Adaptive QoS Control for Real-time Video Communication over Wireless Channels

Dapeng Wu* Y. Thomas Hou† Wenwu Zhu‡ Zhihai He§ Ya-Qin Zhang¶

Abstract

Robust transmission of real-time video over wireless channels is a challenging problem. This is because the bit error rate (BER) in wireless channels is very high and the bit errors could have devastating effects on the video presentation quality. This paper addresses this problem by introducing adaptive quality-of-service (QoS) control for real-time video communication over wireless channels. The adaptive QoS control consists of optimal mode selection and delay-constrained hybrid automatic repeat request (ARQ). Specifically, optimal mode selection provides QoS support (*i.e.*, error resilience) on the compression layer while delay-constrained hybrid ARQ provides QoS support (*i.e.*, bounded delay and reliability) on the link layer.

To provide QoS on the compression layer, we employ a novel approach to rate-distortion (R-D) optimized mode selection for real-time video communication over wireless channels. Different from the classical approach that only takes quantization distortion into account, our optimal mode selection also considers wireless channel characteristics and receiver behavior. To provide QoS on the link layer, we employ delay-constrained hybrid ARQ, which integrates rate-compatible punctured convolutional (RCPC) codes with delay-constrained retransmissions. Our delay-constrained hybrid ARQ has the advantage of providing reliability for the compression layer while guaranteeing delay bound and achieving high throughput.

By combining the best features of error-resilient source encoding, forward error correction (FEC) and delay-constrained retransmission, our proposed adaptive QoS control system is capable of achieving bounded delay, high reliability, and efficiency. Simulation results show that the proposed adaptive QoS control achieves better quality for real-time video under varying wireless channel conditions and utilizes network resources efficiently.

Key Words: Real-time video, wireless channel, optimal mode selection, delay-constrained hybrid ARQ.

*Please direct all correspondence to Prof. Dapeng Wu, University of Florida, Dept. of Electrical & Computer Engineering, Gainesville, FL 32611-6130, USA. Tel. (352) 392-4954, Fax (352) 392-0044, Email: wu@ece.ufl.edu. URL: http://www.wu.ece.ufl.edu.

†Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061-0111, USA. Tel. (540) 231-2950, Fax (540) 231-8292, Email: thou@vt.edu.

‡Microsoft Research, Asia, 5F, Beijing Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing 100080, China. Tel. (86-10) 6261-7711 Ext. 5405, Fax (86-10) 6255-5337, Email: wwzhu@microsoft.com.

§Department of Electrical & Computer Engineering, University of Missouri, Columbia, MO 65203, USA. Email: HeZhi@missouri.edu.

¶Microsoft Research, Asia, 5F, Beijing Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing 100080, China. Tel. (86-10) 6261-7711 Ext. 5790, Fax (86-10) 8809-7305, Email: yzhang@microsoft.com.

1 Introduction

With technological advances and the emergence of personal communication era, the demand for wireless video communication is increasing rapidly. Compared with wired links, wireless channels are typically much more noisy and have both small-scale and large-scale fades [13], making the bit error rate very high. The resulting bit errors could have devastating effects on the video presentation quality. Thus, robust transmission of real-time video over wireless channels is a critical problem.

To address the above problem, we introduce adaptive QoS control for video communication over wireless channels. The objective of our adaptive QoS control is to achieve good perceptual quality and utilize network resources efficiently through adaptation to time-varying wireless channels. Our adaptive QoS control consists of optimal mode selection and delay-constrained hybrid ARQ. Optimal mode selection provides QoS support (*i.e.*, error resilience) on the compression layer while delay-constrained hybrid ARQ provides QoS support (*i.e.*, bounded delay and reliability) on the link layer. The main contributions of this paper are: (1) an optimal mode selection algorithm which provides the best trade-off between compression efficiency and error resilience in rate-distortion (R-D) sense, and (2) delay-constrained hybrid ARQ which is capable of providing reliability for the compression layer while guaranteeing delay bound and achieving high throughput.

To facilitate the discussion of our adaptive QoS control, we first present an overall transport architecture for real-time video transmission over the wireless channel, which includes source rate adaptation, packetization, adaptive QoS control, and modulation. Since a wireless link is typically bandwidth-limited, it is essential for the source to adjust its transmission rate to the available bandwidth in the wireless channel. Therefore, source rate adaptation must be in place. Since bit-oriented syntax of a compressed video stream has to be converted into packets for transport over wireless channel, packetizing the compressed-video bit-stream is required. Adaptive QoS control is our primary weapon to combat bit errors under wireless environments. Finally, since modulation is required by any wireless communication, modulation is also an indispensable component of our architecture.

Under the above transport architecture, we discuss our adaptive QoS control. The first component is optimal mode selection. As we know, the effect of bit errors on the perceptual quality depends on the compression scheme used at the source, the wireless channel conditions and the error concealment scheme used at the receiver. High-compression coding algorithms usually employ inter-coding (*i.e.*, prediction) and variable length coding to achieve efficiency. Variable length coding is very susceptible to bit errors since a single bit error can corrupt a whole segment between two resynchronization markers in the compressed bit-stream [4], making the whole segment undecodable and resulting in discard of the video segment. For inter-coded video, discard of a video segment due to bit errors may degrade video quality over a large number of frames, until the next intra-coded frame is received. Intra-coding effectively stops error propagation, but at the cost of losing compression efficiency; inter-coding achieves compression efficiency, but at the risk of error propagation. Therefore, an optimal mode selection algorithm should be in place to achieve the best trade-off between efficiency and robustness.

The problem of mode selection has been well studied under the R-D framework [9, 15]. The previous approach to mode selection is to choose a mode that minimizes the quantization distortion between the original frame/macroblock and the reconstructed one under a given bit budget [9, 15], which is the so-called R-D optimized mode selection. We refer such R-D optimized mode selection as the classical approach. The classical approach is not able to achieve global optimality under the error-prone environment since it does not consider the channel characteristics and the receiver behavior. To remedy the drawback of the classical approach, we propose an end-to-end approach to R-D optimized mode selection. Different from the classical approach, our approach also considers the channel characteristics and the receiver behavior (*i.e.*, error concealment), in addition to quantization. Such an end-to-end approach was introduced in [19] for Internet video communication. In this paper, we show that such a methodology is also applicable to wireless environments with some modifications.

Previous work on optimal mode selection that considered wireless channel characteristics and error concealment has been reported in Ref. [1]. However, the distortion metrics introduced there are not accurate since the derivation of the distortion metrics was done at the block level. In addition, the wireless channel characteristics is dynamically changing, that is, the BER is varying from time to time. The scheme in Ref. [1] assumes that the BER is fixed and known *a priori*, which may not reflect the error behavior in the wireless channel. Thus, the scheme in Ref. [1] may not achieve optimality for dynamic wireless environments. This paper addresses these problems by deriving the distortion metrics at the pixel level and employing feedback mechanism to deal with the time-varying nature of the wireless channel. We will show how to formulate the globally R-D optimized mode selection and how to derive accurate global distortion metrics under our architecture. Along the same direction, Zhang et al. [22] also derived distortion metrics at the pixel level. Our distortion metrics are similar to theirs but are different.

Besides the QoS support on the compression layer provided by optimal mode selection, QoS support for video communication can be provided on the link layer as well.

The second component of our adaptive QoS control is delay-constrained hybrid ARQ, which is an error-control mechanism on the link layer. There are two kinds of error-control mechanisms on the link layer, namely, FEC and ARQ. The principle of FEC is to add redundant information so that original message can be recovered in the presence of bit errors. The use of FEC is primarily because throughput can be kept constant and delay can be bounded under FEC. However, the redundancy ratio (the ratio of redundant bit number to total bit number) should be made large enough to guarantee target QoS requirements under the worst channel conditions. In addition, FEC is not adaptive to varying wireless channel condition and it works best only when the BER is stable. If the number of bit errors exceeds the FEC code's recovery capability, the FEC code cannot recover any portion of the original data. In other words, FEC is useless when the short-term BER exceeds the recovery capability of the FEC code. On the other hand, when the wireless channel is in good state (*i.e.*, the BER is very small), using FEC will cause unnecessary overhead and waste bandwidth. Different from FEC, ARQ is adaptive to varying wireless channel condition. When the channel is in good state, no retransmissions are required and no bandwidth is wasted. Only when the channel condition becomes poor, the retransmissions will be used to recover the errors.

However, adaptiveness and efficiency of ARQ come with the cost of unbounded delay. That is, in the worst case, a packet may be retransmitted in unlimited times to recover bit errors.

To deal with the problems associated with FEC and ARQ, truncated type-II hybrid ARQ schemes have been proposed [8, 21]. Different from conventional type-II hybrid ARQ [3, 5, 7, 17], the truncated type-II hybrid ARQ has the restriction of maximum number of transmissions for a packet. Due to the maximum number of transmissions, delay can be bounded. The truncated type-II hybrid ARQ combines the good features of FEC and ARQ: bounded delay and adaptiveness. However, the maximum number of transmissions N_m is assumed to be fixed and known *a priori* [8, 21], which may not reflect the time-varying nature of delay. If N_m is set too large, retransmitted packets may arrive too late for play-out and thereby be discarded, resulting in waste of bandwidth; if N_m is set too small, the perceptual quality will be reduced due to unrecoverable errors that could have been corrected with more retransmissions. We address this problem by introducing delay-constrained hybrid ARQ. In our delay-constrained hybrid ARQ, the receiver makes retransmission requests in an intelligent way: when errors in the received packet are detected, the receiver decides whether to send a retransmission request according to the delay bound of the packet. Our delay-constrained hybrid ARQ is shown to be capable of achieving bounded delay, adaptiveness, and efficiency.

Our adaptive QoS control is aimed at providing good visual quality under time-varying wireless channels. By combining the best features of error-resilient source encoding, FEC and delay-constrained retransmission, the proposed adaptive QoS control is shown to be capable of achieving better quality for real-time video under varying wireless channel conditions while utilizing network resources efficiently.

The remainder of this paper is organized as follows. Section 2 sketches the overall transport architecture for real-time video communication over the wireless channel. In Section 3, we describe our optimal mode selection algorithm. Section 4 presents the delay-constrained hybrid ARQ. In Section 5, we use simulation results to demonstrate the effectiveness of our adaptive QoS control. Section 6 concludes this paper.

2 An Architecture for Real-time Video Communication over a Wireless Channel

In this section, we describe an architecture for real-time video communication over a wireless channel. We organize this section as follows. In Section 2.1, we overview our architecture. From Sections 2.2 to 2.4, we briefly describe key components in our architecture.

2.1 Overview

Figure 1 shows our architecture for one-to-one video communication over the wireless channel. On the sender side, raw bit-stream of live video is encoded by a rate-adaptive video encoder, which also employs optimal mode selection. After this stage, the compressed video bit-stream

is first packetized and then passed to CRC & RCPC encoder. Cyclic redundancy check (CRC) code and rate-compatible punctured convolutional (RCPC) codes are used as the error detection and correction codes. After modulation, packets are transmitted over a wireless channel. At the receiver, the video sequence is reconstructed in the reverse manner shown in Fig. 1.

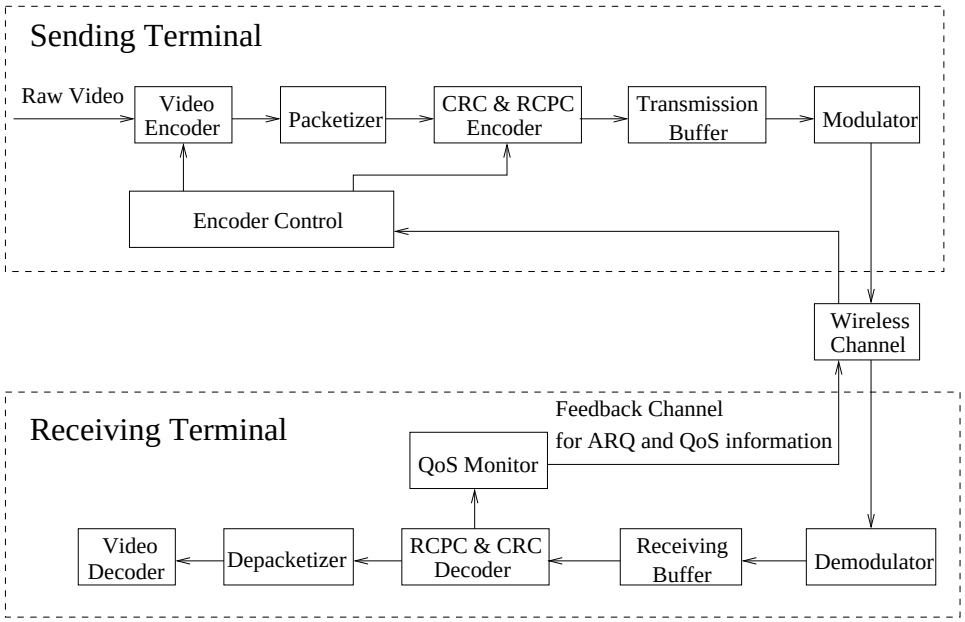

Figure 1: An architecture for MPEG-4 video over the wireless channel.

Under our architecture, a QoS monitor is kept at the receiver side to infer channel status based on the behavior of the arriving packets, *e.g.*, bit errors on the link layer, delay, and macroblock error ratio (MER). Link-layer bit errors and delay are used by hybrid ARQ to decide whether a retransmission should be requested. In addition, MER is periodically sent back to the source; the video encoder makes the optimal mode selection based on the returned MER. The feedback channel conveys two kinds of messages to the source: retransmission request and MER.

We use rate-adaptive video encoder [20] so that the rate of compressed video (denoted by $R_s(t)$) can match the available bandwidth of the wireless link (denoted by $R_a(t)$), which is time-varying due to probabilistic bit errors and retransmissions. In [20], we described how to adapt $R_s(t)$ to the target $R_a(t)$. Here, $R_a(t)$ is meant to be the maximum data rate that a video source can transmit at time t without error. It does not include the overheads due to ARQ and channel codes.

Next, we briefly describe three key components in our system: optimal mode selection, resynchronization markers, and delay-constrained hybrid ARQ.

2.2 Optimal Mode Selection

The problem of classical R-D optimized mode selection is to find the mode that minimizes the quantization distortion D_q for a given macroblock (MB), subject to a constraint. The classical

R-D optimized mode selection is optimal with respect to quantization distortion. However, under wireless environments, where packets may get discarded due to unrecoverable errors, the classical R-D optimized mode selection is not optimal with respect to the distortion D_r, which measures the difference between the original MB at the source and the reconstructed one at the receiver. This is because the classical R-D optimized mode selection does not consider the channel characteristics and receiver behavior, which also affect the distortion D_r.

In Section 3, we will present an optimal mode selection algorithm for MPEG-4 video over wireless channels. Our algorithm is capable of achieving best trade-off between error resilience and efficiency in an R-D sense.

2.3 Resynchronization Markers

Resynchronization markers are used to prevent error propagation in the bit-stream when errors occur [16]. Resynchronization markers are specially designed bit patterns that are usually placed at approximately regular intervals in the video bit stream. The function of these markers is to divide the compressed video bit stream into segments that are as independent of each other as possible. By searching for these markers, the decoder can reliably locate each segment without actually decoding the packet, and thereby prevent error propagation across different segments separated by markers. Each data segment of the bit stream should generally contain one or several complete logical entities of video information (*e.g.*, MB's) so that the decrease in coding efficiency (due to lack of exploiting dependencies between segments) can be minimized. The length of each segment is usually chosen to achieve a good trade-off between the overhead introduced by the markers, and reliability of the detection of markers when errors occur.

Our resynchronization scheme takes the video packet approach adopted by MPEG-4 [16]. Under the video packet approach, resynchronization markers are inserted to the bit-stream periodically. Specifically, if the number of bits after the current resynchronization marker exceeds a predetermined threshold, a new resynchronization marker is inserted at the start of the next MB.

After insertion of resynchronization markers, the bit-stream is passed to packetizer. We use a fixed packet size for packetization. After packetization, the video packet is passed to the link layer (CRC & RCPC encoder). For each video packet, the CRC & RCPC encoder generates a link-layer packet by using CRC/RCPC codes.

2.4 Delay-constrained Hybrid ARQ

To provide adaptive QoS control on the link layer, delay-constrained hybrid ARQ is performed in our architecture. Like other hybrid ARQ schemes [8, 21], our hybrid ARQ is also a combination of FEC and ARQ. In our hybrid ARQ, CRC and RCPC are employed as the FEC codes and RCPC offers variable rates to adapt to the time-varying wireless channel.

Our hybrid ARQ works as follows. When errors in the received link-layer packet is detected, the receiver decides whether to send a retransmission request according to the delay bound of the packet.

Table 1: Notations.

N_f	:	the total number of MBs in a frame
N_h	:	the highest location number of MBs in a frame ($N_h = N_f - 1$)
N_G	:	the number of MBs in a group of blocks (GOB)
F_i^n	:	the MB at location i in frame n
$\bar{F}_i^n$	:	the coded MB at location i in frame n.
$F_{\bar{i}}^n$	:	the MB (at location $\bar{i}$ in frame n) which is above F_i^n, if it exists.
$F_{\tilde{i}}^n$	:	the MB (at location $\tilde{i}$ in frame n) which is below F_i^n, if it exists.
$\mathcal{G}^n$	:	the set of macroblocks F_i^n ($i \in [0, N_h]$) that does not have $F_{\bar{i}}^n$ or does not have $F_{\tilde{i}}^n$.†
f_{ij}^n	:	the original value of pixel j in F_i^n (raw data).
$\tilde{f}_{ij}^n$	:	the value of reconstructed pixel j in F_i^n at the encoder.
$\hat{f}_{ij}^n$	:	the value of reconstructed pixel j in F_i^n at the receiver.
e_{ij}^n	:	the prediction error of pixel j in inter-coded F_i^n.
$\tilde{e}_{ij}^n$	:	the reconstructed prediction error of pixel j in inter-coded F_i^n.
$\tilde{f}_{uv}^{n-1}$	:	the value of reconstructed pixel v in F_u^{n-1} for prediction of f_{ij}^n.
$\hat{f}_{ml}^{n-1}$	:	the value of reconstructed pixel l in F_m^{n-1} to replace $\hat{f}_{ij}^n$ due to the error concealment.
$\mathcal{I}$	:	the set of coding modes (*i.e.*, $\mathcal{I} = \{\text{intra, inter}\}$).
M_i^n	:	the mode selected to code macroblock F_i^n ($M_i^n \in \mathcal{I}$).
λ	:	the Lagrange multiplier.

† For example, in QCIF, an MB in the first GOB does not have an MB above it; an MB in the last GOB does not have an MB below it.

If the delay requirement cannot be met, the request will not be sent. That is, the retransmission is truncated, instead of repeating retransmissions until a successful reception occurs. If the delay requirement can be met, a retransmission request will be sent to the source. Upon a retransmission request, the source only transmits the necessary incremental redundancy bits.

In Section 4, we will present our delay-constrained hybrid ARQ that is capable of providing reliability while achieving high throughput.

Table 1 lists the notations used in this paper. Under QCIF format, N_f is 99 and N_h is 98.

3 Optimal Mode Selection

As discussed in Section 2.2, the classical approach to mode selection is not able to achieve global optimality under the error-prone environment since it does not consider the channel conditions and the receiver behavior. In this section, we present an optimal mode selection algorithm.

We organize this section as follows. Section 3.1 discusses global distortion from an end-to-end perspective and set a stage for derivation of the global distortion metrics. In Sections 3.2, we derive

the global distortion metrics. In Section 3.3, we design an algorithm for optimal mode selection based on the global distortion metrics.

3.1 An End-to-End Perspective

An end-to-end approach to R-D optimized mode selection was introduced in Ref. [19] for Internet video communication. With the end-to-end approach, the distortion D_r is assumed to be the difference between the original MB at the source and the reconstructed one at the receiver. Under wireless environments, the distortion D_r is a random variable, which may take the value of either (1) the quantization distortion D_q plus the distortion D_{ep} caused by error propagation, or (2) distortion D_c caused by errors due to error concealment. We define the *global* distortion D as the expectation of the random variable D_r. That is,

$$D = E\{D_r\}, \tag{1}$$

where D_r takes the value of $(D_q + D_{ep})$ or D_c with certain probability, which is determined by channel error characteristics.

Then, the problem of globally R-D optimized mode selection is to find the mode that minimizes the global distortion D for a given MB, subject to a constraint R_c on the number of bits used. For the MB at location i in frame n (denoted by F_i^n), this constrained problem reads as follows:

$$\min_{M_i^n} D(M_i^n) \quad \text{subject to} \quad R(M_i^n) \leq R_c, \tag{2}$$

where $D(M_i^n)$ and $R(M_i^n)$ denote the *global* distortion and bit budget for macroblock F_i^n with a particular mode M_i^n, respectively.

In terms of mean squared error (MSE), we define the global distortion metric for macroblock F_i^n as follows:

$$\text{MSE}(F_i^n) = \frac{E\left\{\sum_{j=1}^{256}(f_{ij}^n - \hat{f}_{ij}^n)^2\right\}}{256} \tag{3}$$

where f_{ij}^n is the original value of pixel j in F_i^n (raw data), and $\hat{f}_{ij}^n$ is the value of reconstructed pixel j in F_i^n at the receiver.

The global distortion is affected by three factors: source behavior, channel characteristics, and receiver behavior, which are described in Sections 3.1.1 to 3.1.3, respectively.

3.1.1 Source Behavior

The source behavior, *i.e.*, quantization, have impact on global distortion. This is because quantization error can cause image distortion.

A block diagram of the video encoder is depicted in Fig. 2. The switches represent the two different paths for the intra- and inter-mode.

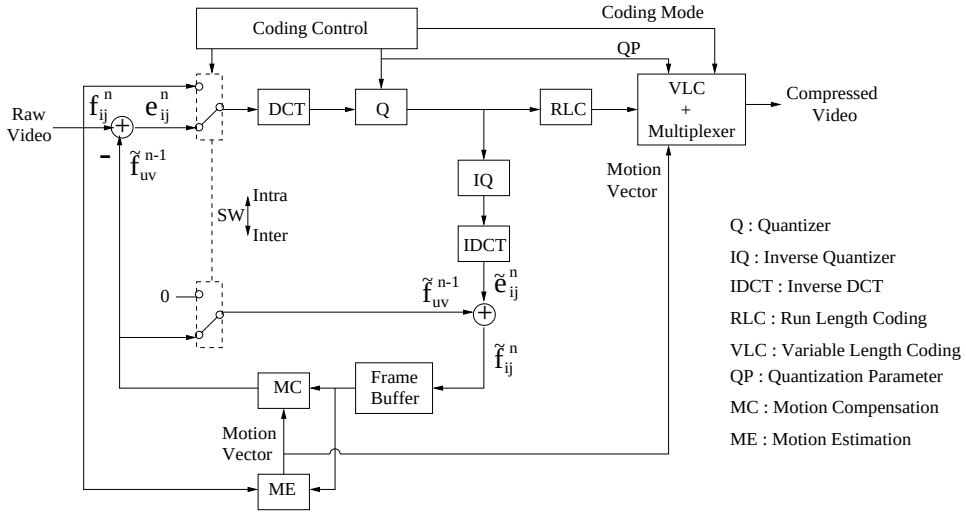

Figure 2: Block diagram of the video encoder working under the inter mode.

Under the intra mode, the raw video f_{ij}^n is transformed by discrete cosine transform (DCT), quantized and coded by run length coding (RLC). The resulting information, as well as coding parameters such as the coding mode and the quantization parameter (QP), is coded by variable length coding (VLC). Then the compressed video stream is formed by the multiplexer. At the same time, the pixel is reconstructed at the encoder for the prediction used by the next frame. The value of the reconstructed pixel is $\tilde{f}_{ij}^n$.

Figure 2 also illustrates the case when the encoder is operating under the inter-mode. Under such mode, the raw video f_{ij}^n is first predicted from the motion-compensated pixel in the previous frame, $\tilde{f}_{uv}^{n-1}$, resulting in the prediction error, e_{ij}^n. The prediction error, e_{ij}^n, is DCT-transformed, quantized and coded by RLC. The resulting information, as well as coding parameters such as the coding mode, the motion vector and the QP, is coded by VLC. Then the compressed video stream is formed by the multiplexer. At the same time, the pixel is reconstructed at the encoder for the prediction used by the next frame. There are two steps for the reconstruction at the encoder. First, the prediction error is reconstructed, resulting in $\tilde{e}_{ij}^n$. Second, $\tilde{e}_{ij}^n$ is added to the predicted value $\tilde{f}_{uv}^{n-1}$, resulting in the reconstructed value, $\tilde{f}_{ij}^n$. That is, the pixel is reconstructed by $\tilde{f}_{ij}^n = \tilde{e}_{ij}^n + \tilde{f}_{uv}^{n-1}$.

3.1.2 Wireless Channel Characteristics

Wireless channel characteristics (*i.e.*, error characteristics) has great impact on the global distortion. Since only MB error ratio is relevant in deriving our global distortion metrics, we are primarily concerned with compression-layer error characteristics rather than link-layer error characteristics.

Under our architecture, delay-constrained hybrid ARQ is used to recover link-layer bit errors. If the bit errors in a link-layer packet cannot be completely recovered before its delay bound, the corrupted packet will be discarded. Then the MPEG-4 decoder uses error concealment to conceal

the MB's associated with the corrupted link-layer packet. We will address error concealment in Section 3.1.3.

There are two ways to obtain MER, namely, analysis and measurement. We can use analysis like Ref. [21] to derive MER. But the analysis is very complicated and we can only get an approximation of MER (*e.g.*, an upper bound). The approximation is typically only applicable to a certain channel model and a certain modulation method. For example, the upper bound on packet error ratio in Ref. [21] is only applicable to Rayleigh fading channel model and $\pi/4$-DQPSK modulation with coherent detection.

Due to the complexity and inflexibility of the analytical method, we choose a measurement approach in our system. That is, we measure the numbers of erroneous/discarded MB's and correct MB's at the receiver. That is, MER, denoted by p_d, is given by

$$MER = p_d = \frac{N_d}{N_d + N_c}, \tag{4}$$

where N_d is the number of erroneous/discarded MB's and N_c the number of correct MB's.

3.1.3 Receiver Behavior

Receiver behavior, *i.e.*, error concealment, also affects the global distortion. In our system, we only consider such an error concealment scheme: the corrupted MB is replaced with the MB from the previous frame pointed by a motion vector. The motion vector of the corrupted MB is copied from one of its neighboring MB (which is above or below the corrupted MB) when available, otherwise the motion vector is set to zero. Specifically, if the MB above the corrupted MB is received correctly, the motion vector of the corrupted MB is copied from the MB above it; if the MB above the corrupted MB is also damaged and the MB below the corrupted MB is received correctly, the motion vector of the corrupted MB is copied from the MB below it; otherwise the motion vector is set to zero.

Figure 3 shows a block diagram of the video decoder, in which three switches represent different scenarios as follows.

- *Switch SW1:* represents the two different paths for the intra- and inter-mode.
- *Switch SW2:* represents the two different paths for the two cases as follows.
  - the case where the coded MB at location i in frame n (denoted by $\bar{F}_i^n$) is received correctly.
  - the case where $\bar{F}_i^n$ is corrupted.
- *Switch SW3:* represents the two different paths for the two cases as follows.
  - the case where the estimated motion vector of the corrupted F_i^n is set to zero.
  - the case where the estimated motion vector of the corrupted F_i^n is copied from its neighboring macroblock $F_{\bar{i}}^n$ or $F_{\tilde{i}}^n$, where $F_{\bar{i}}^n$ denotes the MB (at location $\bar{i}$ in frame n) which is above F_i^n and $F_{\tilde{i}}^n$ denotes the MB (at location $\tilde{i}$ in frame n) which is below F_i^n.

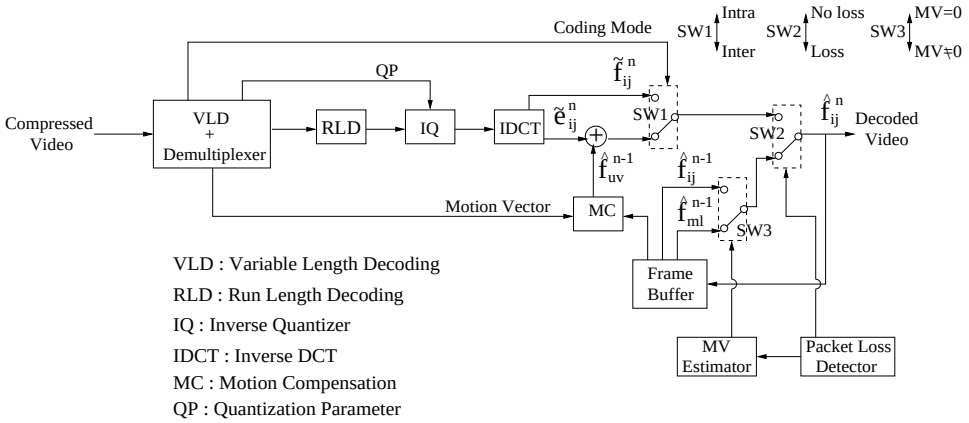

Figure 3: Block diagram of the video decoder.

The compressed video stream is first demultiplexed by the demultiplexer and decoded by variable length decoding (VLD). The resulting coding mode information will control switch SW1. The resulting QP will control the inverse quantizer (IQ). The resulting motion vector is used for motion compensation. The information containing DCT coefficient will be decoded by run length decoding (RLD), then inversely quantized, and inversely DCT-transformed.

There are three cases for the reconstructed pixel at the receiver as follows.

- *Case (i):* $\bar{F}_i^n$ is received correctly. If F_i^n is intra-coded, then we have $\hat{f}_{ij}^n = \tilde{f}_{ij}^n$, which is illustrated in Fig. 3 with state {SW1: Intra, SW2: No loss, SW3: don't care}. If F_i^n is inter-coded, then we have
$$\hat{f}_{ij}^n = \tilde{e}_{ij}^n + \hat{f}_{uv}^{n-1}, \tag{5}$$
which is illustrated in Fig. 3 with state {SW1: Inter, SW2: No loss, SW3: don't care}.

- *Case (ii):* $\bar{F}_i^n$ is corrupted and the neighboring MB ($\bar{F}_{\tilde{i}}^n$ or $\bar{F}_{\tilde{i}}^n$) has been received correctly. Then we have $\hat{f}_{ij}^n = \hat{f}_{ml}^{n-1}$, which is illustrated in Fig. 3 with state {SW1: don't care, SW2: Loss, SW3: MV≠0}.

- *Case (iii):* $\bar{F}_i^n$, $\bar{F}_{\tilde{i}}^n$, and $\bar{F}_{\tilde{i}}^n$ are all corrupted. Then we have $\hat{f}_{ij}^n = \hat{f}_{ij}^{n-1}$, which is illustrated in Fig. 3 with state {SW1: don't care, SW2: Loss, SW3: MV=0}.

So far we have examined three key components that affect the global distortion. In the next section, we derive the global distortion metrics based on the three factors discussed in this section.

3.2 Global Distortion Metrics

This section derives the global distortion metrics, which will be used by our optimal mode selection in Section 3.3.

Without loss of generality, we consider the distortion for macroblock s in frame N, where s ($s \in [0, N_h]$) is the location number and N ($N \geq 0$) is the frame number. Note that the sequence number for both frame and packet start from zero. Denote $\mathcal{G}^n$ the set of macroblocks F_i^n that does not have $F_{\bar{i}}^n$ or does not have $F_{\tilde{i}}^n$. For example, in QCIF, an MB in the first GOB does not have an MB above it; an MB in the last GOB does not have an MB below it.

The following proposition shows how to compute MSE for the intra-coded MB under our architecture.

Proposition 1 *Assume the packet loss at the compression layer follows Bernoulli distribution with probability p_d. Under our architecture, the MSE for the intra-coded MB at location s of frame N ($N > 0$) is given by*

$$MSE(F_s^N, intra) = \frac{\sum_{j=1}^{256} \left(\left(f_{sj}^N \right)^2 - 2 \times f_{sj}^N \times E\{\hat{f}_{sj}^N\} + E\left\{ \left(\hat{f}_{sj}^N \right)^2 \right\} \right)}{256}, \tag{6}$$

where

$$E\{\hat{f}_{ij}^n\} = \begin{cases} (1-p_d) \times \tilde{f}_{ij}^n + p_d \times (1-p_d) \times E\{\hat{f}_{ml}^{n-1}\} + p_d^2 \times E\{\hat{f}_{ij}^{n-1}\} & \text{if } F_i^n \in \mathcal{G}^n \\ (1-p_d) \times \tilde{f}_{ij}^n + p_d \times (1-p_d^2) \times E\{\hat{f}_{ml}^{n-1}\} + p_d^3 \times E\{\hat{f}_{ij}^{n-1}\} & \text{if } F_i^n \notin \mathcal{G}^n \end{cases}, \tag{7}$$

$$E\left\{ \left(\hat{f}_{ij}^n \right)^2 \right\} = \begin{cases} (1-p_d) \times \left(\tilde{f}_{ij}^n \right)^2 + p_d \times (1-p_d) \times E\left\{ \left(\hat{f}_{ml}^{n-1} \right)^2 \right\} + p_d^2 \times E\left\{ \left(\hat{f}_{ij}^{n-1} \right)^2 \right\} & \text{if } F_i^n \in \mathcal{G}^n \\ (1-p_d) \times \left(\tilde{f}_{ij}^n \right)^2 + p_d \times (1-p_d^2) \times E\left\{ \left(\hat{f}_{ml}^{n-1} \right)^2 \right\} + p_d^3 \times E\left\{ \left(\hat{f}_{ij}^{n-1} \right)^2 \right\} & \text{if } F_i^n \notin \mathcal{G}^n \end{cases}, \tag{8}$$

and $E\{\hat{f}_{ij}^0\} = \tilde{f}_{ij}^0$ *and* $E\left\{ \left(\hat{f}_{ij}^0 \right)^2 \right\} = \left(\tilde{f}_{ij}^0 \right)^2$.

Proof. From Eq. (3), one can easily obtain Eq. (6). Note that due to the random nature of the channel characteristics, $\hat{f}_{sj}^N$ at the receiver is a random variable while f_{sj}^N is not a random variable.

To see that Eq. (7) holds, we consider the following two cases based on the position of F_i^n.

- *Case 1:* Suppose that F_i^n only has $F_{\bar{i}}^n$ or $F_{\tilde{i}}^n$ (*e.g.*, F_i^n is located at the top/bottom of the frame), *i.e.*, $F_i^n \in \mathcal{G}^n$. According to the packet loss behavior, there are three cases for the random variable $\hat{f}_{ij}^n$ as follows.
  - *Subcase 1.1:* Suppose that $\bar{F}_i^n$ is received correctly. Then we have $\hat{f}_{ij}^n = \tilde{f}_{ij}^n$, and the probability of this event is $1 - p_d$.

– *Subcase 1.2:* For simplicity, we only show the situation where F_i^n is located at the top of the frame. The situation where F_i^n is located at the bottom of the frame is the same. Suppose that $\bar{F}_i^n$ is corrupted and the MB below (*i.e.*, $\bar{F}_{\tilde{i}}^n$) has been received correctly. Then due to our error concealment scheme, the corrupted pixel j in F_i^n is concealed by the pixel in frame $n-1$, which is pointed by the motion vector of $\bar{F}_{\tilde{i}}^n$. Since this pixel is denoted by pixel l in macroblock m of frame $n-1$ (see Table 1), then we have $\hat{f}_{ij}^n = \hat{f}_{ml}^{n-1}$, and the probability of this event is $p_d \cdot (1-p_d)$. Since $\hat{f}_{ml}^{n-1}$ is also a random variable, from Fact 1 (see Appendix), the expectation of $\hat{f}_{ml}^{n-1}$ should be chosen in computing the expectation of $\hat{f}_{ij}^n$. So we have $\hat{f}_{ij}^n = E\{\hat{f}_{ml}^{n-1}\}$.

– *Subcase 1.3:* For simplicity, we only show the situation where F_i^n is located at the top of the frame. The situation where F_i^n is located at the bottom of the frame is the same. Suppose that $\bar{F}_i^n$ and $\bar{F}_{\tilde{i}}^n$ are all corrupted. Since $\bar{F}_{\tilde{i}}^n$ is corrupted, the estimated motion vector of the corrupted F_i^n is set to zero. Then the corrupted pixel j in F_i^n is concealed by pixel j in F_i^{n-1}. Thus, we have $\hat{f}_{ij}^n = \hat{f}_{ij}^{n-1}$, and the probability of this event is p_d^2. Since $\hat{f}_{ij}^{n-1}$ is also a random variable, from Fact 1, the expectation of $\hat{f}_{ij}^{n-1}$ should be chosen in computing the expectation of $\hat{f}_{ij}^n$. So we have $\hat{f}_{ij}^n = E\{\hat{f}_{ij}^{n-1}\}$.

Based on the analysis of Subcases 1.1 to 1.3, the expectation of $\hat{f}_{ij}^n$ can be given by

$$E\{\hat{f}_{ij}^n\} = (1-p_d) \cdot \tilde{f}_{ij}^n + p_d \cdot (1-p_d) \cdot E\{\hat{f}_{ml}^{n-1}\} + p_d^2 \cdot E\{\hat{f}_{ij}^{n-1}\}. \tag{9}$$

- *Case 2:* Suppose that F_i^n has both $F_{\bar{i}}^n$ and $F_{\tilde{i}}^n$ (*e.g.*, F_i^n is not located at the top/bottom of the frame), *i.e.*, $F_i^n \notin \mathcal{G}^n$.

  According to the packet loss behavior, there are three cases for the random variable $\hat{f}_{ij}^n$ as follows.

  – *Subcase 2.1:* Suppose that $\bar{F}_i^n$ is received correctly. Then we have $\hat{f}_{ij}^n = \tilde{f}_{ij}^n$, and the probability of this event is $1-p_d$.

  – *Subcase 2.2:* Suppose that $\bar{F}_i^n$ is corrupted and the MB above and/or below (*i.e.*, $\bar{F}_{\bar{i}}^n$ and/or $\bar{F}_{\tilde{i}}^n$) have/has been received correctly. Then due to our error concealment scheme, the corrupted pixel j in F_i^n is concealed by the pixel in frame $n-1$, which is pointed by the motion vector of $\bar{F}_{\bar{i}}^n$ or $\bar{F}_{\tilde{i}}^n$. Since this pixel is denoted by pixel l in macroblock m of frame $n-1$, then we have $\hat{f}_{ij}^n = \hat{f}_{ml}^{n-1}$, and the probability of this event is $p_d \cdot (1-p_d^2)$. Since $\hat{f}_{ml}^{n-1}$ is also a random variable, from Fact 1, the expectation of $\hat{f}_{ml}^{n-1}$ should be chosen in computing the expectation of $\hat{f}_{ij}^n$. So we have $\hat{f}_{ij}^n = E\{\hat{f}_{ml}^{n-1}\}$.

  – *Subcase 2.3:* Suppose that $\bar{F}_i^n$, $\bar{F}_{\bar{i}}^n$, and $\bar{F}_{\tilde{i}}^n$ are all corrupted. Since the MBs above and below F_i^n are corrupted, the estimated motion vector of the corrupted F_i^n is set to

zero. Then the corrupted pixel j in F_i^n is concealed by pixel j in F_i^{n-1}. Thus, we have $\hat{f}_{ij}^n = \hat{f}_{ij}^{n-1}$, and the probability of this event is p_d^3. Since $\hat{f}_{ij}^{n-1}$ is also a random variable, from Fact 1, the expectation of $\hat{f}_{ij}^{n-1}$ should be chosen in computing the expectation of $\hat{f}_{ij}^n$. So we have $\hat{f}_{ij}^n = E\{\hat{f}_{ij}^{n-1}\}$.

Based on the analysis of Subcases 2.1 to 2.3, the expectation of $\hat{f}_{ij}^n$ can be given by

$$E[\hat{f}_{ij}^n] = (1-p_d)\cdot \tilde{f}_{ij}^n + p_d \cdot (1-p_d^2)\cdot E\{\hat{f}_{ml}^{n-1}\} + p_d^3 \cdot E\{\hat{f}_{ij}^{n-1}\} \tag{10}$$

Combining Cases 1 and 2 above, we complete the proof of Eq. (7).

The proof for Eq. (8) is similar to that for Eq. (7). ■

The following proposition shows how to compute MSE for the inter-coded MB under our architecture.

Proposition 2 *Assume the packet loss at the compression layer follows Bernoulli distribution with probability p_d. Under our architecture, the MSE for the inter-coded MB at location s of frame N ($N > 0$) is given by*

$$MSE(F_s^N, inter) = \frac{\sum_{j=1}^{256}\left(\left(f_{sj}^N\right)^2 - 2\times f_{sj}^N \times E\{\hat{f}_{sj}^N\} + E\left\{\left(\hat{f}_{sj}^N\right)^2\right\}\right)}{256}, \tag{11}$$

where

$$E\{\hat{f}_{ij}^n\} = \begin{cases} (1-p_d)\times(\tilde{e}_{ij}^n + E\{\hat{f}_{uv}^{n-1}\}) + p_d \times (1-p_d)\times E\{\hat{f}_{ml}^{n-1}\} + p_d^2 \times E\{\hat{f}_{ij}^{n-1}\} & \text{if } F_i^n \in \mathcal{G}^n \\ (1-p_d)\times(\tilde{e}_{ij}^n + E\{\hat{f}_{uv}^{n-1}\}) + p_d \times (1-p_d^2)\times E\{\hat{f}_{ml}^{n-1}\} + p_d^3 \times E\{\hat{f}_{ij}^{n-1}\} & \text{if } F_i^n \notin \mathcal{G}^n \end{cases}, \tag{12}$$

$$E\left\{\left(\hat{f}_{ij}^n\right)^2\right\} = \begin{cases} (1-p_d)\times\left(\left(\tilde{e}_{ij}^n\right)^2 + 2\times\tilde{e}_{ij}^n\times E\{\hat{f}_{uv}^{n-1}\} + E\left\{\left(\hat{f}_{uv}^{n-1}\right)^2\right\}\right) + \\ p_d\times(1-p_d)\times E\left\{\left(\hat{f}_{ml}^{n-1}\right)^2\right\} + p_d^2\times E\left\{\left(\hat{f}_{ij}^{n-1}\right)^2\right\} & \text{if } F_i^n \in \mathcal{G}^n \\ (1-p_d)\times\left(\left(\tilde{e}_{ij}^n\right)^2 + 2\times\tilde{e}_{ij}^n\times E\{\hat{f}_{uv}^{n-1}\} + E\left\{\left(\hat{f}_{uv}^{n-1}\right)^2\right\}\right) + \\ p_d\times(1-p_d^2)\times E\left\{\left(\hat{f}_{ml}^{n-1}\right)^2\right\} + p_d^3\times E\left\{\left(\hat{f}_{ij}^{n-1}\right)^2\right\} & \text{if } F_i^n \notin \mathcal{G}^n \end{cases}, \tag{13}$$

and $E\{\hat{f}_{ij}^0\} = \tilde{f}_{ij}^0$ *and* $E\left\{\left(\hat{f}_{ij}^0\right)^2\right\} = \left(\tilde{f}_{ij}^0\right)^2$.

Proof. Similar to the derivation of Eq. (6), we have Eq. (11). To see that Eqs. (12) and (13) hold, we can also use the same way as that to derive Eqs. (7) and (8). The only difference is the case where $\bar{F}_i^n$ is received correctly. In this case, under the inter code, we have $\hat{f}_{ij}^n = \tilde{e}_{ij}^n + \hat{f}_{uv}^{n-1}$, which is Eq. (5). ■

3.3 Optimal Mode Selection

Given the channel characteristics (obtained from measurement described in Section 3.1.2), we can design a globally R-D optimized mode selection algorithm for our architecture.

Consider a group of blocks (GOB) or slice denoted by $\mathcal{F}_g^n = (F_g^n, \cdots, F_{g+N_G-1}^n)$, where N_G is the number of MBs in a GOB/slice. Assume each MB in $\mathcal{F}_g^n$ can be coded using only one of the two modes in set $\mathcal{I}$. Then for a given GOB/slice, the modes assigned to the MBs in $\mathcal{F}_g^n$ are given by the N_G-tuple, $\mathcal{M}_g^n = (M_g^n, \cdots, M_{g+N_G-1}^n) \in \mathcal{I}^{N_G}$. The problem of globally R-D optimized mode selection is to find the combination of modes that minimizes the distortion for a given GOB/slice, subject to a constraint R_c on the number of bits used. This constrained problem can be formulated as

$$\min_{\mathcal{M}_g^n} D(\mathcal{F}_g^n, \mathcal{M}_g^n) \quad \text{subject to} \quad R(\mathcal{F}_g^n, \mathcal{M}_g^n) \leq R_c, \tag{14}$$

where $D(\mathcal{F}_g^n, \mathcal{M}_g^n)$ and $R(\mathcal{F}_g^n, \mathcal{M}_g^n)$ denote the total distortion and bit budget, respectively, for the GOB/slice $\mathcal{F}_g^n$ with a particular mode combination $\mathcal{M}_g^n$.

The constrained minimization problem in (14) can be converted to an unconstrained minimization problem by Lagrange multiplier technique as follows:

$$\sum_{i=g}^{g+N_G-1} \min_{M_i^n} J(F_i^n, M_i^n) = \sum_{i=g}^{g+N_G-1} \min_{M_i^n} \{D(F_i^n, M_i^n) + \lambda R(F_i^n, M_i^n)\}, \tag{15}$$

where the global distortion $D(F_i^n, M_i^n)$ can be expressed by the formulae we derived in Section 3.2, according to the coding mode.

The problem of (15) is a standard R-D optimization problem and can be solved by the approaches described in [6, 10, 12, 18]. Different from these approaches, we use a simpler method to obtain λ.

Since a large λ in the optimization problem of (15) can reduce the bit-count of the coded frame, we employ this nature in choosing λ. To be specific, at the end of frame n, we adjust λ for frame $n+1$ (*i.e.*, λ_{n+1}) as follows:

$$\lambda_{n+1} = \frac{2 \cdot B_n + (\beta - B_n)}{B_n + 2 \cdot (\beta - B_n)} \cdot \lambda_n \tag{16}$$

where B_n is the current buffer occupancy at the end of frame n and β is the buffer size. λ_n is initialized by a preset value λ_0. The adjustment in Eq. (16) is to keep the buffer occupancy at the middle level to reduce the chance of buffer overflow or underflow. In other word, Eq. (16) also achieves the objective of rate control.

4 Hybrid ARQ

As mentioned in Section 1, FEC schemes offer a constant throughput but their reliability degrades as the channel condition becomes poor. ARQ achieves high efficiency and high throughput under

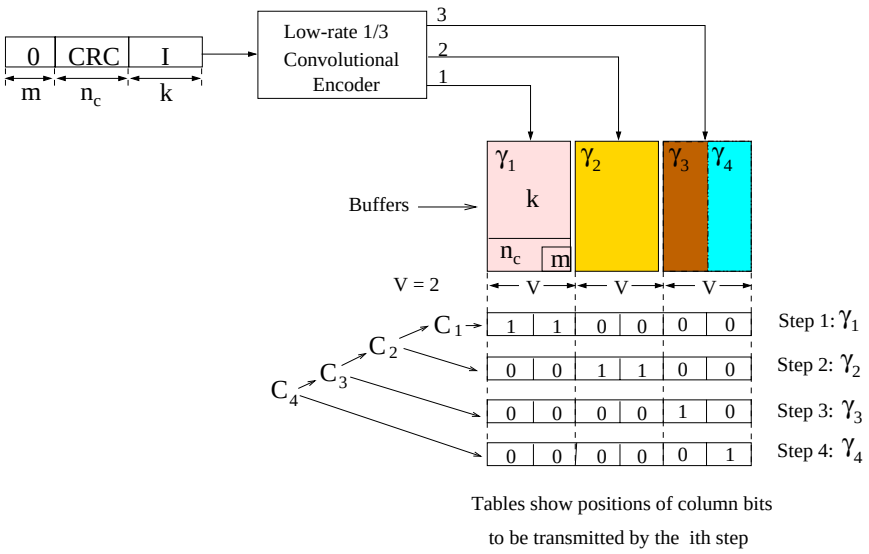

Figure 4: Structure of the RCPC codes.

good channel conditions, but the throughput is shrunken rapidly due to increase of retransmissions under poor channel conditions. The performance of an ARQ scheme can be improved by embedding FEC capability in it, which is so-called *hybrid ARQ*. The advantage of using FEC with ARQ is that many errors can be corrected by the FEC, and retransmissions are requested only when errors are unrecoverable by the FEC. This results in a reduction in the overall retransmission frequency and hence improves throughput during poor channel conditions.

There are two classes of hybrid ARQ schemes: type-I and type-II [7]. A general type-I hybrid ARQ scheme uses a block code (*e.g.*, a Reed-Solomon code), which is designed for simultaneous error correction and error detection. When a received packet is detected in error, the receiver first attempts to locate and correct the errors. If the number of bit errors is within the error-correcting capability of the code, the errors will be corrected and the decoded message will be passed to the user (*e.g.*, compression layer in our system). If an uncorrectable error pattern is detected, the receiver discards the received packet and requests a retransmission.

In the following sections, we describe a general type-II hybrid ARQ scheme and our delay-constrained hybrid ARQ in detail.

4.1 General Type II Hybrid ARQ

A general type-II hybrid ARQ scheme uses a family of codes (*e.g.*, RCPC codes), each of which has different correction capability. Let $C_1 > C_2 > \cdots > C_M$ denote the M rates offered by a family of RCPC codes which are obtained from a low rate C_M (*e.g.*, 1/3) code, called the parent code. The parent code is an ordinary convolutional code. Figure 4 shows the structure of the RCPC codes with a 1/3 convolution code as a parent code. In Fig. 4, the puncture period V is 2 and the rates are 1, 1/2, 2/5, and 1/3. The corresponding puncturing matrices $\mathbf{P}$ are given by

$$\mathbf{P}_{1/1} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{P}_{1/2} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{P}_{2/5} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{P}_{1/3} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{17}$$

As shown in Fig. 4, a number n_c of CRC bits for error detection are attached to the information packet (k bits). Tail bits m zeros are also added to properly terminate the encoder memory and decoder trellis. Each $(k + n_c + m)$-bit segment is encoded with the parent code C_M encoder.

There are three buffers corresponding to three output ports of the 1/3 convolutional encoder. Each buffer contains $(k + n_c + m)$ bits. Each row of the buffers has two bits since the puncturing period is two. The tables in Fig. 4 are generated from the puncturing matrices $\mathbf{P}_{1/1}$ to $\mathbf{P}_{1/3}$ in Eq. (17). Specifically, the first row corresponds to $\mathbf{P}_{1/1}$; the first row and the second row correspond to $\mathbf{P}_{1/2}$; the first three rows correspond to $\mathbf{P}_{2/5}$; the four rows correspond to $\mathbf{P}_{1/3}$.

In Fig. 4, each row in the tables shows the positions of column bits to be transmitted by the i-th step. As a result, we get packets $\gamma_1, \gamma_2, \gamma_3$, and γ_4, which correspond to each row, respectively. In other words, packet γ_1 is generated by code C_1; packets γ_1 and γ_2 are generated by code C_2; packets γ_1, γ_2, and γ_3 are generated by code C_3; packets $\gamma_1, \gamma_2, \gamma_3$ and γ_4 are generated by code C_4.

After the encoding, the sender first sends packet γ_1 to the receiver. If packet γ_1 is received with unrecoverable errors, the receiver generates an ARQ request (note that packet γ_1 is not discarded, which is different from a type-I hybrid ARQ scheme). Then the sender transmits packet γ_2. Upon receipt of packet γ_2, the receiver decodes the combination of packets γ_1 and γ_2 by using code C_2. This procedure of retransmission is continued until decoding process results in no errors being detected. For a truncated type-II hybrid ARQ scheme, the procedure of retransmission is continued until decoding process results in no errors being detected or the maximum number of retransmissions is reached. For our delay-constrained hybrid ARQ, the procedure of retransmission is continued until decoding process results in no errors being detected or the delay bound is violated.

4.2 Delay-constrained Hybrid ARQ

Unlike data, video transmission can tolerate certain errors, but it has stringent delay requirements. This is because the packets arriving later than their playout time will not be displayed. Too many retransmissions could incur excessive delay, which exceeds the delay requirement of real-time video. For this reason, we introduce delay constraint to limit the number of retransmissions and satisfy the QOS requirements of video communications.

Our delay-constrained hybrid ARQ has the same transmission process as a general type-II hybrid ARQ scheme described in Section 4.1. The difference between them is the procedure of retransmission. Next, we describe the retransmission mechanisms for our delay-constrained hybrid ARQ.

Based on who determines whether to send and/or respond to a retransmission request, we design three delay-constrained retransmission mechanisms, namely, receiver-based, sender-based,

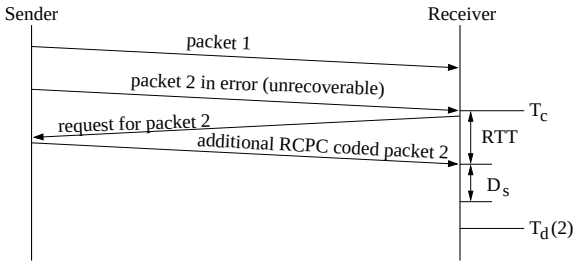

Figure 5: Timing diagram for receiver-based control.

and hybrid control.

Receiver-based control. The objective of the receiver-based control is to minimize the requests of retransmission that will not arrive timely for display. Under the receiver-based control, the receiver executes the following algorithm.

> When the receiver detects unrecoverable errors in a link-layer packet (corresponding to information packet N):
>
> if $(T_c + RTT + D_s < T_d(N))$
>
> send the request for retransmission of a repair packet (corresponding to information packet N) to the sender;

where T_c is the current time, RTT is an estimated round trip time, D_s is a slack term, and $T_d(N)$ is the time when packet N is scheduled for display. The slack term D_s could include tolerance of error in estimating RTT, the sender's response time to a request, and/or the receiver's processing delay (*e.g.*, decoding). If $T_c + RTT + D_s < T_d(N)$ holds, it is expected that the retransmitted packet will arrive timely for display. The timing diagram for receiver-based control is shown in Fig. 5.

Since RTT may not change in wireless environment, RTT can be obtained during set-up phase through a test. Specifically, the source sends out a packet with a timestamp; the receiver sends back the packet when the packet is received; when the packet is received by the source, the source can determine the RTT based on the difference between its current time and the timestamp embedded in the packet.

Sender-based control. The objective of the sender-based control is to suppress retransmission of packets that will miss their display time at the receiver. Under the sender-based control, the sender executes the following algorithm.

> When the sender receives a request for retransmission of a repair packet (corresponding to information packet N):
>
> if $(T_c + T_o + D_s < T'_d(N))$
>
> retransmit a repair packet to the receiver

where T_o is the estimated one-way trip time (from the sender to the receiver), and $T'_d(N)$ is an estimate of $T_d(N)$. To obtain $T'_d(N)$, the receiver has to feedback $T_d(N)$ to the sender. Then, based

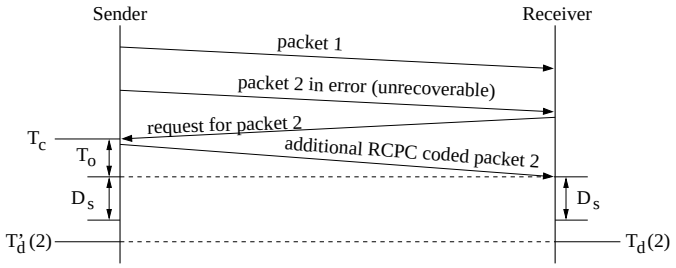

Figure 6: Timing diagram for sender-based control.

on the differences between the sender's system time and the receiver's system time, the sender can derive $T'_d(N)$. The slack term D_s may include error terms in estimating T_o and $T'_d(N)$, as well as tolerance in the receiver's processing delay (*e.g.*, decoding). If $T_c + RTT + D_s < T'_d(N)$ holds, it can be expected that retransmitted packet will reach the receiver in time for display. The timing diagram for sender-based control is shown in Fig. 6.

Hybrid control. The objective of the hybrid control is to minimize the request of retransmissions that will not arrive for timely display, and to suppress retransmission of the packets that will miss their display time at the receiver. The hybrid control is a simple combination of the sender-based control and the receiver-based control. Specifically, the receiver makes decisions on whether to send retransmission requests while the sender makes decisions on whether to disregard requests for retransmission. The hybrid control may achieve better performance at the cost of higher complexity.

Our delay-constrained hybrid ARQ could use any of the three retransmission mechanisms, *i.e.*, receiver-based, sender-based, and hybrid control. In addition, to maximize throughput, we use selective-reject ARQ[1] rather than stop-and-wait ARQ or go-back-N ARQ [14]. Under selective-reject ARQ, the receiver responds with an acknowledgment containing the sequence number (*e.g.*, N) if the Nth packet is received correctly. On the other hand, if the Nth packet is corrupted (there is no way to decode it and know which packet it is), the sender retransmits the incremental redundant packet (*e.g.*, γ_2) when the counter times out; before the counter times out, if an acknowledgment for the $(N+1)$th packet rather than the Nth is received, the sender retransmits the incremental redundant packet for the Nth. To achieve highest throughput for selective-reject ARQ, the sliding-window size needs to be large enough (with respect to the ratio of RTT to T_{trans}, where T_{trans} is the transmission time of the packet) and the time-out counter for a packet needs to be set to $T_{trans} + RTT$ (suppose the transmission time of the acknowledgment packet and the processing time are negligible) [14].

Like other hybrid ARQ schemes [21], our hybrid ARQ combines the advantages of ARQ and FEC. Furthermore, our delay-constrained hybrid ARQ is more adaptive to varying wireless channel conditions while guaranteeing delay bound.

Next, we evaluate the performance of our adaptive QoS control mechanism through simulations.

[1]Selective-reject ARQ uses sliding-window flow control. Each packet has a sequence number indicating the position of the packet in the sliding window. Selective-reject ARQ is also called selective-repeat ARQ [7].

5 Simulation Results

In this section, we implement the architecture described in Section 2 on our simulator and perform a simulation study of video conferencing with MPEG-4. The purpose of this section is to demonstrate (1) the performance improvement of our approach over the classical approach for optimal mode selection, and (2) the performance improvement of our delay-constrained hybrid ARQ over the truncated Type-II hybrid ARQ.

5.1 Simulation Settings

We implement the architecture shown in Fig. 1. At the source side, we use the standard raw video sequence "Miss America" in QCIF format for the video encoder. The encoder employs the rate control described in [20] so that the output rate of MPEG-4 encoder can adapt to the varying available bandwidth $R_a(t)$. The threshold for inserting resynchronization marker is set to 612 bits to match the link-layer payload size k. After compression, the video bit-stream is chopped into packets by the packetizer with a fixed size k of 612 bits. Then the packets are passed to CRC & RCPC encoder. We use 16-bit CRC code to detect bit errors. The 16-bit CRC error detection code will miss error patterns in packets with a probability of the order 10^{-5} of independent identically distributed (i.i.d.) errors. By using more powerful CRC codes, the probability of undetected error can be made negligible at the cost of reduced throughput.

RCPC codes are used as the error correction codes. Two sets of RCPC codes are used with the ARQ schemes. Both sets are generated from a rate 1/4 code with constraint length $K = 5$ and puncturing period $V = 8$ (see Table B.189 in Ref. [2])). The rates of Set $\mathcal{A}$ are 1, 5/6, 5/8, 5/10, 5/12, 5/14, 5/16, 5/17, 5/18, 5/19, and 1/4. The rates of Set $\mathcal{B}$ are 1, 5/10, 5/15, and 1/4. At the receiver side, if an erroneous link-layer packet cannot be recovered by RCPC at its playout time, the link-layer packet and its retransmitted redundancy will be discarded. The sliding-window size for ARQ is set to 256 (using 8 bits), which is large enough (with respect to the ratio of RTT to the packet transmission time, which is approximately 4). We only use receiver-based control for our delay-constrained hybrid ARQ. This is because receiver-based control has the same performance as that of hybrid control when the RTT estimated by the receiver is accurate.

In our simulations, we assume that the feedback channel is error-free. Finite buffers are available at the transmitter and receiver and are large enough for storing transmitted and received packets for the link-layer ARQ protocol. For the simulations, we employ two kinds of wireless channels, namely, additive white Gaussian noise (AWGN) channel and Rayleigh fading channel. Binary phase shift keying (BPSK) with coherent detection is used as the modulation/demodulation scheme with a carrier frequency of 1.9 GHz.

Table 2 lists the parameters used in our simulation. Under such simulation settings, we consider three different encoders for MPEG-4 video as follows.

Encoder A: employs the classical approach for R-D optimized mode selection.

Table 2: Simulation parameters.

MPEG-4 encoder	Frame rate	10 frames/s
	I-VOP refreshment period	50 frames
	Mode	Rectangle
	λ_0	1
	End-to-end delay bound	100 ms
	Resynchronization marker length	17 bits
	Threshold for resynchronization marker	612 bits
Packetizer	Information packet size k	612 bits
CRC & RCPC encoder	Number of bits for packet sequence number	8 bits
	Number of CRC bits n_c	16 bits
	Number of tail bits m	4 bits
Hybrid ARQ	Sliding-window size	256 packets
	Packet processing delay	0 s
	Data transmission rate	64 kb/s
Wireless channel	RTT	10 ms or 30 ms
	channel model	AWGN or Rayleigh fading

Encoder B: implements the globally R-D optimized mode selection described in Section 3. However, feedback of MER is not employed.

Encoder C: implements the globally R-D optimized mode selection described in Section 3. Feedback of MER to the source is employed.

Next, we show our simulation results under an AWGN channel and a Rayleigh fading channel in Sections 5.2 and 5.3, respectively. Each simulation is conducted for 100-second simulation time.

5.2 Performance over an AWGN Channel

In this section, we evaluate the performance of optimal mode selection and hybrid ARQ under the setting of an AWGN channel.

5.2.1 Performance of Optimal Mode Selection

In this simulation, we evaluate the performance of different optimal mode selection schemes, namely, Encoder A, B, and C. For Encoder B, we set MER=1% for optimal mode selection. The round-trip time is set to 10 ms. The delay-constrained hybrid ARQ uses the rates of Set $\mathcal{A}$ for RCPC codes and the delay bound is set to be 100 ms since the frame rate of the MPEG-4 stream is 10 frames/sec. Figure 7 shows the average peak signal-to-noise ratios (PSNR's) of Y component (denoted by PSNR-

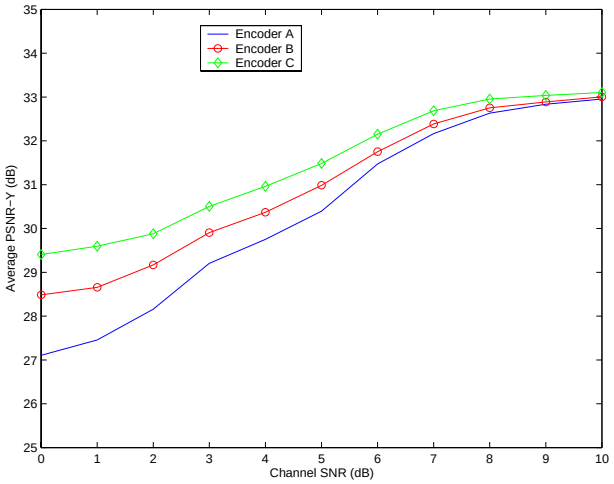

Figure 7: Performance of different video encoders.

Y) under an AWGN channel with different channel signal-to-noise ratios (SNR's).[2] We observe that Encoder C achieves the best performance, Encoder B has the second best performance, and Encoder A performs the worst. This demonstrates that our approach achieves better performance than the classical one, even if feedback mechanism is not employed. In addition, feedback-based scheme (*i.e.*, Encoder C) achieves better performance than non-feedback-based scheme (*i.e.*, Encoder B) since Encoder C has more accurate channel information than Encoder B. Moreover, Figure 7 indicates that the lower the channel SNR, the larger the performance gain achieved by our proposed scheme (*i.e.*, Encoder C). This feature enables our scheme to improve visual presentation quality substantially when a mobile device works under low channel SNR (*e.g.*, between 0 to 5 dB).

5.2.2 Performance of Hybrid ARQ

To compare the performance of the truncated type-II hybrid ARQ [21] and our delay-constrained hybrid ARQ, we run our simulations under two scenarios where only Encoder C is employed.

In the first scenario, we set RTT=10 ms and the maximum number of retransmissions N_m=3 for the truncated hybrid ARQ. The truncated hybrid ARQ uses the rates of Set $\mathcal{B}$ for RCPC codes; our delay-constrained hybrid ARQ uses the rates of Set $\mathcal{A}$ for RCPC codes. Figure 8(a) shows that our delay-constrained hybrid ARQ achieves higher average PSNR-Y than the truncated hybrid ARQ, when the channel SNR is less than 7 dB. This is because when the channel SNR is less than 7 dB, an erroneous link-layer packet can be retransmitted 10 times under our delay-constrained hybrid ARQ without violating the delay bound (*i.e.*, 100 ms) while the truncated hybrid ARQ only allows 3 retransmissions, resulting in lower recovery probability. In other words, the truncated hybrid ARQ has higher probability of packet corruption. When the channel SNR is greater than or equal

[2]Specifically, the channel SNR here is E_b/N_0, where E_b is the bit energy and N_0 is the single-sided power spectral density of noise.

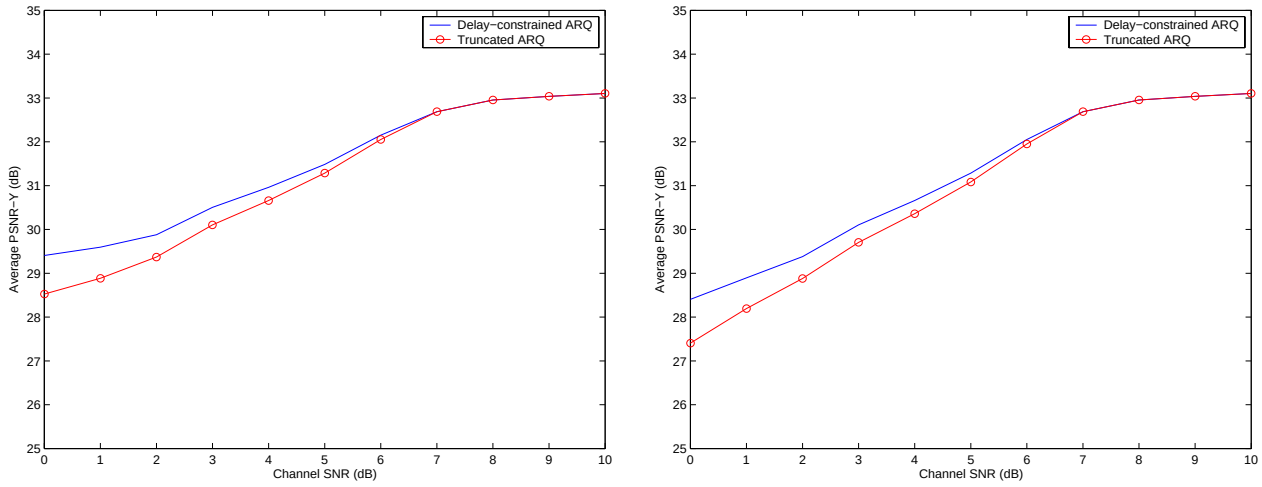

Figure 8: Average PSNR-Y for hybrid ARQ schemes: (a) the first scenario, (b) the second scenario.

to 7 dB, at most 3 retransmissions are enough to recover all the bit errors. So, in this case, the two hybrid ARQ achieve the same average PSNR-Y.

In the second scenario, we set RTT=30 ms and the maximum number of retransmissions N_m=10 for the truncated hybrid ARQ. Both hybrid ARQ scheme use the rates of Set $\mathcal{A}$ for RCPC codes. Figure 8(b) shows that our delay-constrained hybrid ARQ achieves higher average PSNR-Y than the truncated hybrid ARQ, when the channel SNR is less than 7 dB. This is because when the channel SNR is less than 7 dB, an erroneous link-layer packet may be retransmitted up to 3 times under our delay-constrained hybrid ARQ without violating the delay bound (*i.e.*, 100 ms) while the truncated hybrid ARQ allows 10 retransmissions, resulting in waste of bandwidth due to discard of packets that missed their playout time. The more the bandwidth is wasted by the truncated hybrid ARQ, the less bandwidth MPEG-4 encoder can consume. This translates into lower average PSNR-Y for the truncated hybrid ARQ, as shown in Fig. 8(b). This is also confirmed by Fig. 9, which shows that the truncated hybrid ARQ achieves lower throughput than our delay-constrained hybrid ARQ, when the channel SNR is less than 7 dB. The throughput is defined as the ratio of the MPEG-4 encoding rate $R_s(t)$ to the link-layer transmission rate 64 kb/s. On the other hand, when the channel SNR is greater than or equal to 7 dB, at most 3 retransmissions are enough to recover all the bit errors. Hence, in this case, the two hybrid ARQ achieve the same throughput and the same average PSNR-Y.

The purpose of using the above two scenarios is to show that the truncated hybrid ARQ is not adaptive to varying RTT. In contrast, our delay-constrained hybrid ARQ is capable of adapting to varying RTT and achieving better performance than the truncated hybrid ARQ, for *real-time* video transmission over wireless channels.

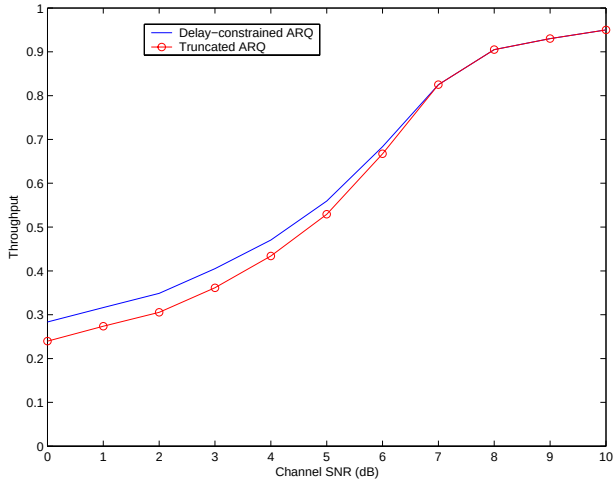

Figure 9: Throughput for hybrid ARQ schemes.

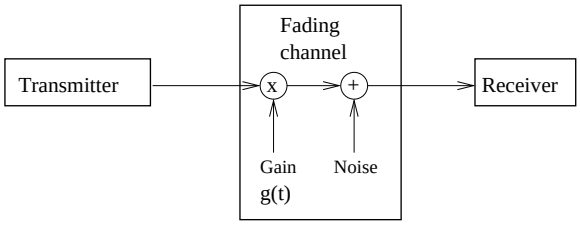

Figure 10: Fading channel model.

5.3 Performance over a Rayleigh Fading Channel

In this section, we evaluate the performance of optimal mode selection and hybrid ARQ, under the setting of a Rayleigh fading channel model as shown in Fig. 10. As compared to an AWGN channel, a Rayleigh fading channel with a specified Doppler spectrum induces correlated error bursts, which are more difficult for the hybrid ARQ to recover.

In our simulations, Rayleigh flat-fading voltage-gains $h(t)$ are generated by a first-order auto-regressive (AR(1)) model as below:

$$h(t) = \kappa \times h(t-1) + v(t), \tag{18}$$

where $v(t)$ are zero-mean i.i.d. complex Gaussian variables. The coefficient κ determines the Doppler rate, i.e., the larger the κ, the smaller the Doppler rate. Specifically, the coefficient κ can be determined by the following procedure: 1) compute the coherence time T_c by [11, page 165]

$$T_c \approx \frac{9}{16\pi f_m}, \tag{19}$$

Table 3: Simulation parameters for a Rayleigh fading channel.

Sampling interval T_s	1/64000 second
Carrier frequency	1.9 GHz
Maximum Doppler rate f_m	5.3, 70, 211 Hz (corresponding to vehicle speed of 3, 40, 120 km/h, corresponding to coherence time T_c of 33.8, 2.6, 0.8 msec.)
Average SNR	0 to 20 dB

where f_m is the maximum Doppler rate, and the coherence time is defined as the time, over which the time auto-correlation function of the fading process is above 0.5; 2) compute the coefficient κ by[3]

$$\kappa = 0.5^{T_s/T_c}, \tag{20}$$

where T_s is the sampling interval (a bit duration). By absorbing the noise variance into the channel gain, we have the power gain $g(t) = |h(t)|^2 P_0/\sigma^2$, where the transmission power P_0 and noise variance σ^2 are constant. Therefore, the average (received) SNR is $E[g(t)]$.

The Rayleigh fading channel gains are generated using the parameters described in Table 3.

5.3.1 Performance of Optimal Mode Selection

In this simulation, we evaluate the performance of different optimal mode selection schemes, namely, Encoder A, B, and C. For Encoder B, we set MER=1% for optimal mode selection. The round-trip time is set to 10 ms. The delay-constrained hybrid ARQ uses the rates of Set $\mathcal{A}$ for RCPC codes and the delay bound is set to be 100 ms. Figure 11 shows the average PSNR-Y under a Rayleigh fading channel with different average SNR's and different Doppler rates. Again, we observe that Encoder C achieves the best performance, Encoder B has the second best performance, and Encoder A performs the worst. This demonstrates that our approach achieves better performance than the classical one, for video transmission over a Rayleigh fading channel. Furthermore, the three figures in Fig. 11 show that as the Doppler rate increases, the average PSNR-Y achieved also increases. This indicates that our hybrid ARQ can utilize the time diversity in the underlying fading channel to improve video presentation quality. This is expected since the delay bound (*i.e.*, 100 ms) is larger than the coherence time T_c of 33.8, 2.6, and 0.8 ms (see Table 3). That is, our hybrid ARQ can utilize the time diversity of the underlying fading channel, if the delay constraint imposed by the video application is larger than the coherence time of the fading channel.

[3]The auto-correlation function of the AR(1) process is κ^n, where n is the number of sampling intervals. Solving $\kappa^{T_c/T_s} = 0.5$ for κ, we obtain (20).

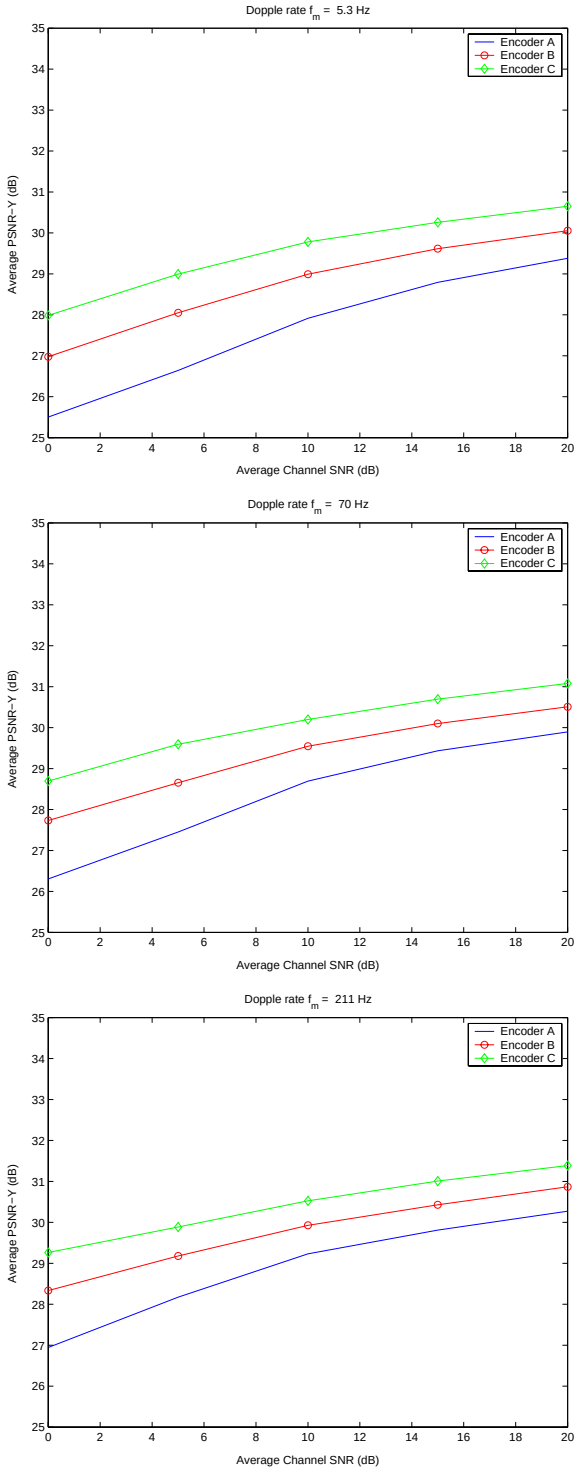

Figure 11: Performance of different video encoders under the setting of a Rayleigh fading channel.

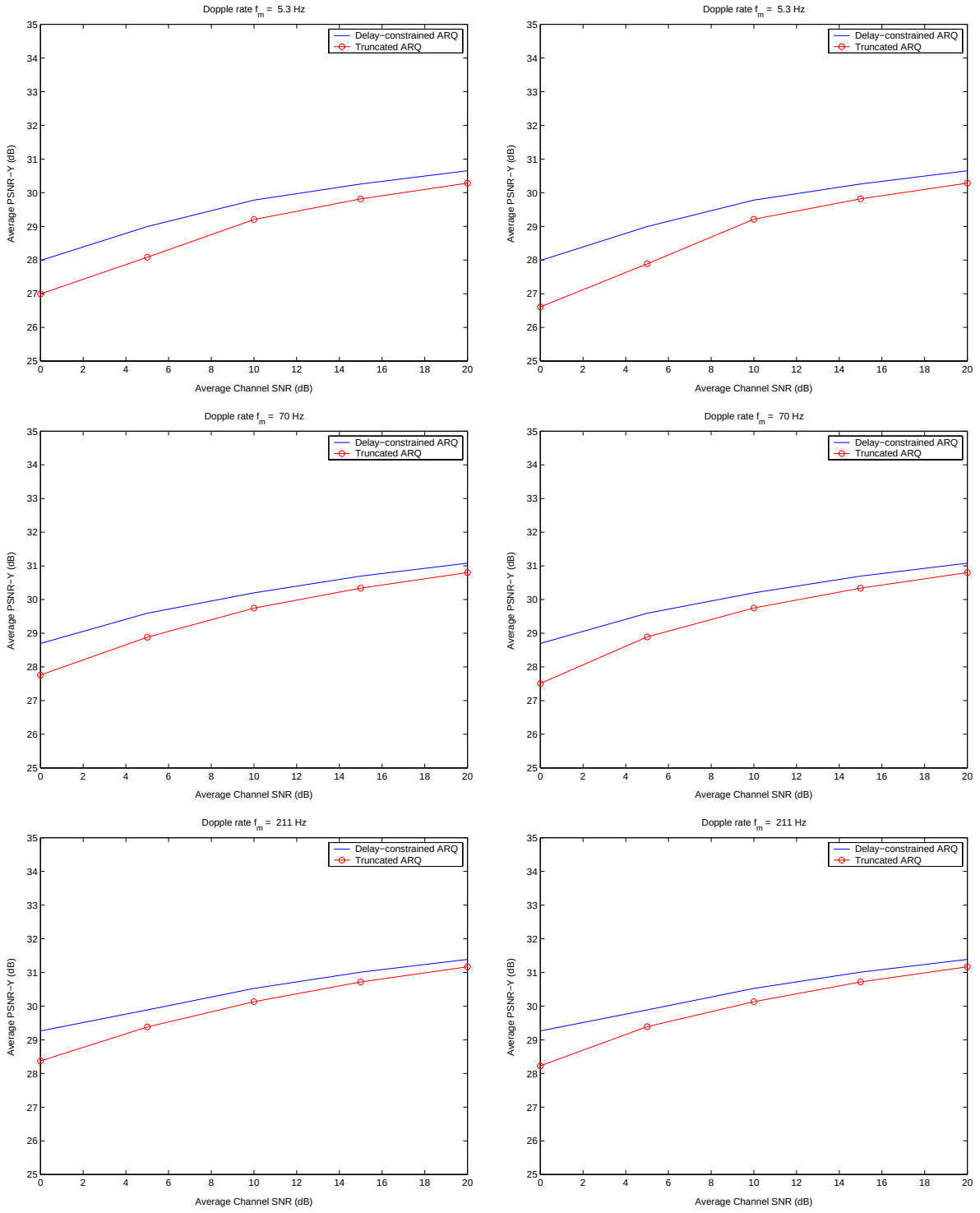

Figure 12: Average average PSNR-Y for hybrid ARQ schemes under the setting of a fading channel: (a) the first scenario, (b) the second scenario.

5.3.2 Performance of Hybrid ARQ

To compare the performance of the truncated type-II hybrid ARQ and our delay-constrained hybrid ARQ, we again run our simulations under two scenarios where only Encoder C is employed.

In the first scenario, we set RTT=10 ms and the maximum number of retransmissions N_m=3 for the truncated hybrid ARQ. The truncated hybrid ARQ uses the rates of Set $\mathcal{B}$ for RCPC codes; our delay-constrained hybrid ARQ uses the rates of Set $\mathcal{A}$ for RCPC codes and the delay bound is set to be 100 ms. Figure 12(a) shows that our delay-constrained hybrid ARQ achieves higher average PSNR-Y than the truncated hybrid ARQ, for different average channel SNR and different Doppler rates. As we mentioned before, the reason is that our delay-constrained hybrid ARQ has lower probability of packet corruption. Compared with Figure 8(a), we see that for the same average channel SNR, the average PSNR-Y achieved under the Rayleigh fading channel is lower than that under the AWGN channel. This is conceivable since compared to an AWGN channel, a Rayleigh fading channel may cause long bursts of errors, which are hard to recover.

In the second scenario, we set RTT=30 ms and the maximum number of retransmissions N_m=10 for the truncated hybrid ARQ. Both hybrid ARQ scheme use the rates of Set $\mathcal{A}$ for RCPC codes. and the delay bound is set to be 100 ms. Figure 12(b) shows that our delay-constrained hybrid ARQ achieves higher average PSNR-Y than the truncated hybrid ARQ, for different average channel SNR and different Doppler rates. Again, the reason is that the truncated hybrid ARQ may waste bandwidth unnecessarily, which translates into lower average PSNR-Y.

In summary, the simulation results have demonstrated (1) the performance improvement of our approach over the classical approach for optimal mode selection, and (2) the performance improvement of our delay-constrained hybrid ARQ over the truncated Type-II hybrid ARQ, for real-time video transmission over wireless channels.

6 Concluding Remarks

Due to high bit-error-rate in wireless environments, real-time video communication over wireless channels poses a challenging problem. To address this challenge, this paper introduced adaptive QoS control to achieve robustness for video transmission over wireless. Our adaptive QoS control consists of optimal mode selection and delay-constrained hybrid ARQ. Our optimal mode selection provides QoS support (*i.e.*, error resilience) on the compression layer; it is adaptive to time-varying channel characteristics due to feedback of macroblock error ratio. Our delay-constrained hybrid ARQ provides QoS support (*i.e.*, reliability and bounded delay) on the link layer; it is adaptive to time-varying wireless channel due to its consideration of the display deadlines of the packets. The main contributions of this paper are: (1) an optimal mode selection algorithm which provides the best trade-off between compression efficiency and error resilience in R-D sense, and (2) delay-constrained hybrid ARQ which is capable of providing reliability for the compression layer while guaranteeing delay bound and achieving high throughput. Simulation results have shown that the proposed adaptive QoS control outperforms the previous approaches for MPEG-4 video communications

over wireless channels. The reason for this is that our adaptive QoS control combines the best features of error-resilient source encoding, FEC and delay-constrained retransmission.

Appendix

6.1 Facts

Fact 1 *Suppose that events $\mathcal{A}_i$ form a partition of event $\mathcal{A}$ and events $\mathcal{B}_{ij}$ form a partition of $\mathcal{A}_i$. Denote p_i the probability of event $\mathcal{A}_i$ and q_j the conditional probability of an event $\mathcal{B}_{ij}$ assuming $\mathcal{A}_i$. Define a random variable $\mathbf{x}$ on $\mathcal{A}$ and a random variable $\mathbf{y}_i$ on $\mathcal{A}_i$. Then we have*

$$E\{\mathbf{x}\} = \sum_i (p_i \cdot E\{\mathbf{y}_i\}). \tag{21}$$

References

[1] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 952–965, June 2000.

[2] P. Frenger, P. Orten, T. Ottosson, and A. Svensson, "Multi-rate convolutional codes," *Tech. Report No. 21,* Dept. of Signals and Systems, Chalmers University of Technology, page 131, April 1998.

[3] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.,* vol. 36, pp. 389–400, April 1988.

[4] ISO/IEC JTC 1/SC 29/WG 11, "Information technology - coding of audio-visual objects, part 1: systems, part 2: visual, part 3: audio," FCD 14496, Dec. 1998.

[5] S. Kallel and D. Haccoun, "Generalized type-II hybrid ARQ scheme using punctured convolutional coding," *IEEE Trans. Commun.,* vol. 38, pp. 1938–1946, Nov. 1990.

[6] J. Lee and B. W. Dickinson, "Rate-distortion optimized frame type selection for MPEG encoding," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 7, no. 3, pp. 501–510, June 1997.

[7] S. Lin and D. Costello, "Error control coding: fundamentals and applications," Englewood Cliffs, NJ: Prentice-Hall, 1983.

[8] H. Liu and M. El Zarki, "Performance of H.263 video transmission over wireless channels using hybrid ARQ," *IEEE J. on Selected Areas in Communications,* vol. 15, no. 9, pp. 1775–1786, Dec. 1997.

[9] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine,* pp. 74–90, Nov. 1998.

[10] K. Ramchandran, A. Ortega and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Processing,* vol. 37, pp. 533–545, Aug. 1994.

[11] T. S. Rappaport, *Wireless Communications: Principles & Practice,* Prentice Hall, 1996.

[12] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoust., Speech and Signal Processing,* vol. 36, no. 9, pp. 1445–1453, Sept. 1988.

[13] B. Sklar, "Rayleigh fading channels in mobile digital communication systems Part I: characterization," *IEEE Commun. Mag.,* vol. 35, pp. 90–100, July 1997.

[14] W. Stallings, "Data & computer communications," 6th ed. pp. 208–233, Upper Saddle River, NJ: Prentice-Hall, 2000.

[15] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine,* pp. 74–90, Nov. 1998.

[16] J. Villasenor, Y.-Q. Zhang, and J. Wen, "Robust video coding algorithms and systems," *Proceedings of the IEEE,* vol. 87, no. 10, pp. 1724–1733, Oct. 1999.

[17] Y. Wang and S. Lin, "A modified selective-repeat type-II hybrid ARQ system and its performance analysis," *IEEE Trans. Commun.,* vol. 31, pp. 593–608, May 1983.

[18] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, no. 2, pp. 182–190, April 1996.

[19] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 977–995, June 2000.

[20] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 10, no. 6, Sept. 2000.

[21] Q. Zhang and S. A. Kassam, "Hybrid ARQ with selective combining for fading channels," *IEEE J. on Selected Areas in Communications,* vol. 17, no. 5, pp. 867–880, May 1999.

[22] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 966–976, June 2000.