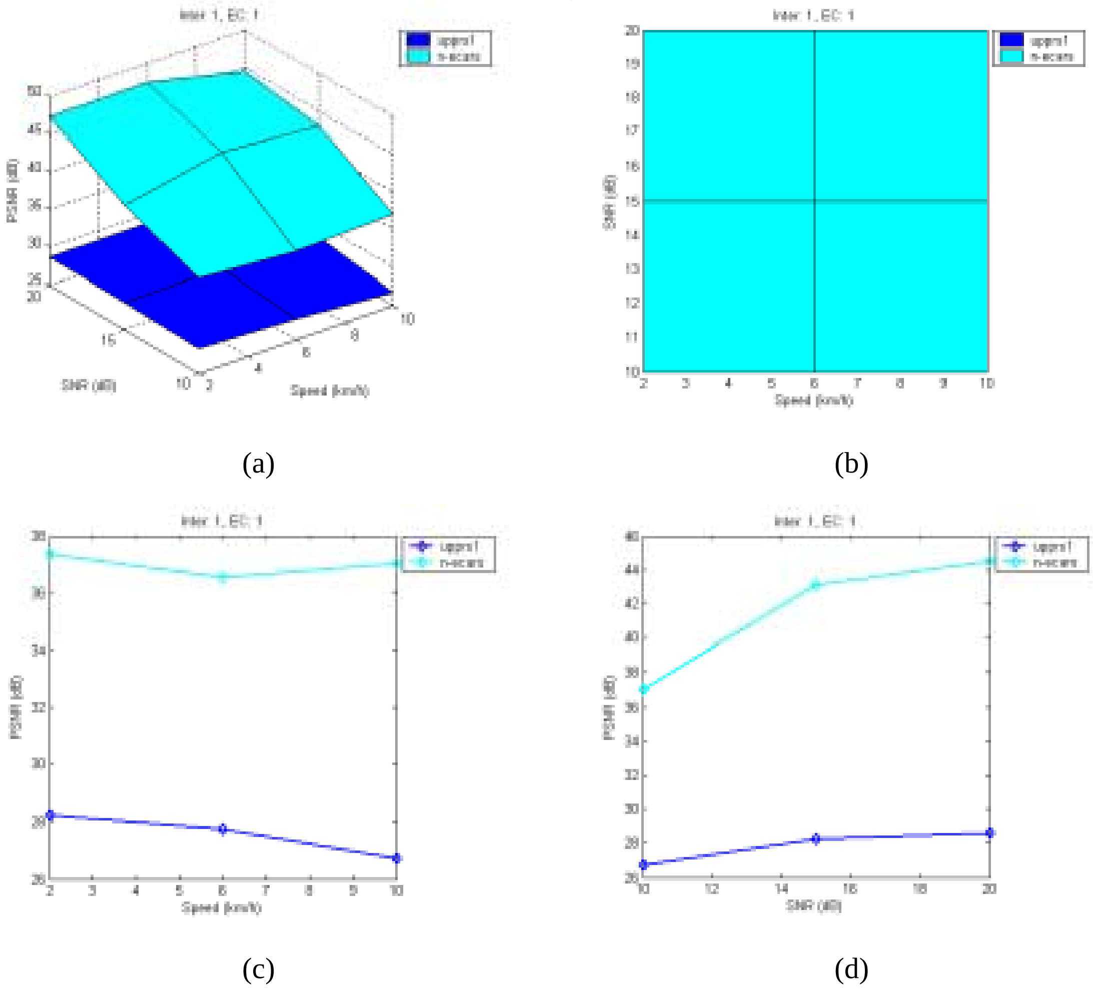


Figure 43. Frame-by-frame PSNR of Methods "upprs1" and "n-ecars" at 10 km/h and SNR=20 dB with Case (1, 2) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 44. Performance of Methods "upprs1" and "n-ecars" at various wireless channel conditions with Case (1, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

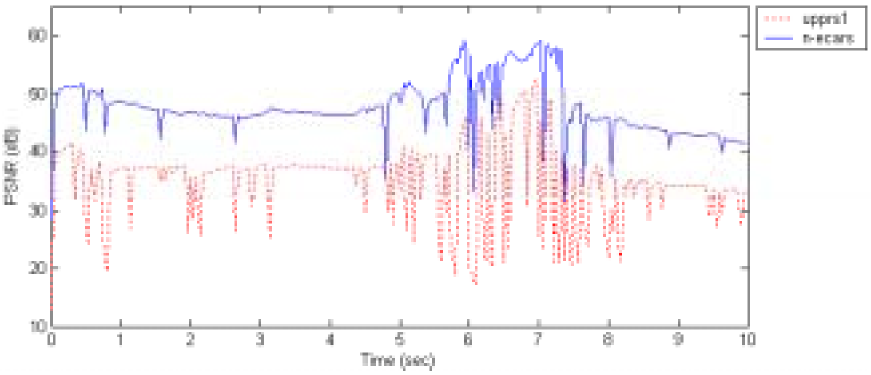


Figure 45. Frame-by-frame PSNR of Methods "upprs1" and "n-ecars" at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence "foreman"

4.4.2. ECARS vs. Non-ECARS

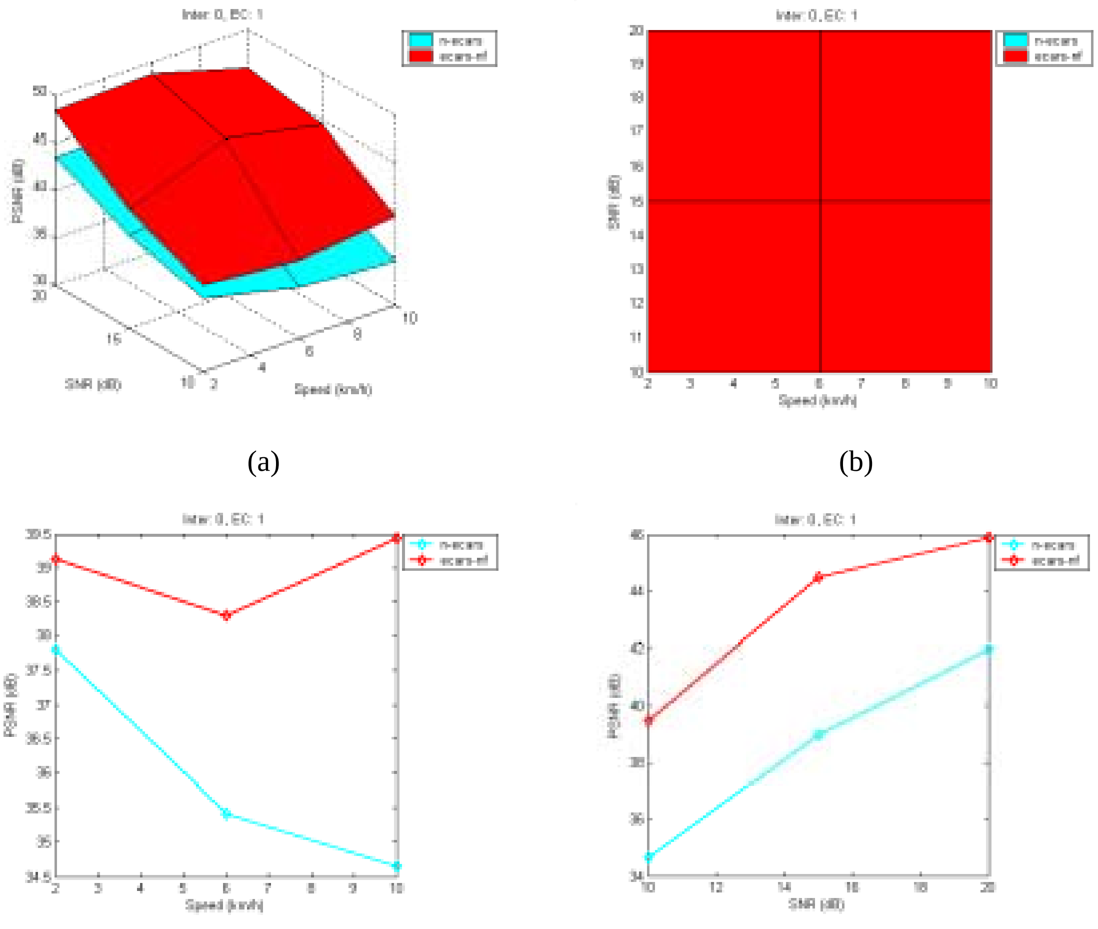
We will compare the results of EC aware rate shaping “ecars-nf” with the non- EC aware rate shaping “n-ecars” here. Results of Case (0, 1) are shown in Figure 46 and Figure 47; results of Case (1, 2) are shown in Figure 48 and Figure 49; and results of Case (0, 1) are shown in Figure 50 and Figure 51.

The performance in terms of the overall PSNR at various wireless channel conditions is shown in Figure 46, Figure 48, and Figure 50. Figure 46 (a), Figure 48 (a), and Figure 50 (a) show the 3-D plots of the overall PSNR. Figure 46 (b), Figure 48 (b), and Figure 50 (b) show the top views (seen from the top of the z-axis) of the 3-D plots. The color shown in the top view represents the color of the method that outperforms the others. At all wireless channel conditions, “ecars-nf” outperforms “n-ecars”.

Figure 46 (c), Figure 48 (c), and Figure 50 (c) show 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. The higher the transition probability is, the higher the packet-loss rate is, given the same BER. On the other hand, the EC performance degrades as the error becomes more bursty because EC relies on spatial or temporal neighbors. Neighbors are usually corrupted if the error is bursty. Therefore, from the results, we do not see the correlation between the overall PSNR and the speed.

Figure 46 (d), Figure 48 (d), and Figure 50 (d) show 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. Also from Section 2.2, we know that the smaller the BER is, the smaller the packet-loss rate is, given the same burstiness. Smaller packet-loss rate then leads to a higher PSNR.

Frame-by-frame PSNR performance is shown in Figure 47, Figure 49, and Figure 51. We also see that “ecars-nf” outperforms “n-ecars”.



(a) (b)

(c) (d)

Figure 46. Performance of Methods “n-ecars” and “ecars-nf” at various wireless channel conditions with Case (0, 1) for Sequence “foreman”: (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

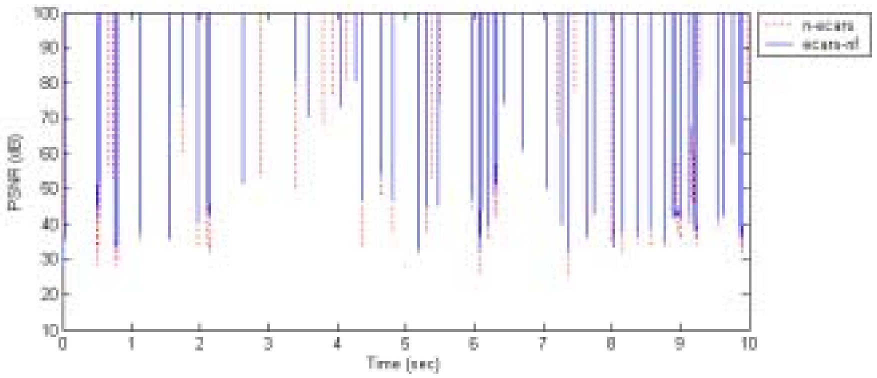
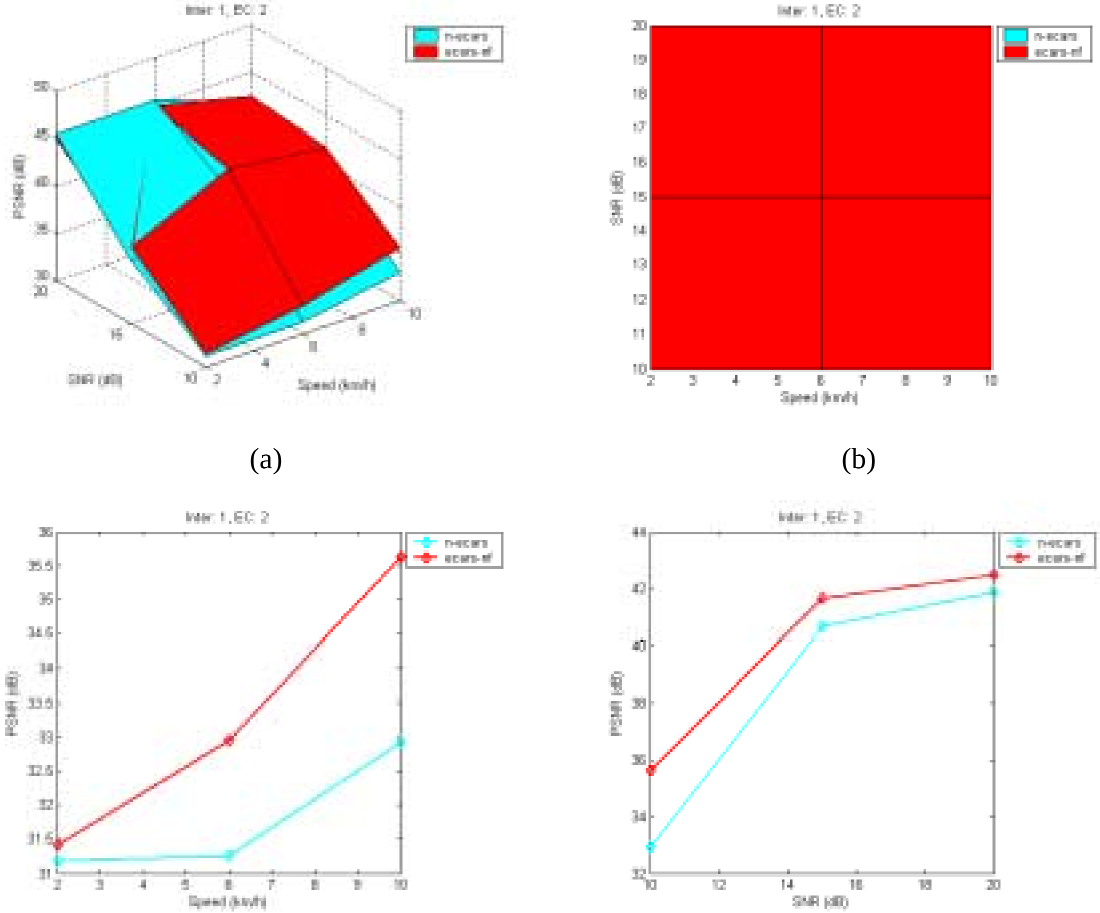


Figure 47. Frame-by-frame PSNR of Methods “n-ecars” and “ecars-nf” at 10 km/h and SNR=20 dB with Case (0, 1) for Sequence “foreman”



(a)

(b)

(c)

(d)

Figure 48. Performance of Methods "n-ecars" and "ecars-nf" at various wireless channel conditions with Case (1, 2) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

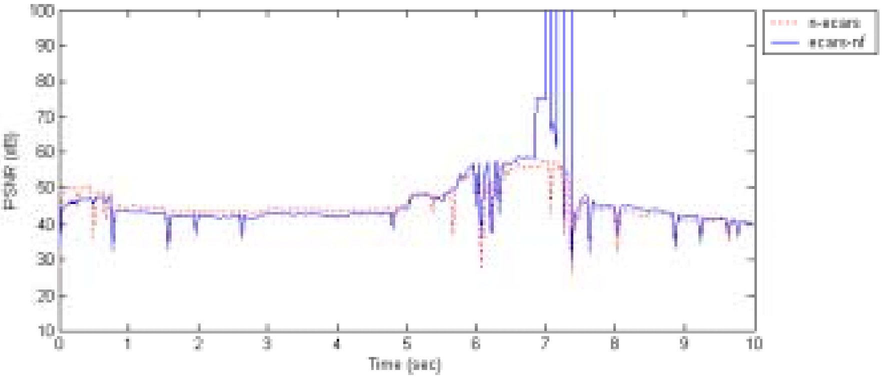
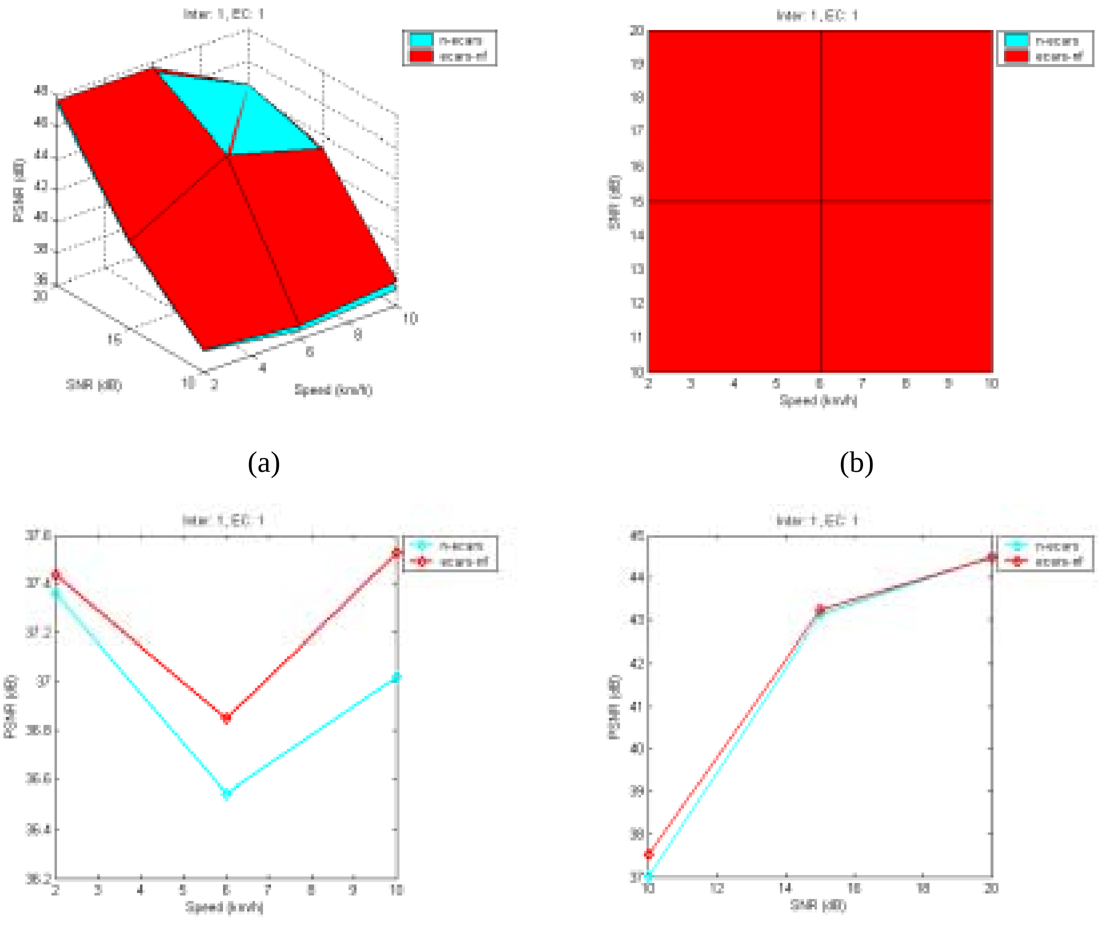


Figure 49. Frame-by-frame PSNR of Methods "n-ecars" and "ecars-nf" at 10 km/h and SNR=20 dB with Case (1, 2) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 50. Performance of Methods "n-ecars" and "ecars-nf" at various wireless channel conditions with Case (1, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

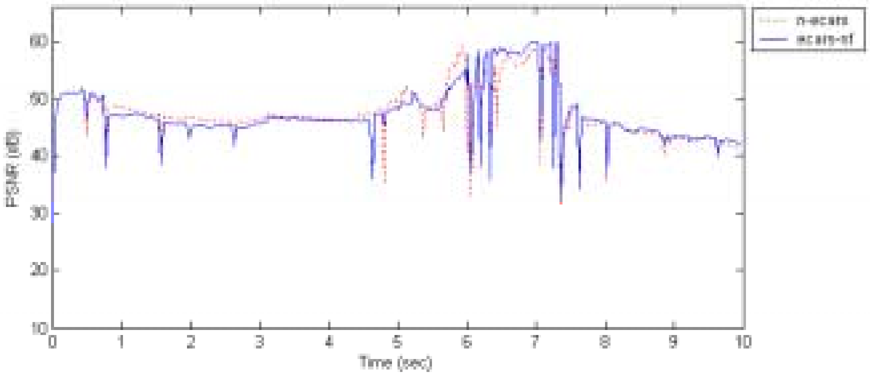


Figure 51. Frame-by-frame PSNR of Methods "n-ecars" and "ecars-nf" at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence "foreman"

4.4.3. ECARS with Feedback vs. ECARS without Feedback

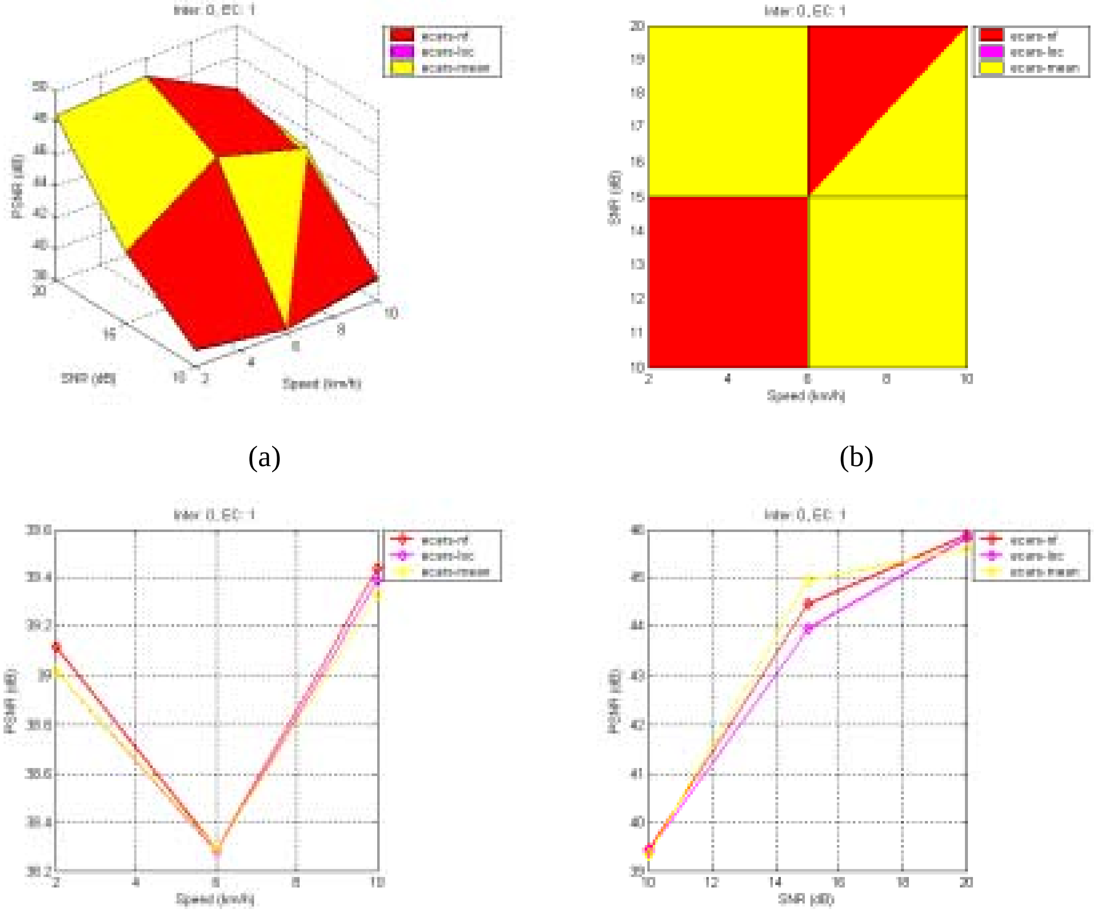
We will compare the results of EC aware rate shaping with feedback “ecars-loc” and “ecars-mean” with the EC aware rate shaping without feedback “ecars-nf” here. Results of Case (0, 1) are shown in Figure 52 and Figure 53; results of Case (1, 2) are shown in Figure 54 and Figure 55; and results of Case (0, 1) are shown in Figure 56 and Figure 57.

The performance in terms of the overall PSNR at various wireless channel conditions is shown in Figure 52, Figure 54, and Figure 56. Figure 52 (a), Figure 54 (a), and Figure 56 (a) show the 3-D plots of the overall PSNR. Figure 52 (b), Figure 54 (b), and Figure 56 (b) show the top views (seen from the top of the z-axis) of the 3-D plots. The color shown in the top view represents the color of the method that outperforms the others. “ecars-mean” outperforms “ecars-loc” and “ecars-nf” at most of the channel conditions with small margins.

Figure 52 (c), Figure 54 (c), and Figure 56 (c) show 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. The higher the transition probability is, the higher the packet-loss rate is, given the same BER. On the other hand, the EC performance degrades as the error becomes more bursty because EC relies on spatial or temporal neighbors. Neighbors are usually corrupted if the error is bursty. Therefore, from the results, we do not see the correlation between the overall PSNR and the speed.

Figure 52 (d), Figure 54 (d), and Figure 56 (d) show 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. Also from Section 2.2, we know that the smaller the BER is, the smaller the packet-loss rate is, given the same burstiness. Smaller packet-loss rate then leads to a higher PSNR.

Frame-by-frame PSNR performance is shown in Figure 53, Figure 55, and Figure 57. We also see that “ecars-mean” outperforms “ecars-loc” and “ecars-nf” at most of the channel conditions with small margins.



(a)

(b)

(c)

(d)

Figure 52. Performance of Methods "ecars-nf", "ecars-loc", and "ecars-mean" at various wireless channel conditions with Case (0, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

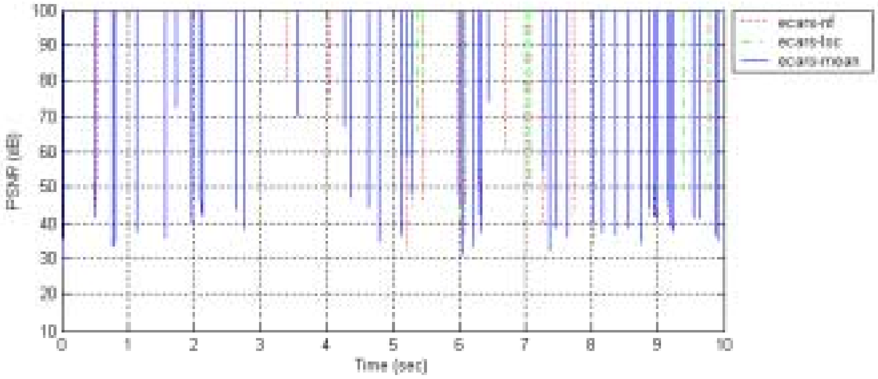
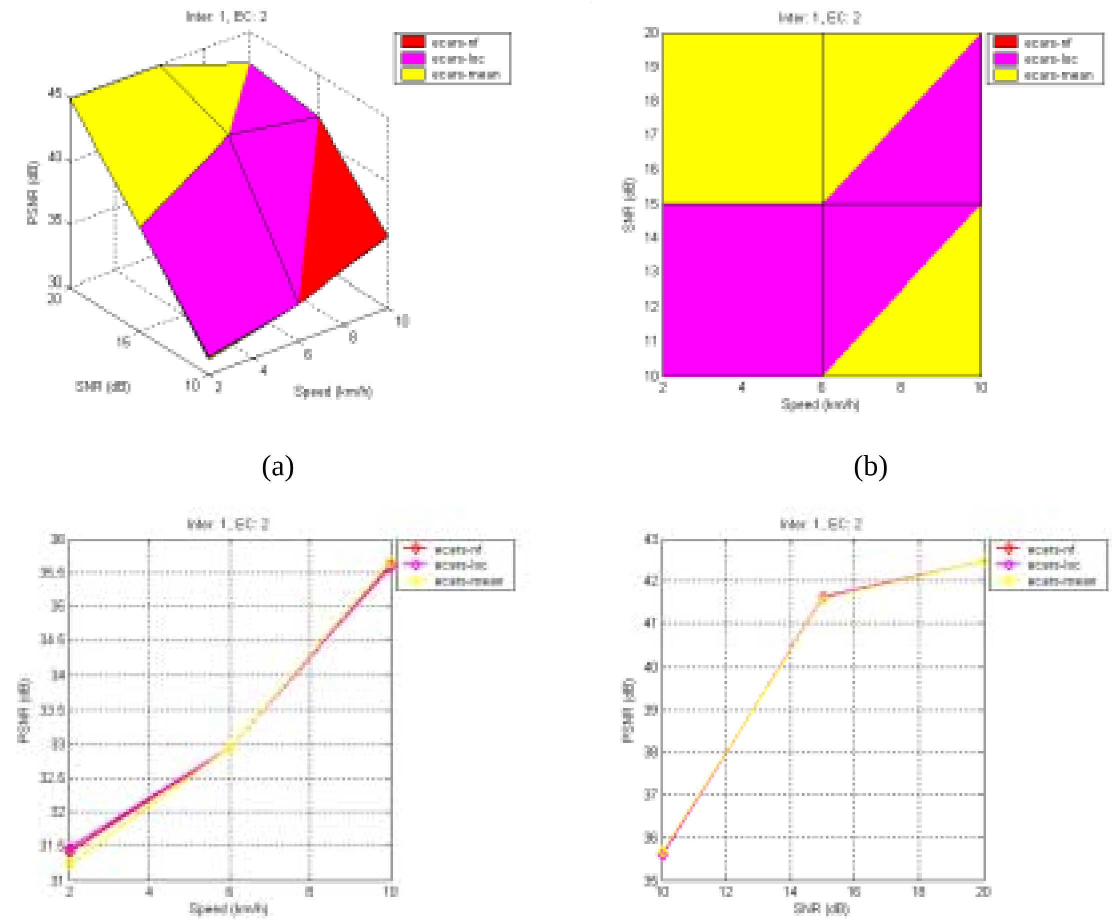


Figure 53. Frame-by-frame PSNR of Methods "ecars-nf", "ecars-loc", and "ecars-mean" at 10 km/h and SNR=20 dB with Case (0, 1) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 54. Performance of Methods “ecars-nf”, “ecars-loc”, and “ecars-mean” at various wireless channel conditions with Case (1, 2) for Sequence “foreman”: (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

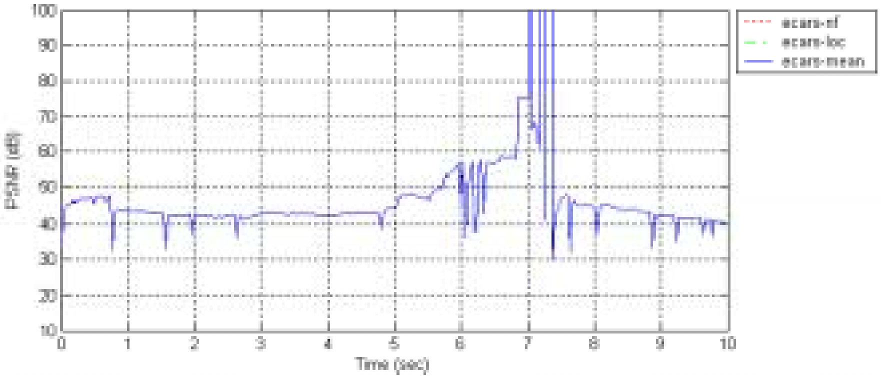
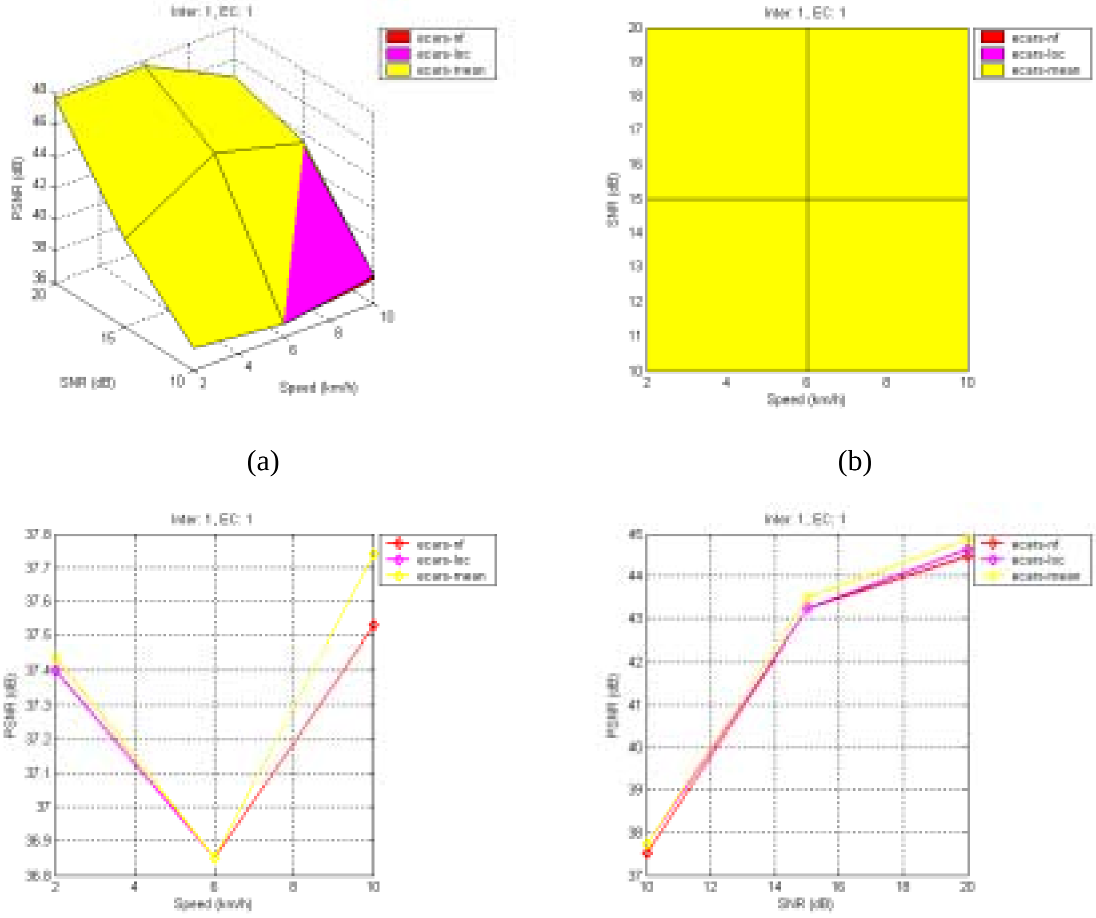


Figure 55. Frame-by-frame PSNR of Methods "ecars-nf", "ecars-loc", and "ecars-mean" at 10 km/h and SNR=20 dB with Case (1, 2) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 56. Performance of Methods "ecars-nf", "ecars-loc", and "ecars-mean" at various wireless channel conditions with Case (1, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

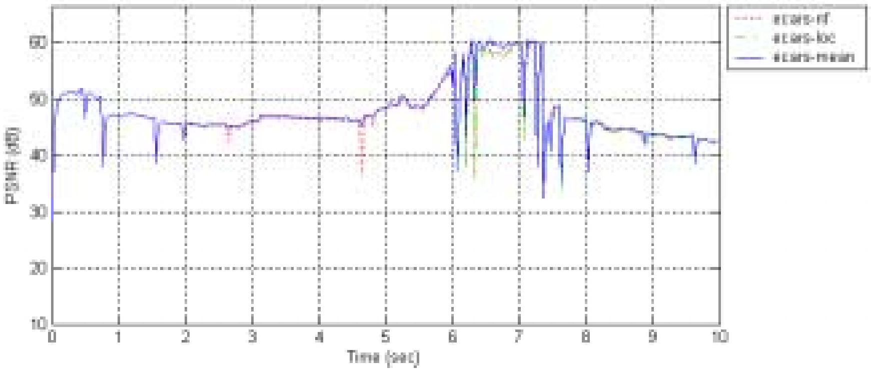


Figure 57. Frame-by-frame PSNR of Methods "ecars-nf", "ecars-loc", and "ecars-mean" at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence "foreman"

To understand why the improvement of "ecars-mean" over "ecars-nf" is marginal, let us consider that the pixel value can be modeled as an AR(1) Gauss-Markov process [51].

$$x_n = \rho x_{n-1} + \sigma\sqrt{1-\rho^2}\, e_n \tag{4.12}$$

Using the pixel value of the previous frame to conceal the error in the current frame can result in the distortion as follows. That is, c_j is from the previous frame.

$$E\left[(x_n - x_{n-1})^2\right] = 2\sigma^2(1-\rho) \tag{4.13}$$

If the previous frame is corrupted, we use the one before the previous frame to conceal the error in the current frame. That is, c'_j is from the one before the previous frame. The resulting distortion is:

$$E\left[(x_n - x_{n-2})^2\right] = 2\sigma^2\left(1-\rho^2\right) \tag{4.14}$$

If $\rho \to 1$ (which is reasonable for natural images), the distortions in (4.13) and (4.14) are almost identical. Therefore, we conclude that the gain g'_j does not change a lot with the feedback.

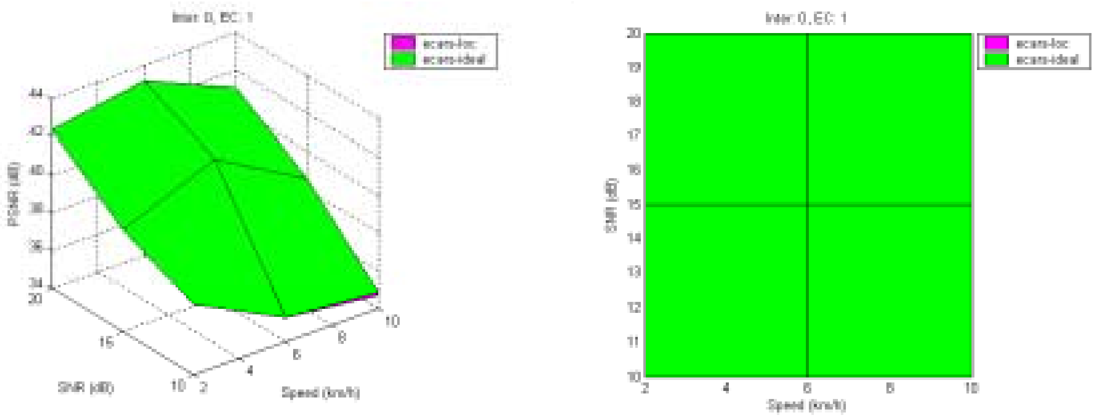
In addition to the analysis, let us examine "ecars-ideal" with respect to "ecars-loc" to see the limitation of the best ECARS method with feedback in the following subsection.

4.4.4. ECARS with EC Method Known vs. ECARS without EC Method Known

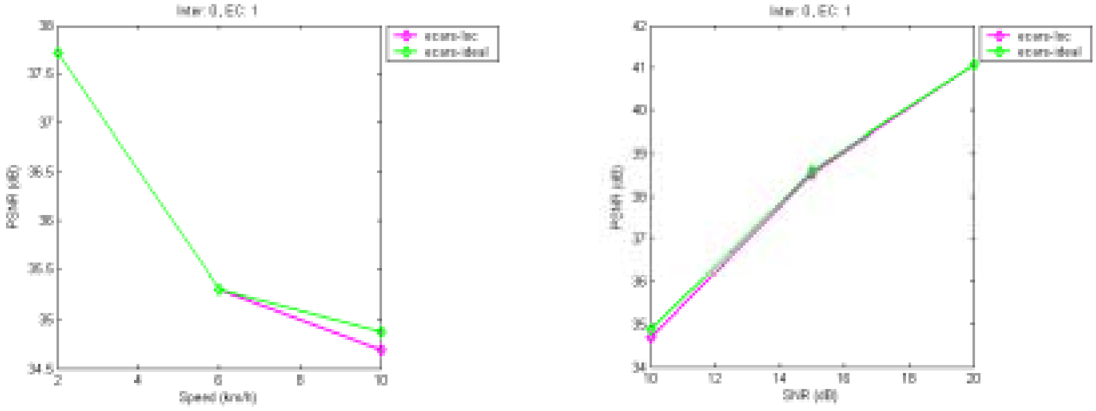
We will compare the results of ECARS with feedback knowing exactly the EC method "ecars-ideal", with ECARS with feedback without knowing exactly the EC method "ecars-loc" here. In general, rate shaper does not know exactly the EC method used at the receiver. Thus, "ecars-ideal" is usually not the case since the receiver might not want to disclose its own EC method. The comparison in this subsection is served for information only.

Results of Case (0, 1) are shown in Figure 58 and Figure 59; results of Case (1, 2) are shown in Figure 60 and Figure 61; and results of Case (0, 1) are shown in Figure 62 and Figure 63.

The performance in terms of the overall PSNR at various wireless channel conditions is shown in Figure 58, Figure 60, and Figure 62. Figure 58 (a), Figure 60 (a), and Figure 62 (a) show the 3-D plots of the overall PSNR. Figure 58 (b), Figure 60 (b), and Figure 62 (b) show the top views (seen from the top of the z-axis) of the 3-D plots. The color shown in the top view represents the color of the method that outperforms the others. At all wireless channel conditions, "ecars-nf" outperforms "n-ecars". Figure 58 (c), Figure 60 (c), and Figure 62 (c) show the overall PSNR at various speeds at $\text{SNR} = 10\ \text{dB}$. Figure 58 (d), Figure 60 (d), and Figure 62 (d) show the overall PSNR at various SNR at $\text{speed} = 10\ \text{km/h}$. Frame-by-frame PSNR performance is shown in Figure 59, Figure 61, and Figure 63. We also see that "ecars-nf" outperforms "n-ecars".



(a) (b)



(c) (d)

Figure 58. Performance of Methods "ecars-loc" and "ecars-ideal" at various wireless channel conditions with Case (0, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

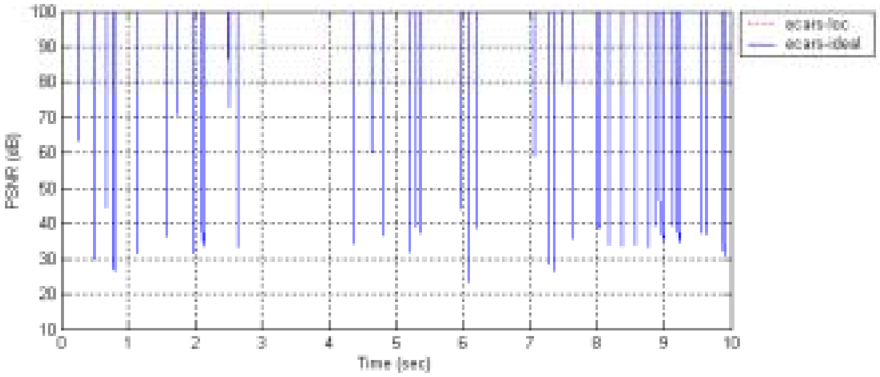
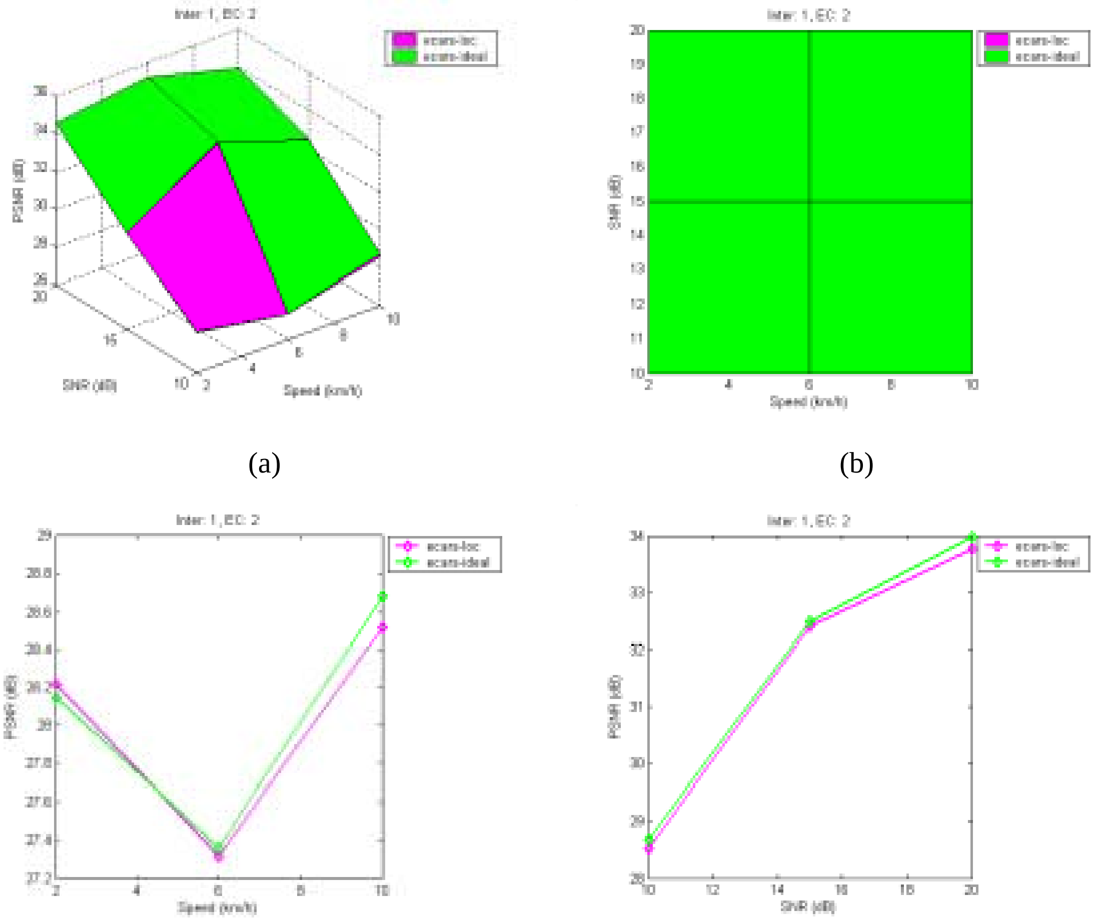


Figure 59. Frame-by-frame PSNR of Methods "ecars-loc" and "ecars-ideal" at 10 km/h and SNR=20 dB with Case (0, 1) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 60. Performance of Methods "ecars-loc" and "ecars-ideal" at various wireless channel conditions with Case (1, 2) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

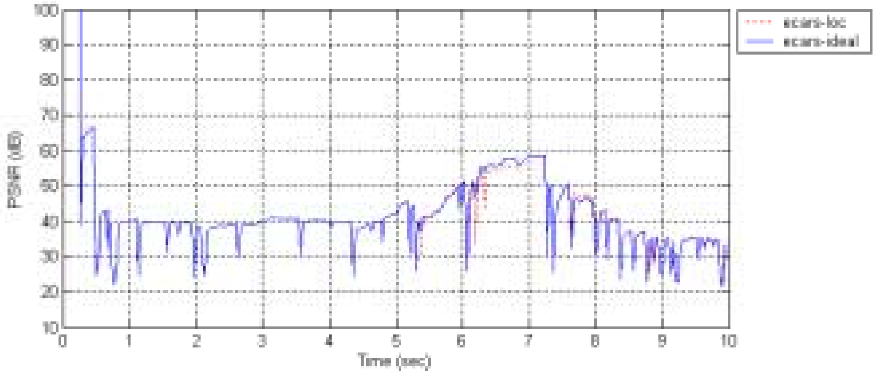
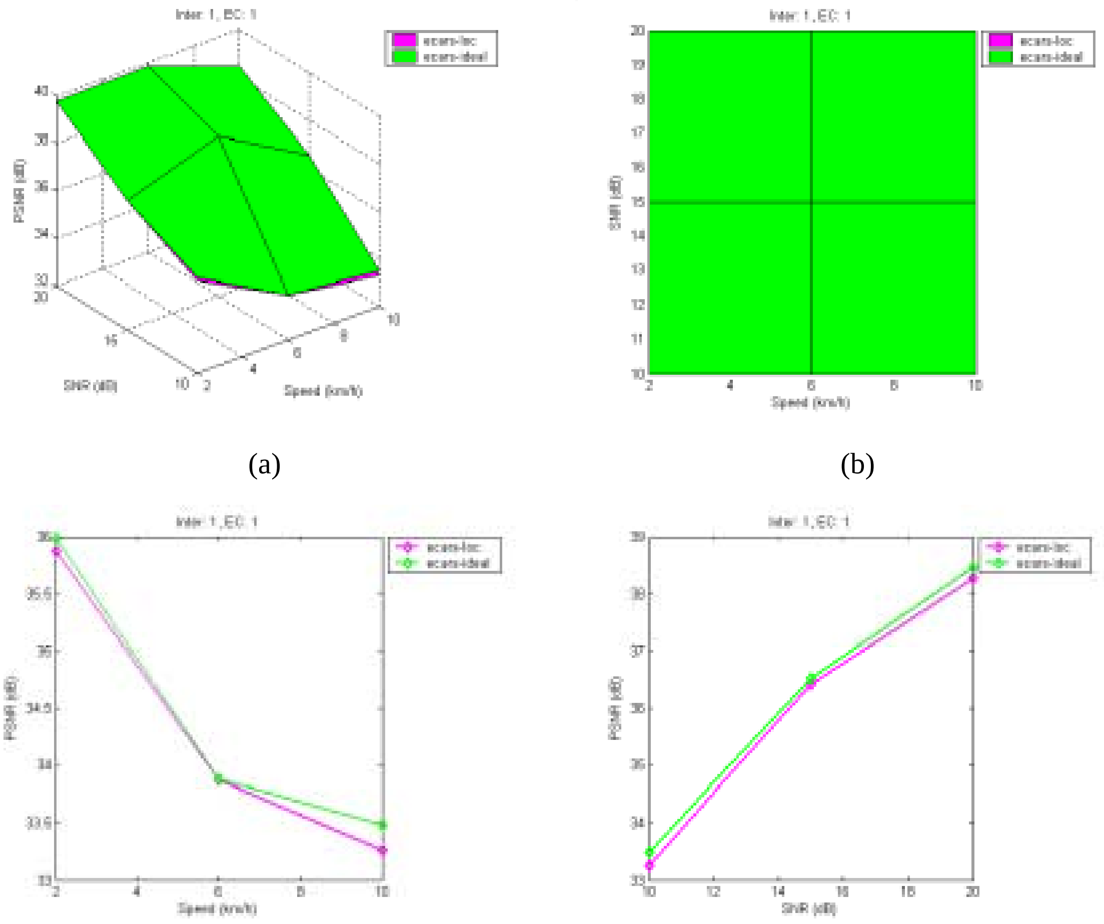


Figure 61. Frame-by-frame PSNR of Methods "ecars-loc" and "ecars-ideal" at 10 km/h and SNR=20 dB with Case (1, 2) for Sequence "foreman"



(a) (b)

(c) (d)

Figure 62. Performance of Methods "ecars-loc" and "ecars-ideal" at various wireless channel conditions with Case (1, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

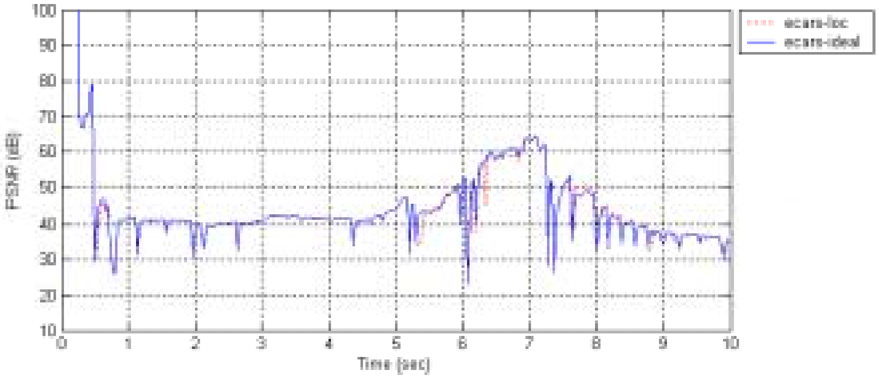


Figure 63. Frame-by-frame PSNR of Methods “ecars-loc” and “ecars-ideal” at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence “foreman”

We can see that “ecars-ideal” on the average outperforms “ecars-loc” by 0.1~0.3 dB, which falls in the same range by which “ecars-mean” outperforms “ecars-loc”. We conclude that “ecars-mean” is an almost ideal ECARS method with feedback without requiring the knowledge of the exact EC method used at the receiver.

4.4.5. All Methods

Sample results of methods where exact EC methods are not required are shown here. Figure 64 shows an example of how each method allocates the rates among sublayers. With the bandwidth constraint specified, Method “rand” allocates the rates randomly among the 27 sublayers; Method “upprs1” allocates the rates equally among the 27 sublayers; Method “upprs2” allocates the rates to the earlier sublayers; and Methods “n-ecars”, “ecars-nf”, “ecars-loc”, and “ecars-mean” allocate the rates smartly among the 27 sublayers (some sublayers are even not allocated with rates) depending on different definitions of the MB gain. The bit allocation processes of “n-ecars”, “ecars-nf”, “ecars-loc”, and “ecars-mean” happen automatically by the proposed two-stage R-D optimization considering the current network condition.

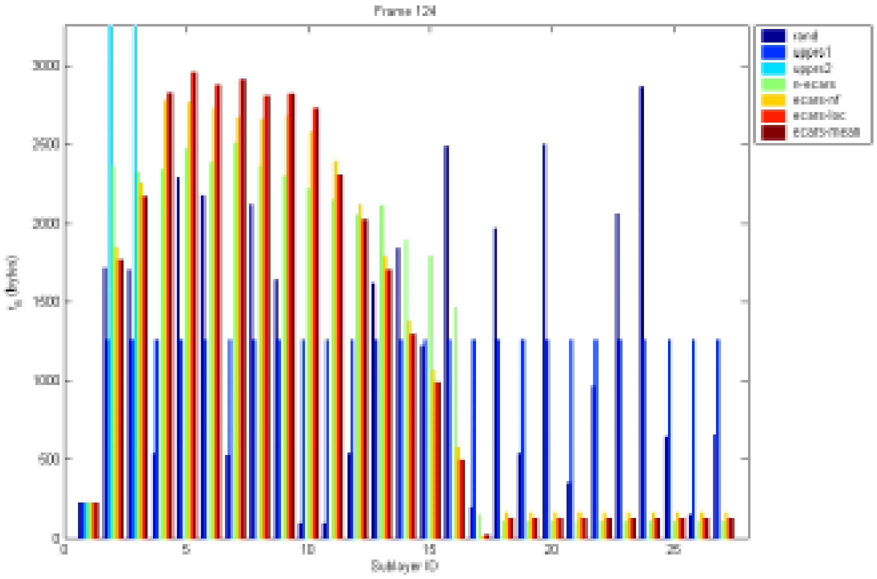
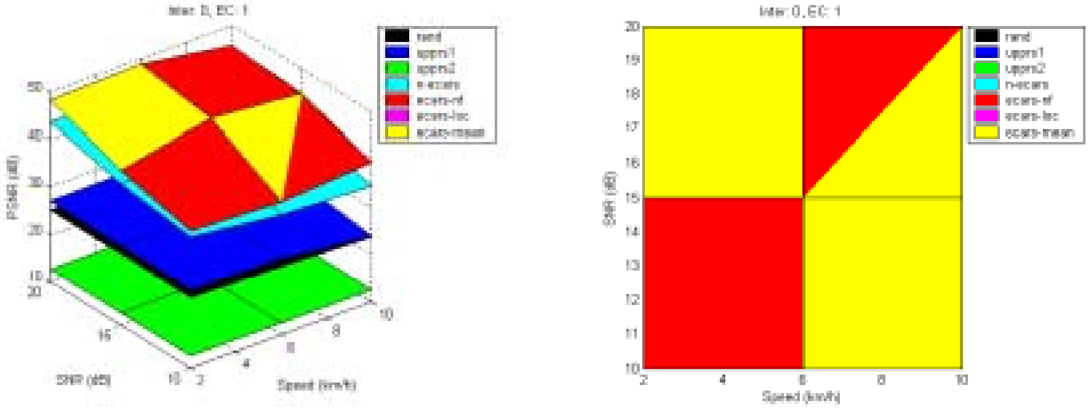


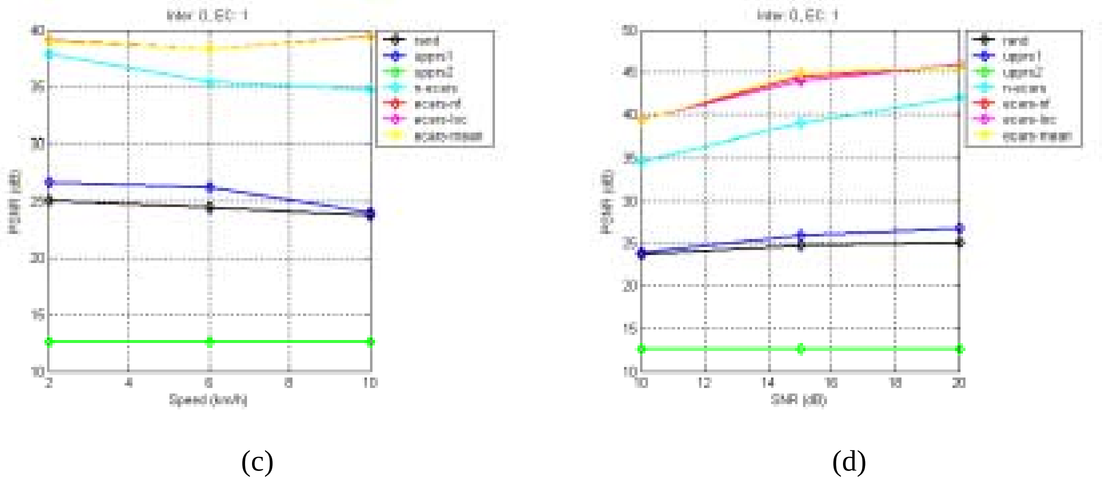
Figure 64. Sublayer bit allocations of all methods at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence "foreman"

We then recap the performance of all seven methods (shown from Figure 65 to Figure 70). We can see that "ecars-mean" outperforms all the others most of the time. Rate shaping based methods "n-ecars", "ecars-nf", "ecars-loc", and "ecars-mean" outperform naïve methods "rand", "upprs1", and "upprs2" at all time.



(a)

(b)



(c) (d)

Figure 65. Performance of all methods at various wireless channel conditions with Case (0, 1) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

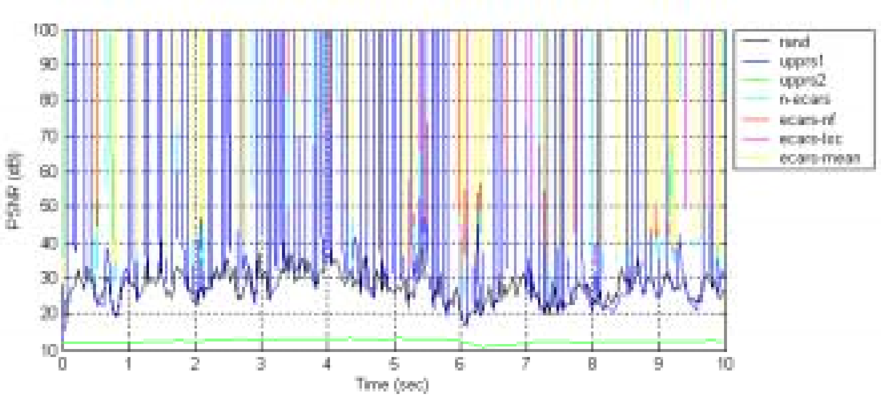
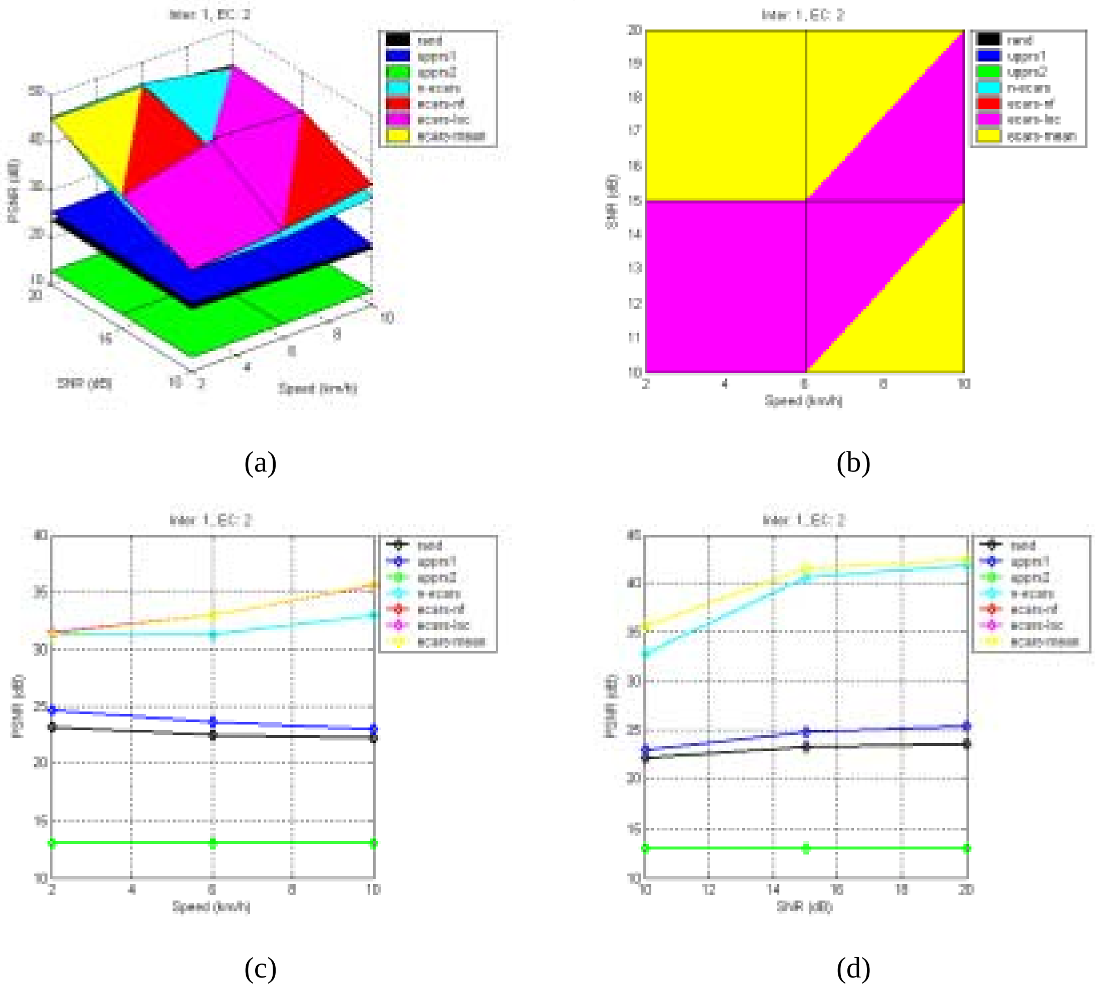


Figure 66. Frame-by-frame PSNR of all methods at 10 km/h and SNR=20 dB with Case (0, 1) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 67. Performance of all methods at various wireless channel conditions with Case (1, 2) for Sequence "foreman": (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

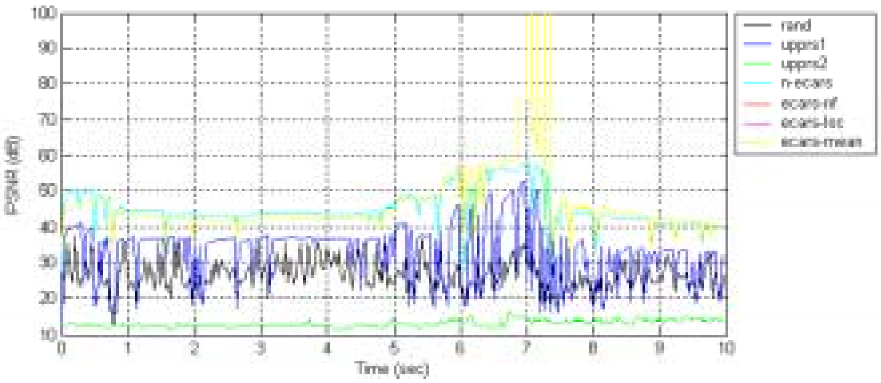
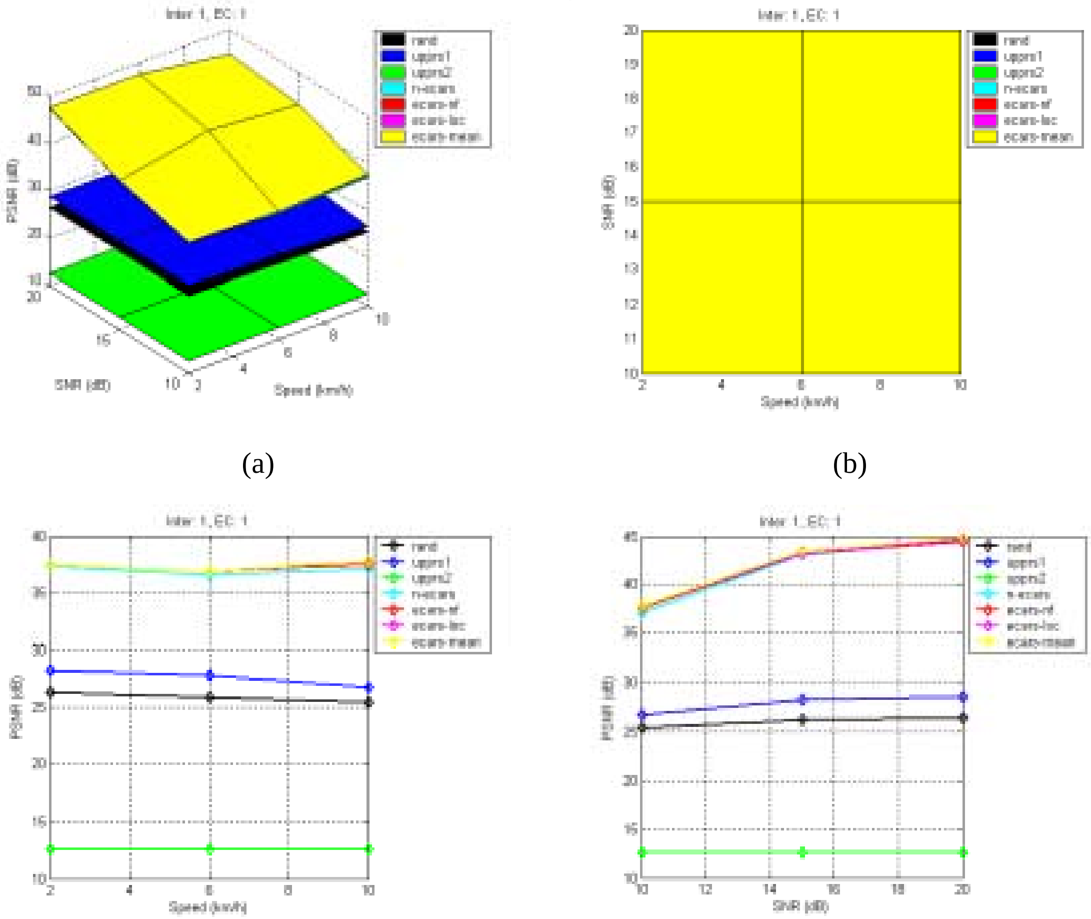


Figure 68. Frame-by-frame PSNR of all methods at 10 km/h and SNR=20 dB with Case (1, 2) for Sequence "foreman"



(a)

(b)

(c)

(d)

Figure 69. Performance of all methods at various wireless channel conditions with Case (1, 1) for Sequence “foreman”: (a) 3-D view of PSNR at various speeds and SNR; (b) top view of PSNR at various speeds and SNR; (c) PSNR at various speeds; (d) PSNR at various SNR

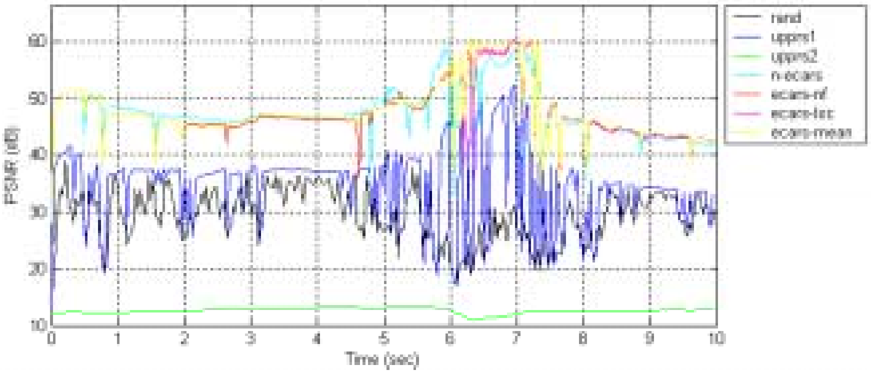


Figure 70. Frame-by-frame PSNR of all methods at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence “foreman”

From Figure 71 to Figure 73, the performance in overall PSNR of different sequences is shown. Given the same network condition, Sequence “akiyo” has higher PSNR than “foreman”; and Sequence “foreman” has higher PSNR than “stefan”. The sequence with texture that is more complex and faster motion, such as “stefan”, gives smaller PSNR value given the same bandwidth budget.

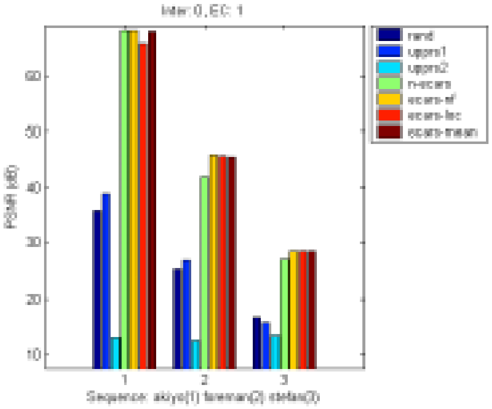


Figure 71. Performance of all methods at 10 km/h and SNR=20 dB with Case (0, 1) for Sequences “akiyo”, “foreman”, and “stefan”

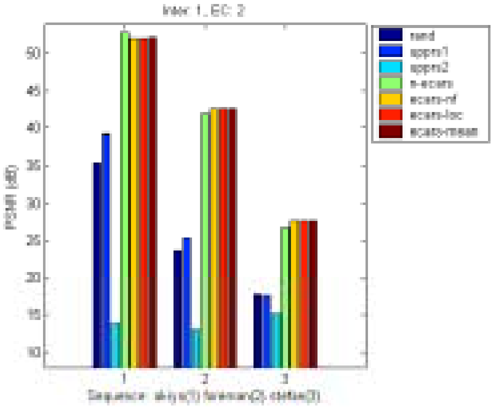


Figure 72. Performance of all methods at 10 km/h and SNR=20 dB with Case (1, 2) for Sequences “akiyo”, “foreman”, and “stefan”

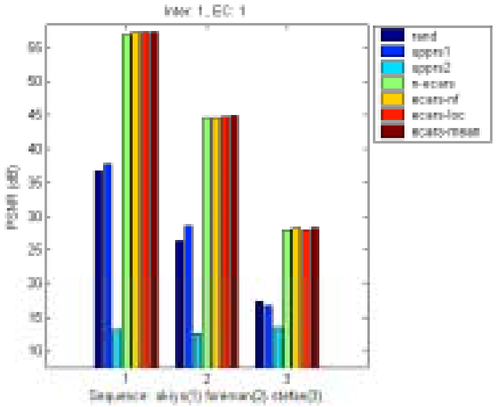
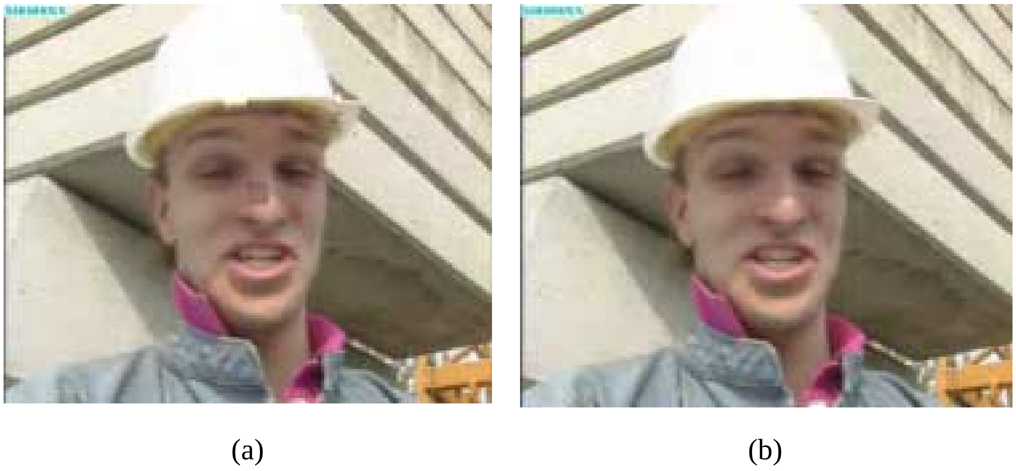


Figure 73. Performance of all methods at 10 km/h and SNR=20 dB with Case (1, 1) for Sequences “akiyo”, “foreman”, and “stefan”

Finally, sample frames of Method “n-ecars” and “ecars-mean” are shown in Figure 74 to demonstrate visually the merit of ECARS with location and mean information as feedbacks.


(a)

(b)

Figure 74. A sample frame of (a) “n-ecars” and (b) “ecars-mean” at 10 km/h and SNR=20 dB with Case (1, 1) for Sequence “foreman”

4.5. Conclusion

We proposed in this paper error concealment aware rate shaping (ECARS) for video transport over wireless networks. ECARS is applied to pre source- and channel- coded video. ECARS first evaluates the gain of sending the MB of the precoded video, as opposed to not sending it but reconstructing it by EC. Then given a certain packet loss rate, the expected accumulated gain can be derived and be included in the R-D optimization problem formulation. Finally, ECARS performs R-D optimization by the proposed two-stage R-D optimization approach.

Two types of ECARS algorithms: without feedback and with feedback from the receiver, were proposed to account for the frame dependency problem in rate shaping. In the case of no feedback, ECARS evaluates the MB gain considering a particular EC method used at the receiver. The case of ECARS with feedback is needed if the video is predictive coded, and/or the EC method performed at the receiver utilizes the temporal information. In order to incorporate the frame dependency into the rate shaping process, we propose to send the location (and mean) of the corrupted MB back to the sender, and use such feedback information to determine the MB gain in the R-D optimized ECARS. Experiments have shown that ECARS is better than other naïve methods. Moreover, ECARS has improved performance with the aids of the feedback information.

The way the MB are grouped into sublayers in this paper is fixed and is not part of the ECARS R-D optimization, since how MB are grouped should be considered in the precoding process but not in the rate shaping stage. In the future, we can consider R-D optimization on the way MB are grouped into sublayers (that is, the number of source-coded symbols that go to each sublayer) given the rate shaping problem is solved.

5. Modeling of Video Traffic

We present a new stochastic process called the *punctured* autoregressive (AR) process, and use it to model the variable bit rate (VBR) video traffic. To model the VBR video traffic, we propose to use punctured autoregressive processes modulated by a doubly Markov process. The doubly Markov process models the state of each video frame while the autoregressive processes describe the number of bits of each frame at each state. The punctured autoregressive process considers the timing information between frames of the same state and thus gives better modeling performance. The model captures the long-range dependency (LRD) characteristics as well as the short-range dependency (SRD) characteristics of the video traffic. Queuing behavior of the punctured autoregressive process is also closer to the real video traffic than the conventional autoregressive process.

This chapter is organized as follows. We first introduce some prior work on VBR traffic modeling. We then introduce "punctured AR process" as opposed to the conventional AR process. The proposed doubly Markov process modulated punctured AR process as well as the conventional non-punctured version are both described for their usage in traffic modeling. Experiments are conducted to compare the proposed punctured process with the non-punctured process. Finally, conclusion remarks are given.

5.1. Introduction

To both the video service providers and the network designers, it is important to have a good model for the video traffic. A good model for video traffic allows for better admission control, scheduling, network resource allocation policies, etc., that guarantee a desired quality of service (QoS) as well as a better utilization of the network resources. A good model captures essential characteristics of the real video traffic. The synthetic trace generated by such a model can be used to test the network performance under a certain, for example, admission control policy. Therefore, the network designers can design a network that is more friendly to the video traffic and thus delivers a better video service.

Because of the importance of both the variable bit rate (VBR) video traffic modeling and the wireless channel dynamic modeling, many models have been proposed. To evaluate different models, there are generally three criteria to be taken into account— (1) A good model should capture the statistical properties of the real trace. A trace is defined as a sequence of data we intend to model. In the case of VBR video traffic modeling, a trace is a sequence of numbers, each represents the number of bits to encode each Group of Blocks (GOB)/frame/Group of Pictures (GOP). In the case of wireless channel modeling, a trace is a sequence of numbers, each represents the channel bit error rate (BER) at different time instant. The statistical properties should include those that are related to the long-range dependency (LRD) of the trace as well as those that are related to the short-range dependency (SRD) of the trace [1]. (2) The synthetic video trace should be similar to the real video trace in terms of the queuing behavior. Likewise, the synthetic trace of the wireless channel should be similar to the real trace in terms of the QoS behavior. (3) The model should be simple and easy to be analyzed. Related discussion on how to evaluate the performance of the models can be found in [28][37][47].

Video traffic modeling is challenging in several aspects. First, video is usually encoded with VBR to be adaptive to the video content. Second, depending on different coding schemes, the video trace has very different properties. Popular video coding schemes include H.263 [52] and MPEG-4 [40]. The video frame can be Intra (I), Predictive (P), or Bidirectionally predictive (B) encoded. The GOP structure, which is defined as frames enclosed by two I frames including the leading I frame, can be fixed or dynamic. A fixed GOP structure that is commonly used is IBBPBBPBBPBB.

Existing work for video traffic modeling includes DAR [24], which fails to capture the LRD property of the video traffic. Models such as [14][36][58][67] are constrained by a fixed GOP structure. Non-statistical methods as [33][62] are usually more difficult to analyze. The wavelet-based model [38] and the autoregressive process (AR) based model [35] do not capture the dynamic nature of the video traffic nicely. In general, it is preferred to use a Markov chain like process to model the dynamic nature of the video traffic. Models such as [30] are too complex. We propose to build a model based on the work done by [53], which models the video traffic as a doubly Markov process with AR processes inside each Markov state. However, the model of [53] does not use the timing information between frames of the same Markov state. Here, we explicitly use the timing information between frames of the same Markov state and refer to the new model as a "*doubly Markov process modulated punctured AR process*". It is shown that the proposed model outperforms the model of [53] in terms of statistics, both SRD and LRD, queuing behavior, and has the same complexity in terms of the number of model parameters.

5.2. Punctured Autoregressive Modeling

A conventional autoregressive (AR) process x_n is defined as follows:

$$x_n = \rho x_{n-1} + \sigma\sqrt{1-\rho^2}e_n,\ n = 1,2,\cdots \tag{5.1}$$

where n is the time index of the process, ρ describes the dependency of the sample at time n with the previous sample at time $n-1$, σ^2 is the variance of the process x_n, and e_n is a Gaussian random variable with mean 0 and variance 1 to characterize the random nature of the process.

Consider the case where more than one AR processes are interleaved together such as shown in Figure 75 (a). At time instant 1, AR process x_n takes place; at time instant 2, AR process y_n takes place; and so on. Conventional method to train two sets of AR parameters of sequences x_n and y_n is by splitting the single process in Figure 75 (a) to two separate processes as shown in Figure 75 (b) and Figure 75 (c). Each one of the processes represents the training sequence of x_n or y_n regardless of the time index associated with each sample. For example, the sequence in Figure 75 (b) is used as if samples are:

$$\tilde{x}_1\tilde{x}_2\tilde{x}_3\tilde{x}_4\tilde{x}_5\tilde{x}_6\tilde{x}_7\tilde{x}_8\tilde{x}_9 \tag{5.2}$$

where $\tilde{x}_1 = x_1$, $\tilde{x}_2 = x_3$, and so on.

$$\begin{tikzcd}
 & x_1 x_3 x_4 x_5 x_6 x_9 x_{11} x_{12} x_{13} \ldots \\
 & \text{(b)} \\
x_1 y_2 x_3 x_4 x_5 x_6 y_7 y_8 x_9 y_{10} x_{11} x_{12} x_{13} y_{14} \ldots \arrow[uur] \arrow[dr] & \\
\text{(a)} & y_2 y_7 y_8 y_{10} y_{14} \cdots \\
 & \text{(c)}
\end{tikzcd}$$

Figure 75. Two interleaved autoregressive processes x_n and y_n: (a) the interleaved process; (b) autoregressive process x_n; (c) autoregressive process y_n.

To synthesize samples using this model, two separate AR processes are generated with parameters trained by $\tilde{x}_n$ and $\tilde{y}_n$. Synthetic samples generated by parameters trained by $\tilde{x}_n$ are taken one after the other to form the final synthetic process. Synthetic samples generated by parameters trained by $\tilde{y}_n$ are formed in a similar way. In brief, "no synthetic samples are skipped".

It can be seen that the conventional method does not take into account the timing information of the two processes in both the training and synthesis stages. To consider the timing information, we propose to train and synthesize the AR processes as follows (Figure 76). First split the single process in Figure 76 (a) to two separate processes as shown in Figure 76 (b) and Figure 76 (c). Notice that the timing information is utilized while leaving the sample of x_n blank if at some particular time instance the original process is with the other process y_n (Figure 76 (b)). Similarly, the sample of y_n is blank if at some time instance the original process is with the other process x_n (Figure 76 (c)).

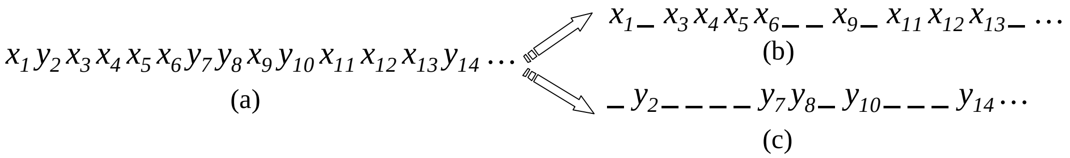

$x_1 y_2 x_3 x_4 x_5 x_6 y_7 y_8 x_9 y_{10} x_{11} x_{12} x_{13} y_{14} \cdots$
(a)
$x_1 \_\, x_3 x_4 x_5 x_6 \_\,\_\, x_9 \_\, x_{11} x_{12} x_{13} \_\, \cdots$
(b)
$\_\, y_2 \_\,\_\,\_\,\_\, y_7 y_8 \_\, y_{10} \_\,\_\,\_\, y_{14} \cdots$
(c)

Figure 76. Two interleaved autoregressive processes x_n and y_n: (a) the interleaved process; (b) punctured autoregressive process x_n; (c) punctured autoregressive process y_n.

To train the punctured AR processes, we find the value $\tilde{\rho}_X$ of the process $\tilde{x}_1\tilde{x}_2\tilde{x}_3\tilde{x}_4\tilde{x}_5\tilde{x}_6\tilde{x}_7\tilde{x}_8\tilde{x}_9$. We also construct the histogram of the sample spacing. For example in Figure 76 (b), the histogram has values [5, 2, 1] at spacing [1, 2, 3]. Recall that in the simplest case where all the samples are adjacent to each other, the histogram has values [8, 0, 0] at spacing [1, 2, 3]. The ρ_X of the process in Figure 76 (b) can be found by solving the following equation:

$$\frac{5\rho_X^1 + 2\rho_X^2 + 1\rho_X^3}{(5+2+1)} = \tilde{\rho}_X \tag{5.3}$$

The value of ρ_Y in Figure 76 (c) can be solved in the same way.

To synthesize samples using this model, two separate AR processes are generated by ρ_X and ρ_Y. The final synthetic process is formed by some means of multiplexing/modulation of the two synthetic processes. Samples are not taken one after the other but with consideration of how separate the samples of the same type are. In brief, "some synthetic samples are skipped".

Punctured AR processes can be modulated by different processes. In this study, we specifically consider the Markov process.

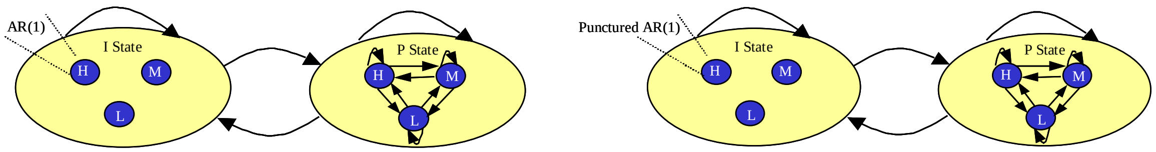
5.3. Variable Bit Rate Video Traffic Modeling

Before we proceed to the proposed model for VBR video traffic, let us briefly describe the method proposed by [53] with which we will compare (Figure 77). The model comprises of two layers of Markov process. Without loss of generality, we consider two frame types, I and P. The doubly Markov process models I and P frame transitions, as well as different frame activities. The outer Markov process describes how I and P frames transit. The frame type can be further categorized into different activity levels. Frames of higher activity level consume more number of bits, while frames of lower activity level consume less number of bits. The inner Markov process describes how the frames of different activity levels transit. This model is not constrained by a fixed GOP structure. There are six states in total, namely, I frame in high activity level, I frame in medium activity level, I frame in low activity level, P frame in high activity level, P frame in medium activity level, and P frame in low activity level.

Each state is modeled as an AR process with different AR parameters. We can consider this as a slightly more complex process than the one described in Figure 75. There are two states in Figure 75 (a) while there are six states in this model. The AR parameters are trained in the way described in Figure 75 of in a conventional non-punctured manner. The synthetic trace is generated by procedures described in the last section.

To simply the model, the inner Markov process of I frames is characterized by initial probabilities of three activity levels only. Since I frames are usually far apart in a video sequence, they do not need to modeled as a Markov process. Such simplification will have similar performance as the full model. We call this model Method 1, as shown in Figure 77 (a), for later discussion.

We propose to model the VBR video traffic as a doubly Markov modulated punctured AR process. This new model explicitly considers the timing information between two frames of the same state. Again, this is a slightly more complex process than the one described in Figure 76. There are two states in Figure 76 (a) while there are six states in this model. The AR parameters are trained in the way described in Figure 76 in a punctured manner. The synthetic trace is generated by procedures described in the last section. We call this model Method 2, as shown in Figure 77 (b), for later discussion.



(a) (b)

Figure 77. Models for VBR video traffic— (a) Method 1: Doubly Markov modulated AR process; (b) Method 2: Doubly Markov modulated *punctured* AR process.

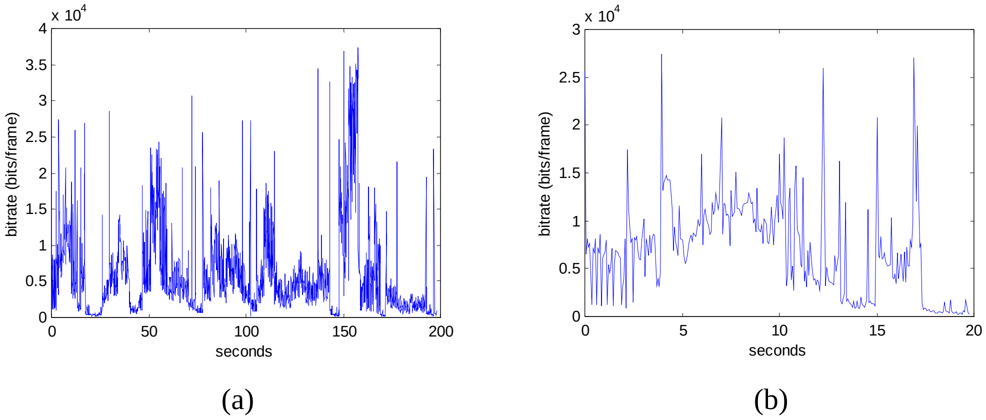
5.4. Experiment

We now compare the performance of the non-punctured Method 1 with the proposed punctured Method 2. To evaluate the performance of the models, we consider four performance metrics: (1) first order statistics by means of the quantile-quantile (Q-Q) plot; (2) second order statistics by means of the auto-correlation function (ACF); (3) LRD property by means of the Hurst parameter from the range/standard deviation (R/S) plot; and (4) queuing behavior by means of the packet loss rate and the queuing delay. Definitions of the performance metrics can be found in [28][37][47].

The experiment setting is as follows. Two different types of TV programs are recorded: “news” as shown in Figure 78 (a) and “talk show” as shown in Figure 78 (b). The two TV programs are encoded using video compression codec H.263 to generate the real video traces. Both of them are encoded with frame rate of 15 frames/sec and with duration of 30 minutes each. The video trace of the clip “news” is shown in Figure 79 (a) and (b) with different scales. The video traces are then fed into both models: Method 1 and Method 2 to generate synthetic traces. The performances of the two models are evaluated.

(a) (b)

Figure 78. Test videos: (a) news; (b) talk show



(a) (b)

Figure 79. Sample traces from the TV program “news”: (a) a 200 second trace; (b) a 20 second trace.

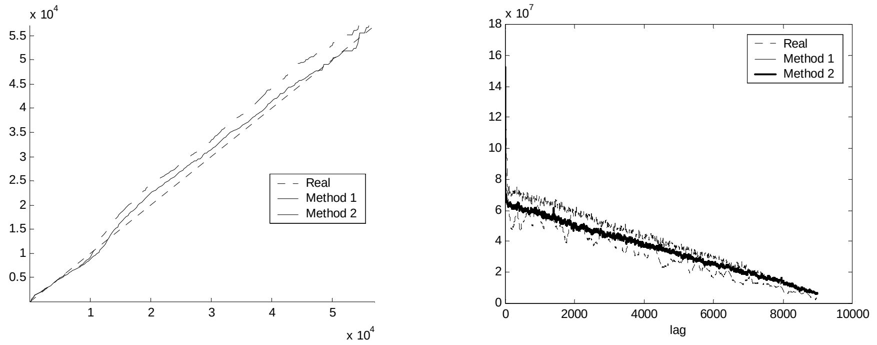
The performance comparison is summarized in Table 2. MSE refers to mean square error compared to the real trace. It can be seen that the proposed Method 2 outperforms Method 1 in all five aspects. The MSE improvement is computed by |(MSE of Method 2) – (MSE of Method 1)| / (MSE of Method 1). Detailed discussion associated with each performance metric will be presented later.

Table 2 Summary of performance comparison between modeling methods Method 1 and Method 2

	MSE of Q-Q plot	MSE of ACF	Hurst parameter	MSE of packet loss rate	MSE of queuing delay
Real			0.2530		
Method 1	1.2048e+7	1.1126e+14	0.1538 (98e-4 in SE)	9.7846e-4	4.6553e-4
Method 2	0.1747e+7	0.4054e+14	0.2725 (3.8025e-4 in SE)	9.6003e-4	3.4044e-4
MSE improvement	85.50%	63.56%	96.12%	1.88%	26.87%

Figure 80 (a)(b) shows the performance of both models in terms of first and second order statistics. The first order statistics in Figure 80 (a) is shown by the Q-Q plot. The Q-Q plot is constructed by a pair of cumulative distribution functions (CDF). The closer one CDF is to the other CDF in one pair, the more the curve will look like a straight line $y = x$. In Figure 80 (a), a dotted straight line is plotted as a reference. We have two pairs of CDF to compare: the synthetic trace by Method 1 with respect to the real video trace and the synthetic trace by Method 2 with respect to the real video trace. The dashed-dotted curve in Figure 80 (a) refers to the first pair. The solid curve in Figure 80 (a) refers to the second pair. It is shown that the curve of Method 2 is closer to the reference dotted straight line than the curve of Method 1.

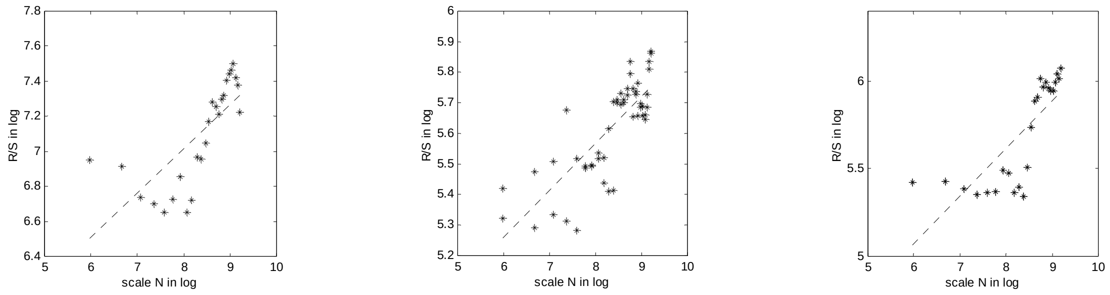
The second order statistics in Figure 80 (b) is shown by the ACF. The dotted curve in Figure 80 (b) refers to the ACF of the real video trace. The dashed-dotted curve in Figure 80 (b) refers to the ACF of the synthetic trace by Method 1. The solid curve in Figure 80 (b) refers to the ACF of the synthetic trace by Method 2. It is shown that the curve of Method 2 is closer to the reference curve than the curve of Method 1.



(a) (b)

Figure 80. First and second order statistics of the synthetic traces generated by Method 1 and Method 2 with respect to the real video trace of the clip “news”. (a) First order statistics: Q-Q plot; (b) Second order statistics: ACF.

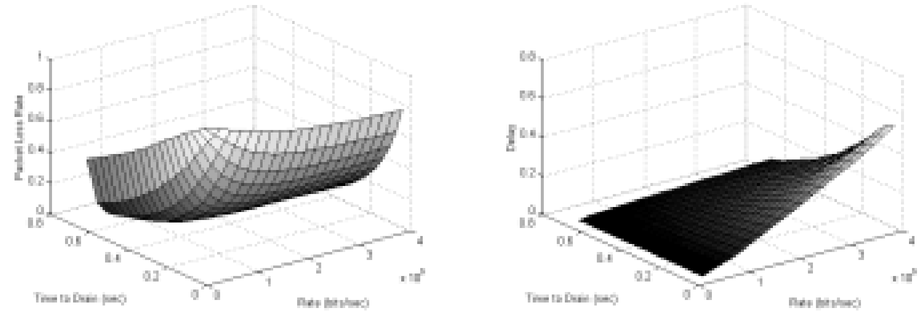
The LRD property in Figure 81 is shown by means of the Hurst parameter, which is the slope of the linear regression line of the points in a R/S plot. Figure 81 (a) shows the R/S plot of the real video trace. The linear regression line is shown as a dotted line. Figure 81 (b) shows the R/S plot of the synthetic trace by Method 1. Figure 81 (c) shows the R/S plot of the synthetic trace by Method 2. It is shown that the synthetic trace by Method 2 has closer Hurst parameter value to the real video trace than the synthetic trace by Method 1.



(a) (b) (c)

Figure 81. LRD properties of three traces by Hurst parameter from the R/S plots: (a) real video trace; (b) synthetic trace by Method 1; (c) synthetic trace by Method 2

The queuing behavior of the traces is evaluated by means of the packet loss rate and the queuing delay. Packet loss rate and queuing delay are measured at different drain rates and buffer sizes. The network is a leaky bucket with drain rate R and time to drain M/R, where M is the buffer size. The queuing performance of the real trace is shown in Figure 82. The queuing performances of the synthetic traces by both Method 1 and Method 2 have similar look. However, the synthetic trace by Method 2 has smaller MSE in both the packet loss rate and the delay than the synthetic trace by Method 1 (Table 2).



(a) (b)

Figure 82. Queuing behavior of the real video trace: (a) packet loss rate; (b) queuing delay.

5.5. Conclusion

We proposed a new punctured AR processes to model the video traffic. The punctured AR processes are modulated by Markov processes. The punctured AR processes explicitly consider the timing information between samples of each state. Thus, it outperforms the conventional approach in VBR video traffic modeling. A good set of performance metrics are experimented showing the novelty of the proposed model in different aspects, especially in the queuing behavior.

6. Summary and Future Directions

This thesis provides an error-resilient rate shaping framework for streaming video over packet-loss networks. The challenges in transmitting multimedia data over packet-loss networks urge the need of closer collaboration between the application layer and the network layer. The proposed error-resilient rate shaping acts as a filtering process to adapt the precoded video from the application layer according to the network conditions given by the network layer.

After introducing the fundamentals in Chapter 2, Chapter 3 and Chapter 4 constitute the proposed error-resilient rate shaping system for streaming the enhancement layer video and base layer video. FGRS is applied to streaming enhancement layer video and ECARS is applied to streaming base layer video.

The contribution of the thesis lies in:

- *Error-resilient rate shaping for pre source- and channel- coded video*

  None of the prior rate shaping work considers rate adaptation of pre source- and channel-coded video. Pre source- and channel- coded video is useful for streaming over packet-loss networks. This thesis provides a R-D optimized solution for streaming the pre source- and channel- coded video.

  - Given the *gain* embedded in the bitstream, FGRS and ECARS consume on the average <0.01% and <1% (the denominator is the bit rates of the source-coded bitstream), respectively, of the original precoded video to carry the gain information ("meta-data"). The performance improvement of FGRS and ECARS in PSNR over non rate shaping based methods is on the average 8 dB. On the other hand, if the gain is not embedded in the bitstream for rate shaping, no extra bits are needed to carry the gain information. Partial decoding to obtain the gain information is required.
  - FGRS provides an error-resilient rate shaping scheme for pre- channel coded MPEG-4 FGS bitstream.
  - ECARS provides an error-resilient rate shaping scheme for source- and channel-coded bitstream that is aware of the error concealment method used at the receiver.

- *Two-stage R-D optimization algorithm*

First stage of the two-stage R-D optimization algorithm provides a model-based approach to find the near-optimal solution. With the refinement of the second stage, the proposed two-stage R-D optimization algorithm finds the solution fast and accurately.

- *Error-resilient rate shaping vs. error-resilient video coding*

  The proposed error-resilient rate shaping does not need to alter the original video encoder and decoder, thus can be adopted by systems, in which tremendous amount of work to modify the video coders is needed.

- *Error-resilient rate shaping vs. joint source-channel coding*

  Joint source-channel coding techniques are limited by only providing the optimization at the time of encoding and are not suitable for streaming the precoded video. The encoded bitstream may not be optimal for transmission along a different path or along the same path at later time. Rate shaping can optimize the video streaming performance adaptive to each link.

Future work includes replacing streaming by simulcast with rate shaping. Simulcast is adopted in the current video streaming applications. Multiple streams of different qualities are sent concurrently to satisfy the needs of different users with different device capabilities and access bandwidths. We can see that the bandwidth utilization is not efficient given the concurrent transmissions of multiple streams. Moreover, it is not only inflexible in adapting the bit rates according to the available bandwidths, but also intolerant to the packet losses. The proposed error-resilient rate shaping provides a solution to overcome the shortages of the current video streaming with simulcast. The precoded video can be adapted to any bit rates to make use of the available bandwidths. Hence, the video quality will not be constrained to a limited amount of quality steps. The R-D optimized decision of rate shaping also guarantees the shaped bitstream to have the best error-resiliency given the current network condition.

For example, we can modify “End System Multicast (ESM)” [13], which currently adopts a simulcast approach for multicasting (Figure 83), to incorporate the propose rate shaping (Figure 84). In that, ESM does not need to transmit two video bitstream concurrently to the host computers. The rate shaping mechanism resides in the parent host computer adapts the rates for the child host computer. Each host computer can enjoy video with fine granular quality.

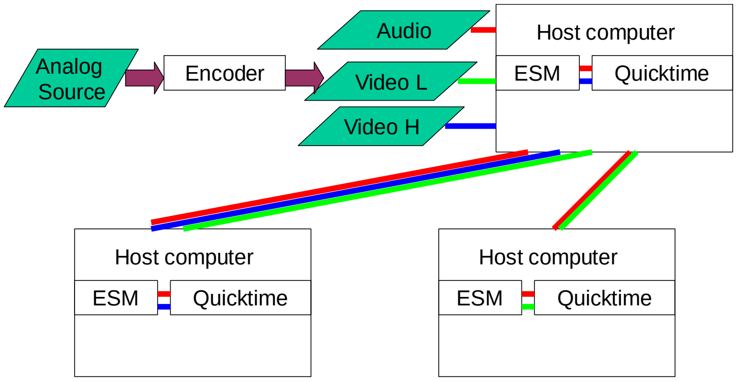


Figure 83. End system multicast (ESM) with simulcast

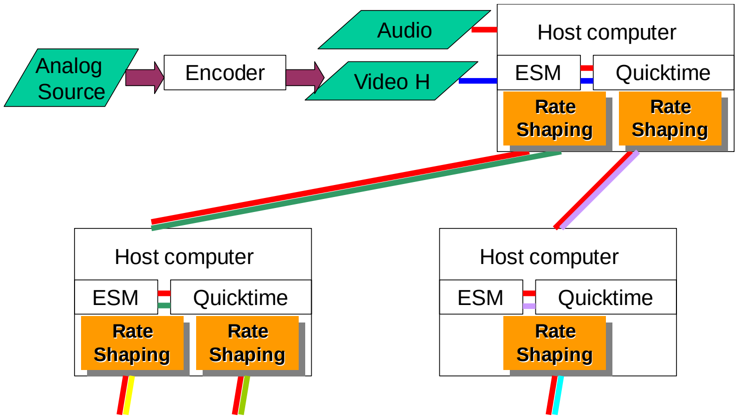


Figure 84. End system multicast (ESM) with rate shaping

Appendix A. Second-Generation Error Concealment

When transmitting video data over error prone channels, the video data may suffer from losses or errors. Error concealment is an effective way to recover the lost information at the decoder. Compared to other error control mechanisms such as FEC [61] and automatic retransmission request (ARQ) [34], error concealment has the advantages of not consuming extra bandwidth as FEC and not introducing retransmission delay as ARQ. On the other hand, error concealment can be used to supplement FEC and ARQ when both FEC and ARQ fail to overcome the transmission errors [5].

Error concealment is performed after error detection. That is, error concealment needs to be preceded with some error detection mechanism to know where the errors in the decoded video locate. For example, error detection provides information as which part of the received video bitstream is corrupted. Various methods, such as checking the video bitstream syntax, monitoring the packet numbers of the received video data, etc., can be applied [2][23]. In this work, we assume that the errors are located and such information is available to us. We focus on the reconstruction for the lost video.

All error concealment methods reconstruct the lost video content by making use of some *a priori* knowledge about the video content. Most existing error concealment methods, which we refer to as *first-generation error concealment*, build such *a prior* in a heuristic manner by assuming smoothness or continuity of the pixel values, etc. The proposed *second-generation error concealment* methods train context-based models as the *a prior*. Methods of such a framework have advantages over first-generation error concealment, as the context-based model is created specifically for the video content hence can capture the statistical variations of the content more effectively.

It is important for a second-generation error concealment approach to choose a model that can represent the video content effectively. Principal component analysis (PCA) has long been used to model visual content of images. The most well known example is using eigenfaces to represent human faces [54]. In this work, we introduce two new adaptive models "adaptive mixture principal component analysis" (AMPCA, prior name "updating mixture of principal components" (UMPC)) [8][9] and "adaptive probabilistic principal component analysis" (APPCA) for second-generation error concealment. AMPCA and APPCA are very suitable for error concealment applications in that it updates with non-stationary video data.

A.1. Adaptive Mixture of Principal Component Analysis (AMPCA)

We consider a single component case of AMPCA in the following, named APCA. Interested readers can read [8][9] for AMPCA (prior name "updating mixture of principal components" (UMPC)), a more general case of APCA with the number of mixture components greater than one.

Given a set of data, we try to model the data with minimum representation error. The data given can be non-stationary, i.e., the stochastic properties of the data are time-varying as shown in Figure 85 (a). For example, at time instant n, the data are distributed as shown by Figure 85 (a). At time instant n', the data are distributed as shown by Figure 85 (b). We see that the mean of the data is shifting and the most representative axes of the data are also rotating.

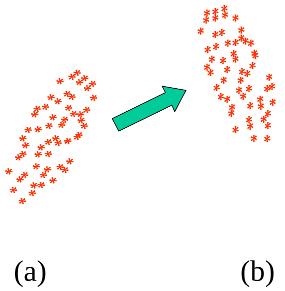
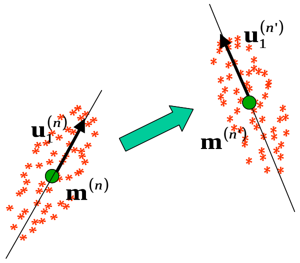


Figure 85. Non-stationary data at (a) time n (b) time n'

At any time instant, we attempt to represent the data as a weighted sum of the mean and principal axes. As time proceeds, the model changes its mean and principal axes as shown in Figure 86 from Figure 86 (a) to Figure 86 (b), so that it always models the current data effectively. To accomplish this, the representation/reconstruction error of the model evaluated at time instant n should have less contribution from the data that are further away in time from the current time instant n.



(a) (b)

Figure 86. APCA for non-stationary data at (a) time n (b) time n'

The optimization objective function at time instant n, that tries to minimize the sum of weighted reconstruction errors of all data, can be written as:

$$\min_{\mathbf{m}^{(n)},\mathbf{U}^{(n)}} \sum_{i=0}^{\infty} \alpha^i \left\| \mathbf{x}_{n-i} - \left[\underbrace{\mathbf{m}^{(n)} + \sum_{k=1}^{P} \left[\left(\mathbf{x}_{n-i} - \mathbf{m}^{(n)} \right)^T \mathbf{u}_k^{(n)} \right] \mathbf{u}_k^{(n)}}_{\hat{\mathbf{x}}_{n-i}} \right] \right\|^2 \tag{A.1}$$

The notations are organized as follows:

n : Current time index

D : Dimension of the data vector

P : Number of eigenvectors

$\mathbf{x}_{n-i}$: Data vector at time $n-i$, where i represents how far away the data are from the current time instant

$\mathbf{m}^{(n)}$: Mean at time n

$\mathbf{u}_k^{(n)}$: k th eigenvector at time n

$\mathbf{U}^{(n)}$: Matrix with P columns of $\mathbf{u}_k^{(n)}$, $k = 1 \sim P$

$\hat{\mathbf{x}}_{n-i}$: Reconstruction of $\mathbf{x}_{n-i}$

α : Decay factor, $0 < \alpha < 1$

The reconstruction errors contributed by previous data are weighted by powers of the decay factor α. The powers are determined by how far away this sample of data is from the current time instant. At any time instant n, we try to re-estimate or update the parameter (mean or eigenvector) given the parameter estimated at the previous time instant $n-1$ and the new data $\mathbf{x}_n$, by minimizing (A.1). The solution of mean $\mathbf{m}^{(n)}$ that minimizes (A.1) at time n is:

$$\mathbf{m}^{(n)} = \alpha\mathbf{m}^{(n-1)} + (1-\alpha)\mathbf{x}_n \tag{A.2}$$

We can see that $\mathbf{m}^{(n)}$ is obtained from the previous estimated $\mathbf{m}^{(n-1)}$ and the current input $\mathbf{x}_n$. The decay factor α tells how fast the new estimation $\mathbf{m}^{(n)}$ adapts to the new data $\mathbf{x}_n$. The smaller the decay factor, the faster the estimated $\mathbf{m}^{(n)}$ adapts to the new data. Similarly, the covariance matrix $\mathbf{C}^{(n)}$ that minimizes (A.1) at time n is:

$$\mathbf{C}^{(n)} = \alpha\mathbf{C}^{(n-1)} + (1-\alpha)\left[\left(\mathbf{x}_n - \mathbf{m}^{(n)}\right)\left(\mathbf{x}_n - \mathbf{m}^{(n)}\right)^T\right] \tag{A.3}$$

Again, $\mathbf{C}^{(n)}$ is obtained by the previous estimated $\mathbf{C}^{(n-1)}$ and the current input $\mathbf{x}_n$. The decay factor α controls how fast the eigenvectors adapt to the new data $\mathbf{x}_n$.

Experiment result of using APCA for error concealment is shown in Figure 87 and Figure 88. Figure 87 shows the updated APCA model at different time instances. Figure 88 shows the concealment result compared with the spatial interpolation method.

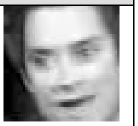
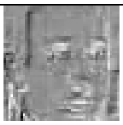
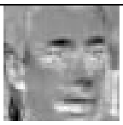
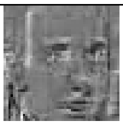
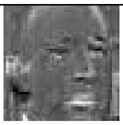
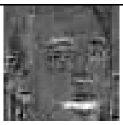
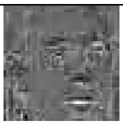
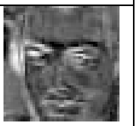
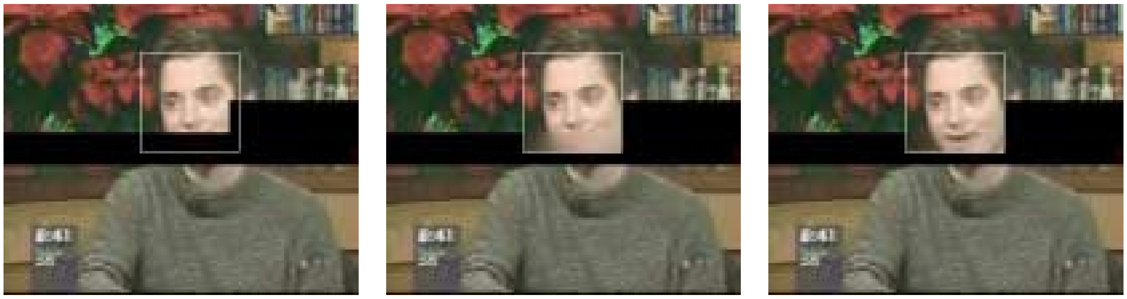
	Time 20	Time 22	Time 60
Mean			
1st Eigenvector			
2nd Eigenvector			
3rd Eigenvector			
4th Eigenvector			
5th Eigenvector			
6th Eigenvector			

Figure 87. Updated means and eigenvectors at time instants 20, 22, and 60



(a) (b) (c)

Figure 88. Sample reconstructed frames of Intra-coded "Interview" with: (a) no concealment; (b) concealment with spatial interpolation; or (c) concealment with APCA

A.2. Adaptive Probabilistic Principal Component Analysis (APPCA)

In the APCA/AMPCA approach, the APCA/AMPCA model is merely a subspace. There is no probability information associated with the model. We propose a new probabilistic-based and non- stationary model "adaptive probabilistic principal component analysis" (APPCA).

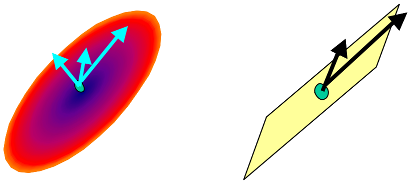


Figure 89. (a) Probabilistic PCA (PPCA) (b) PCA

Given a set of training data in real time, we try to describe them with a statistical model. The data can be non-stationary, that is, the statistical properties of the data are time-varying. With such non-stationary data, we want the statistical model to be trained based on the recent data more than the older data, so as to describe the recent data better. Such a statistical model adjusts its model parameters to adapt to the incoming data.

Let us represent the set of training data as $\mathbf{Y} = \left(\mathbf{y}_n \quad \mathbf{y}_{n-1} \quad \cdots \quad \mathbf{y}_{n-i} \quad \cdots\right)$, where each of the training data $\mathbf{y}_{n-i}$ is a d-dimensional vector. The indices n, $n-1$, …, $n-i$, etc., indicate the time, where n is the current time instant, $n-1$ is the previous time instant, and so on.

Before we proceed further, let us introduce 1) the weighted sample mean $\mathbf{m}^{(n)}$; and 2) the eigenanalysis of the weighted sample covariance matrix $\mathbf{S}^{(n)}$, of the training data $\mathbf{Y}$, at time instant n. These two results will be used later. The weighted sample mean $\mathbf{m}^{(n)}$ is defined as:

$$\mathbf{m}^{(n)} \equiv \frac{1}{1+\alpha+\alpha^2+\cdots}\sum_{i=0}^{\infty}\alpha^i \mathbf{y}_{n-i} \tag{A.4}$$

It can be expressed in a recursive form as:

$$\mathbf{m}^{(n)} = \alpha\mathbf{m}^{(n-1)} + (1-\alpha)\mathbf{y}_n \tag{A.5}$$

The weighted sample covariance matrix $\mathbf{S}^{(n)}$ is defined as:

$$\mathbf{S}^{(n)} \equiv \frac{1}{1+\alpha+\alpha^2+\cdots}\sum_{i=0}^{\infty}\alpha^i\left(\mathbf{y}_{n-i}-\mathbf{m}^{(n-i)}\right)\left(\mathbf{y}_{n-i}-\mathbf{m}^{(n-i)}\right)^T = (1-\alpha)\mathbf{Y}'\mathbf{Y}'^T \tag{A.6}$$

where $\mathbf{Y}'$ is defined as:

$$\mathbf{Y}' \equiv \left(\left(\mathbf{y}_n-\mathbf{m}^{(n)}\right) \quad \sqrt{\alpha}\left(\mathbf{y}_{n-1}-\mathbf{m}^{(n-1)}\right) \quad \cdots \quad \sqrt{\alpha^i}\left(\mathbf{y}_{n-i}-\mathbf{m}^{(n-i)}\right) \quad \cdots\right) \tag{A.7}$$

Similarly, $\mathbf{S}^{(n)}$ can be written in a recursive form as:

$$\mathbf{S}^{(n)} = \alpha\mathbf{S}^{(n-1)} + (1-\alpha)\left(\mathbf{y}_n-\mathbf{m}^{(n)}\right)\left(\mathbf{y}_n-\mathbf{m}^{(n)}\right)^T \tag{A.8}$$

The rank of $\mathbf{S}^{(n)}$ is $\operatorname{rank}\left(\mathbf{S}^{(n)}\right)=r$, where $r \le d$. The eigenanalysis result of $\mathbf{S}^{(n)}$ is:

$$\mathbf{S}^{(n)} = \mathbf{E}\mathbf{\Lambda}\ \mathbf{E}^T \text{, where}$$

$$\mathbf{E} = \left(\mathbf{e}_1 \quad \mathbf{e}_2 \quad \cdots \quad \mathbf{e}_r\right),\ \mathbf{e}_i^T\mathbf{e}_j = \delta_{ij} \text{, and}$$

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_r \end{pmatrix} = \operatorname{diag}(\lambda_1,\lambda_2,\cdots,\lambda_r) \tag{A.9}$$

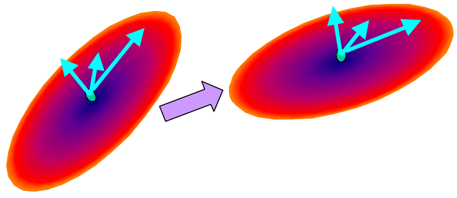
Now we are ready to introduce the statistical model and the adaptive estimators to obtain the model parameters. The Gaussian latent variable model is:

$$\mathbf{y} = \boldsymbol{\mu}^{(n)} + \mathbf{W}^{(n)}\mathbf{x} + \mathbf{e} \tag{A.10}$$

$$\mathbf{x} \sim N\left(\mathbf{0},\mathbf{I}_p\right)\big|_{\mathbf{x}} \tag{A.11}$$

$$\mathbf{e} \sim N\left(\mathbf{0},\mathbf{R}^{(n)}\right)\big|_{\mathbf{e}} = N\left(\mathbf{0},\varepsilon^{(n)}\mathbf{I}_d\right)\big|_{\mathbf{e}} \tag{A.12}$$

where $\mathbf{y}$ is the observed date, $\boldsymbol{\mu}^{(n)}$ is the mean of the data, $\mathbf{x}$ is the hidden variable, and $\mathbf{e}$ is the noise. To illustrate, the model is shown in Figure 90.



(a) (b)

Figure 90. PPCA at (a) time n (b) time n'

To solve for the model parameter, we propose an adaptive maximum likelihood (ML) estimator.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left[\sum_{i=0}^{\infty} \alpha^i \ln p\left(\mathbf{y}_{n-i} \middle| \boldsymbol{\theta}\right) \right] = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \qquad \text{(A.13)}$$

where we define $L(\boldsymbol{\theta}) \equiv \sum_{i=0}^{\infty} \alpha^i \ln p\left(\mathbf{y}_{n-i} \middle| \boldsymbol{\theta}\right)$.

To perform the adaptive ML estimation, let us first write out $p(\mathbf{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \mathbf{W}, \varepsilon\}$. $p(\mathbf{y}|\boldsymbol{\theta})$ can be derived from the pdf of $\mathbf{x}$ and $\mathbf{e}$, which are listed in (A.11) and (A.12).

$$p(\mathbf{y}|\mathbf{W},\varepsilon) = N\left(\mathbf{0}, \mathbf{W}\mathbf{W}^T + \mathbf{R}\right)\Big|_{\mathbf{y}} \qquad \text{(A.14)}$$

The sum of the weighted log-likelihoods function $L(\boldsymbol{\theta})$ can then be expressed as:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_{i=0}^{\infty} \alpha^i \ln p\left(\mathbf{y}_{n-i} \middle| \mathbf{W}, \varepsilon\right) \\ &= -\frac{1}{2(1-\alpha)} \begin{bmatrix} d\ln(2\pi) \\ +\ln\left|\mathbf{W}\mathbf{W}^T + \mathbf{R}\right| \end{bmatrix} - \frac{1}{2} \sum_{i=0}^{\infty} \alpha^i \begin{bmatrix} (\mathbf{y}_{n-i} - \boldsymbol{\mu})^T \left(\mathbf{W}\mathbf{W}^T + \mathbf{R}\right)^{-1} \\ (\mathbf{y}_{n-i} - \boldsymbol{\mu}) \end{bmatrix} \end{aligned} \qquad \text{(A.15)}$$

To find $\hat{\boldsymbol{\mu}}$, let us take the derivative of $L(\boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}$ and set the derivative to zero. $\hat{\boldsymbol{\mu}}$ is therefore,

$$\hat{\boldsymbol{\mu}} = (1-\alpha) \sum_{i=0}^{\infty} \alpha^i \mathbf{y}_{n-i} = \mathbf{m}^{(n)} \qquad \text{(A.16)}$$

With (A.16), we can express (A.15) as:

$$L(\boldsymbol{\theta}) = -\frac{1}{2(1-\alpha)} \left\{ d\ln(2\pi) + \ln\left|\mathbf{W}\mathbf{W}^T + \mathbf{R}\right| + tr\left[\left(\mathbf{W}\mathbf{W}^T + \mathbf{R}\right)^{-1} \mathbf{S}^{(n)}\right] \right\} \qquad \text{(A.17)}$$

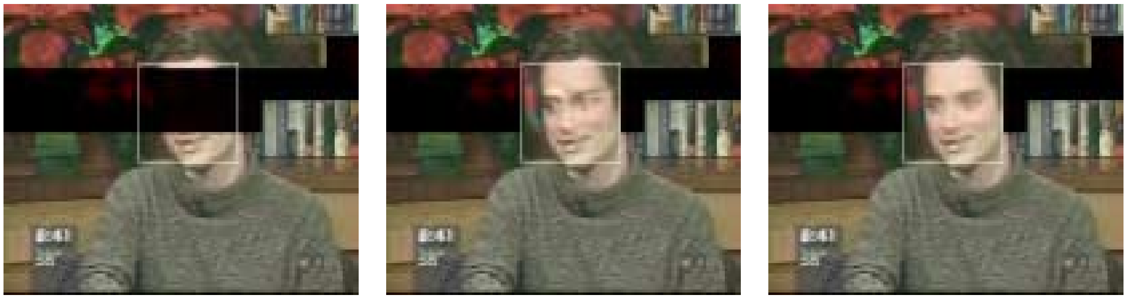
To find $\hat{\mathbf{W}}$, let us take the derivative of $L(\boldsymbol{\theta})$ with respect to $\mathbf{W}$ and set the derivative to zero. $\hat{\mathbf{W}}$ is:

$$\hat{\mathbf{W}} = \mathbf{E}_{(p)} \left(\boldsymbol{\Lambda}_{(p)} - \varepsilon \mathbf{I}_p\right)^{1/2} \qquad \text{(A.18)}$$

where $\mathbf{E}_{(p)}$ represents the eigenvectors of $\mathbf{S}^{(n)}$ up to the p th and $\boldsymbol{\Lambda}_{(p)}$ represents the eigenvalues of $\mathbf{S}^{(n)}$ up to the p th, where $p \le r$. To find $\hat{\varepsilon}$, let us take the derivative of $L(\boldsymbol{\theta})$ with respect to ε and set the derivative to zero. $\hat{\varepsilon}$ is:

$$\hat{\varepsilon} = \frac{1}{d-p}\sum_{p+1}^{r}\lambda_j \tag{A.19}$$

Experiment result of using APPCA for error concealment is shown in Figure 91. Error concealment result using APPCA compared with result using APCA is shown.



(a) (b) (c)

Figure 91. Sample reconstructed frames of Intra-coded “Interview” with: (a) no concealment; (b) concealment with APCA; or (c) concealment with APPCA

Appendix B. Finite-State Markov Model for Bit Error Simulation

For modeling the error characteristics of a wireless channel, a simple Gilbert-Elliot model [18][21] can be used. It considers a two-state Markov chain with “Good” and “Bad” states as shown in Figure 92. The “Good” state represents an error-free state with bit error rate (BER) $e_G = 0$, while the “Bad” state represents an erroneous state with BER e_B. Within one state the bit errors occur independently from each other.

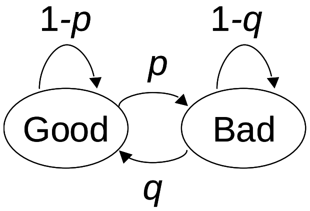


Figure 92. Two-state Markov chain for bit error simulation

B.1. K-State Markov Chain Model

A more sophisticated K-State Markov chain model [15][57] can be used for bit error simulations given:

- Raleigh fading, producing a time-varying receiver signal-to-noise ratio (SNR)
- BPSK coding
- Time variations of the received signal level are assumed to come from mobility (Doppler effect)

A homogeneous discrete-time Markov chain is then constructed where transitions occur only at every bit interval. Each state corresponds to a specific channel quality and has its own BER (with independent bit errors). The range of SNR is grouped into K intervals.

The Markov chain parameters are calculated as follows:

- Choose K as the number of states. State vector is $\mathbf{s} = (s_0, \quad s_1, \quad \cdots, \quad s_{K-1})^T$
    - The state transition matrix is $\mathbf{T} = \{t_{i,j}\},\ i, j \in \{0,1,\cdots,K-1\}$
    - The steady state probability is $\mathbf{p} = (p_0, \quad p_1, \quad \cdots, \quad p_{K-1})^T$
    - The vector of the error rates of all states is $\mathbf{e} = (e_0, \quad e_1, \quad \cdots, \quad e_{K-1})^T$
- Choose bit rate R_t
- The SNR are divided into K intervals $0 = A_0 < A_1 < \cdots < A_K = \infty$ and ρ sits in the mean of the interval $[A_k, A_{k+1})$. The probability distribution function (pdf) of the SNR is $p(a) = \frac{1}{\rho}\exp\left(-\frac{a}{\rho}\right)$
- The Doppler frequency is $f_m = \frac{v}{\lambda}$, where v is the speed of the mobile unit and λ is the wavelength
- For $k \in \{0,1,\cdots,K-1\}$,

$$p_k = \exp\left(-\frac{A_k}{\rho}\right) - \exp\left(-\frac{A_{k+1}}{\rho}\right) \tag{B.1}$$

  We also define $R_t^{(k)} = R_t p_k$ and $N_k = \sqrt{\frac{2\pi A_k}{\rho}} f_m \exp\left(-\frac{A_k}{\rho}\right)$

- The transition probability is:

$$\begin{cases} t_{i,j} = 0, & \text{for } |i-j| > 1 \\ t_{k,k+1} = \frac{N_{k+1}}{R_t^{(k)}}, & \text{for } k \in \{0,1,\cdots,K\text{-}2\} \\ t_{k,k-1} = \frac{N_k}{R_t^{(k)}}, & \text{for } k \in \{1,2,\cdots,K\text{-}1\} \\ t_{0,0} = 1 - t_{0,1}, & \\ t_{K-1,K-1} = 1 - t_{K-1,K-2}, & \\ t_{k,k} = 1 - t_{k,k-1} - t_{k,k+1}, & \text{for } k \in \{1,2,\cdots,K\text{-}2\} \end{cases} \tag{B.2}$$

- The error rate is:

$$e_k = \frac{\gamma_k - \gamma_{k+1}}{p_k} \tag{B.3}$$

where $\gamma_k = \exp\left(-\frac{A_k}{\rho}\right)Q\left(\sqrt{2A_k}\right) + \sqrt{\frac{\rho}{\rho+1}}\left[1 - Q\left(\sqrt{\frac{2A_k(\rho+1)}{\rho}}\right)\right]$

- The sojourn time is:

$$T_k = \frac{1}{1 - t_{k,k}} \tag{B.4}$$

- The mean BER is: $e_b = \mathbf{p}^T\mathbf{e}$

B.2. Simulation

In the simulations performed in the study, we use $K = 2$, $R_t = 2M$, $R_c = 2.4G$ ($\lambda = {}^{c}\!/\!_{R_c}$). Three different speeds 2 km/h, 6 km/h, 10 km/h and three different SNR 10 dB, 15 dB, and 20 dB are simulated. Thus, there are nine channel conditions simulated in total.

The variations in spend v provide bit errors with different burstiness because the larger the v is, the shorter the sojourn time is and thus the less bursty the bit errors are. The three different BER traces shown in Figure 93 are all with the same mean BER while the one with 10 km/h is more bursty than the others.

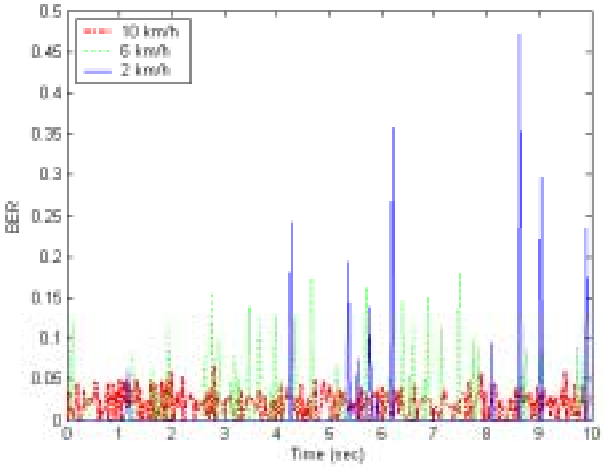


Figure 93. BER traces of wireless channel with units moving at different speeds

The variations in SNR ρ provide bit errors with different mean BER. The three different BER traces shown in Figure 94 are all with the same burstiness while the one with 10 dB is higher in BER than the others.

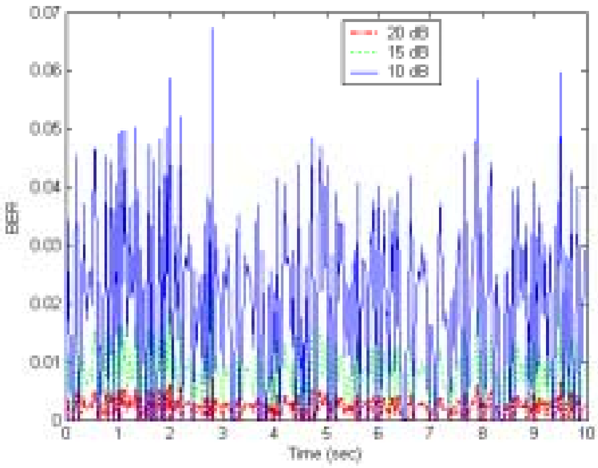


Figure 94. BER traces of wireless channel with different SNR

7. Bibliography

[1] H. Ahn, J.-K. Kim, S. Chong, B. Kim, and B. D. Choi, “A Video Traffic Model based on the Shifting-Level Process: the Effects of SRD and LRD on Queueing Behavior”, *INFOCOM 2000*, pp. 1036-1045.

[2] S. Aign, “Error concealment improvements for MPEG-2 using enhanced error detection and early re-synchronization”, *ICASSP 1997*, vol. 4, pp. 2625–2628.

[3] M. E. Al-Mualla, N. Canagarajah, D. R. Bull, “Multiple-reference temporal error concealment”, *ISCAS 2001*, vol. 5, pp. 149-152.

[4] P. Assuncao and M. Ghanbari, “Optimal Transcoding of Compressed Video”, *ICIP 1997*, vol. 1, pp. 739-742, Santa Barbara, CA, U.S.A., October 1997.

[5] M. Bystrom, V. Parthasarathy, and J. W. Modestino, “Hybrid Error Concealment Schemes for Broadcast Video Transmission over ATM Networks”, *IEEE Trans. on CSVT*, 9(6), September 1999, pp. 868-881.

[6] 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.

[7] T. P.-C. Chen and T. Chen, “Adaptive Joint Source-Channel Coding using Rate Shaping”, *ICASSP 2002*, Orlando, FL, May 2002.

[8] T. P.-C. Chen and T. Chen, “Updating Mixture of Principal Components for Error Concealment”, *ICIP 2002,* Rochester, NY, September 2003.

[9] T. P.-C. Chen and T. Chen, "Second-Generation Error Concealment for Video Transport over Error Prone Channels", *Wireless Communications and Mobile Computing*, Special Issue on Multimedia over Mobile IP, October 2002.

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

[11] P. A. Chou, T. Lookabaugh, and R. Gray, “Optimal Prunning with Applications to Tree-Structured Source Coding and Modeling”, *IEEE Trans. on Information Theory*, 35(2), March 1989, pp. 299-315.

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

[13] 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.

[14] A. M. Dawood and M. Ghanbari, “Content-Based MPEG Video Traffic Modeling”, *IEEE Trans. on Multimedia*, vol. 1, March 1999, pp. 77-87.

[15] 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.

[16] 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.

[17] A. Eleftheriadis and D. Anastassiou, “Constrained and General Dynamic Rate Shaping of Compressed Digital Video”, *ICIP 1995*, vol. 3, pp. 396-399, Washington D.C., October 1995.

[18] E. O. Elliot, “Estimates of Error Rates for Codes on Burst-Noise Channels”, *Bell Syst. Tech. J.*, 42: 1977-1997, September 1963.

[19] H. Everitt III, “Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources”, *Operation Research*, vol. 11, 1963, pp. 399-417.

[20] M. L. Fisher, “The Lagrangian Relaxation Method for Solving Integer Programming Problems”, *Management Science*, vol. 27, January 1981, pp. 1-18.

[21] E. N. Gilbert, “Capacity of a Burst-Noise Channel”, *Bell Syst. Tech.* J., 39:1253-1265, September 1960.

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

[23] S. S. Hemami, “Robust image transmission using resynchronizing variable-length codes and error concealment”, *IEEE Journal on Selected Areas in Communications*, 18(6), June 2000, pp. 927 –939.

[24] D. Heyman, A. Tabatabai, and T. V. Lakshman, “Statistical Analysis and Simulation Study of Video Conference Traffic in ATM Networks”, *IEEE Trans. on Circuits and Systems for Video Technology*, 2(1), March 1992.

[25] ITU-T Recommendation G. 114, “One-Way Transmission Time”, May 2000.

[26] ITU-T Recommendation G. 131, “Control of Talker Echo”, 1996.

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

[28] P. R. Jelenkovic, A. A. Lazar, and N. Semret, “The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queueing Behavior”, *IEEE Journal on Selected Areas in Communications*, 15(6), August 1997, pp. 1052-1071.

[29] 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.

[30] M. M. Krunz, A. M. Makowski, “Modeling Video Traffic using M/G/∞ Input Processes: A Compromise between Markovian and LRD Models”, *IEEE Journals on Selected Areas in Communications*, 16(5), 1998, pp. 733-748.

[31] W. M. Lam, A. Reibman, and B. Liu, “Recovery of Lost or Erroneously Received Motion Vectors”, *ICASSP 1993*, vol. 5, pp. 417-420.

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

[33] Q. Liang, J. M. Mendel, “MPEG VBR Video Traffic Modeling and Classification Using Fuzzy Techniques”, *IEEE Trans. on Fuzzy Systems*, 9(1), February 2001, pp. 183-193.

[34] S. Lin, D. J. Costello, and M. J. Miller, “Automatic Repeat Request Error Control Schemes”, *IEEE Communication Magazine*, 22(12), 1984, pp. 5-17.

[35] D. Liu, E. I. Sara, and Wei Sun, “Nested Auto-regressive Processes for MPEG-Encoded Video Traffic Modeling”, *IEEE Trans. on Circuits and Systems for Video Technology*, 11(2), February 2001, pp. 169-183.

[36] H. Liu, N. Ansari, and Y. Q. Shi, “The Tale of a Simple Accurate MPEG Video Traffic Molel”, *ICC 2001*, pp. 1042-1046.

[37] D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, “Method for Performance Evaluation of VBR Video Traffic Models”, *IEEE/ACM Transactions on Networking*, 2(2), April 1994, pp. 176-180.

[38] S. Ma and C. Ji, “Modeling Video Traffic Using Wavelets”, *ICC 98*, pp. 559-562.

[39] 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), June 2000, pp. 819-828.

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

[41] Y. Nakajima, H. Hori, and T. Kanoh, “Rate Conversion of MPEG Coded Video by Re-quantization Process”, *ICIP 1995*, vol. 3, pp. 408-411.

[42] G. L. Nemhauser and L. A. Wolsey, “Integer and Combinatorial Optimization”, New York: Wiley, 1988.

[43] A. Ortega and K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression”. *IEEE Signal Processing Magazine*, 15(6), November 1998, pp. 23-50.

[44] J. Ott, S. Wenger, S. Fukunaga, N. Sato, K. Yano, A. Miyazaki, K. Hata, R. Hakenberg, C. Burmeister, “Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)”, Internet Draft, draft-ietf-avt-rtcp-feedback-02.txt, March 2002.

[45] K. Ramchandran, A. Ortega, and M. Vetterli, “Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders”, *IEEE Transactions on Image Processing*, 3(5), September 1994, pp. 533-545.

[46] L. Rizzo, “Effective Erasure Codes for Reliable Computer Communication Protocols”, *Computer Communication Review*, April 1997.

[47] O. Rose, “Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems”, *Proceedings of 20th Conference on Local Computer Networks*, 1995, pp. 397-406.

[48] K.-D. Seo, S.-C. Heo, and J.-K. Kim, “Adaptive Rate Control Algorithm based on Logarithmic R-Q Model for MPEG-1 to MPEG-4 Transcoding”, *Signal Processing: Image Communication*, 17(10), November 2002, pp. 857-876.

[49] T. Shanableh and M. Ghanbari, “Heterogeneous Video Transcoding to Lower Spatial-Temporal Resolutions and Different Encoding Formats”, *IEEE Trans. on Multimedia*, 2(2), June 2000, pp. 101-110.

[50] Y. Shoham and A. Gersho, “Efficient Bit Allocation for an Arbitrary Set of Quantizers”, *IEEE Trans. on Acoustic, Speech, and Signal Processing*, 36(9), September 1988, pp. 1445-1453.

[51] T. Sikora, “MPEG Digital Video Coding Standards”, in Compressed Video over Networks, edited by M.-T. Sun and A. R. Reibman, Marcel Dekker, Inc., 2001.

[52] 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.

[53] D. S. Turaga and T. Chen, “Hierarchical Modeling of Variable Bit Rate Video Sources”, *Packet Video 2001*.

[54] M. Turk, A. Pentland, “Eigenfaces for Recognition”, *Journal of Cognitive Neuroscience*, 3(1), 1991, pp. 71-86.

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

[56] A. Vetro, and C. W. Chen, “Rate-Reduction Transcoding Design for Wireless Video Streaming”, *ICIP 2002*, vol. 1, pp. 29-32.

[57] H. S. Wang and N. Moayeri, “Finite-State Markov Channel-A Useful Model for Radio Communication Channels”, *IEEE Trans. On Vehicular Technology*, 44(1), February 1995.

[58] X. Wang, S. Jung, and J. S. Meditch, “VBR Broadcast Video Traffic Modeling- A Wavelet Decomposition Approach”, *GLOBECOM 97*, pp.1052-1056.

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

[60] O. Werner, “Requantization for Transcoding of MPEG-2 Intra-Frames”, *IEEE Trans. On Image Processing*, 8(2), February 1999, pp. 179-191.

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

[62] S. Xiong, J. Liang, Z. Yang, and Z. Lei, “Video Traffic Modeling Based on RBF Networks”, *Proceedings of 1998 International Conference on Communication Technology*, 1998.

[63] 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*, September 2002, pp. 737-740.

[64] W. Zeng and B. Liu, “Geometric-Structure-Based Directional Filtering for Error Concealment in Image/Video Transmission”, *SPIE Wireless Data Transmission at Information Systems/Photonics East’95*, October 1995, vol. 2601, pp. 145-156.

[65] 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, 1996.

[66] W. Zeng and B. Liu, “Geometric-Structure-Based Error Concealment with Novel Applications in Block-Based Low-Bit-Rate Coding”, *IEEE Trans. on Circuits and Systems for Video Technology*, 9(4), June 1999, pp. 648-665.

[67] Y. Zhang, X. Chen, B. Xiong, “A New VBR Model of MPEG Streams on ATM Networks”, *Proceedings of 1998 International Conference on Communication Technology*, 1998.