Adaptive source rate control for real-time wireless video transmission

Hang Liu * and Magda El Zarki

Video Processing and Telecommunications Laboratory, Department of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA

Hybrid ARQ schemes can yield much better throughput and reliability than static FEC schemes for the transmission of data over time-varying wireless channels. However these schemes result in extra delay. They adapt to the varying channel conditions by retransmitting erroneous packets, this causes variable effective data rates for current PCS networks because the channel bandwidth is constant. Hybrid ARQ schemes are currently being proposed as the error control schemes for real-time video transmission. An important issue is how to ensure low delay while taking advantage of the high throughput and reliability that these schemes provide for. In this paper we propose an adaptive source rate control (ASRC) scheme which can work together with the hybrid ARQ error control schemes to achieve efficient transmission of real-time video with low delay and high reliability. The ASRC scheme adjusts the source rate based on the channel conditions, the transport buffer occupancy and the delay constraints. It achieves good video quality by dynamically changing both the number of the forced update (intracoded) macroblocks and the quantization scale used in a frame. The number of the forced update macroblocks used in a frame is first adjusted according to the allocated source rate. This reduces the fluctuation of the quantization scale with the change in the channel conditions during encoding so that the uniformity of the video quality is improved. The simulation results show that the proposed ASRC scheme performs very well for both slow fading and fast fading channels.

1. Introduction

Real-time video services require high transmission reliability and stringent end-to-end delay. Wireless links on the other hand are error-prone, bandlimited and time-varying. Error control schemes are necessary to obtain high transmission reliability required by video services. Traditionally, forward error correction (FEC) codes have been used for real-time services because they maintain a constant throughput and a bounded delay. Wireless channels are time-varying. FEC codes can be chosen to guarantee certain error rate requirements for the worst channel conditions. However, this causes unnecessary overhead and wastes bandwidth when the channel is in a good state.

Recently, for wireless environments, it has been shown that automatic repeat request (ARQ) and hybrid ARQ schemes can significantly improve the video transmission reliability and provide for much higher throughput than FEC schemes because they can effectively adapt to the varying channel conditions [10,12]. The new video conferencing standard, H.324, will support hybrid ARQ schemes for wireless video communications [6,7]. The MPEG-4 standard committee is also considering to adopt a hybrid ARQ scheme for video transmission in error-prone environments [4]. Several ARQ-based or hybrid ARQ-based schemes have been proposed for wireless ATM networks to support real-time video transmission [18].

However, retransmissions in hybrid ARQ schemes cause delay. Long delays are intolerable for interactive real-time applications. If data cannot arrive at the receiver within the required delay bound, it is considered as being lost. For example, it has been suggested that for video conferencing applications the value of the end-to-end delay must be less than 400 ms and it is preferable if the end-to-end delay is below 200 ms [1]. In the current personal communication services (PCS) networks [14], the total bandwidth for a channel is constant. The wireless channel condition changes over time. Hybrid ARQ schemes adapt to the varying channel conditions by retransmitting erroneous packets. When the channel is good, no retransmissions are required and the effective data rate can be high. When the channel becomes poor, the retransmissions use up bandwidth and thus reduce the effective data rate (the effective data rate is defined as the rate of the information that is correctly transmitted). This results in a varying effective data rate from the point of view of the video source. Especially for a slow fading channel, when the channel is in a state, poor or good, it will remain in that state for a long duration so that the effective data rate becomes very bursty due to retransmissions. An important issue therefore is how to guarantee low end-to-end delay while taking advantage of the high throughput and reliability provided by the hybrid ARQ schemes.

In this paper we investigate the impact of hybrid ARQ on real-time video transmission. We focus on the transmission of H.263 coded low bitrate video over a constant bandwidth wireless channel (QCIF format and 15 frames/s over 32 kbit/s channels) which complies with the current PCS networks. We propose an adaptive source rate control (ASRC) scheme and demonstrate that it can work together with the hybrid ARQ error control schemes to achieve efficient transmission of real-time video with low delay and

* Current address: NEC USA, Inc., C&C Research Lab, 4 Independence Way, Princeton, NJ 08540, USA.

high reliability. The channel condition can be estimated by the transmitter from the outcome of recent packet transmissions. This information is available to the error control module in the form of ACKs received as feedback from the receiver. It can be passed up to the adaptive source rate control scheme. The adaptive source rate control scheme uses this to forecast the channel's effective data rate before a video frame is encoded. Once such an estimate is available, the adaptive source rate control scheme determines the target number of bits for the next encoded frame (i.e., the target source rate) based on the estimated effective data rate, the current transport buffer occupancy (i.e., the amount of data waiting for transmission before this frame), and the required delay bound so that the available channel bandwidth is efficiently used and the generated video data can correctly arrive at the receiver within the delay bound.

The source rate control scheme is also responsible for efficiently assigning the available bits in a frame to achieve overall good video quality. Traditionally a target source rate is achieved by changing the quantization scale during the encoding of a frame [2,3,5,13,16]. Adjusting the source rate according to the channel condition may result in the undesirable fluctuation of the encoded video quality if only the quantization scale is changed. In H.263, some macroblocks can be forced to be intracoded in a frame to stop possible error propagations [8]. These macroblocks are called forced update macroblocks (FUMBs). The proposed ASRC scheme changes both the number of the FUMBs and the quantization scale used in a frame to achieve the allocated target source rate. The number of FUMBs in a frame is first adjusted. Only when the target source rate cannot be obtained by adjusting the number of FUMBs in a frame, is the quantization scale changed. This reduces the fluctuation in the value of the quantization scale during encoding so that a more uniform video quality is obtained. The simulation results are presented to show that the proposed ASRC scheme performs very well for both slow fading and fast fading transmission environments.

Although we consider PCS networks, this study can be applied to wireless ATM networks [15]. In wireless ATM networks, a channel can be guaranteed for constant bit rate (CBR) video transmission and retransmissions are realized by allocating extra bandwidth. However, the extra bandwidth required for retransmissions may not always be available (for example, during periods of high network load or very long fades). If there is not enough extra bandwidth for retransmissions, some retransmissions may use the reserved CBR channel. The scheme proposed here can be used to achieve good video quality.

We use a type-II hybrid ARQ scheme in our study because it has been shown to be efficient and has powerful error correction capability [11,12]. The general conclusions of this paper, however, should hold for other ARQ and hybrid ARQ schemes because the underlying operation for any retransmission-based error control scheme is similar.

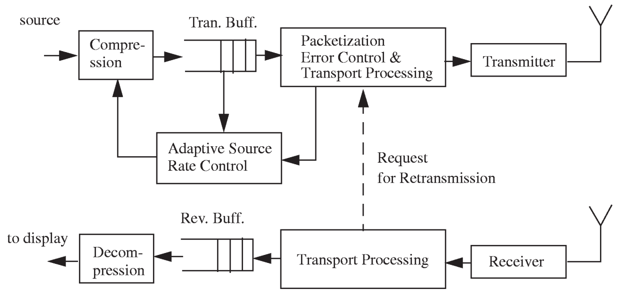

Figure 1. Wireless video transmission system with hybrid ARQ error control.

This paper is organized as follows: In section 2, we briefly describe the communication model. We discuss the proposed adaptive source rate control scheme in section 3. In section 4, the performance of the proposed scheme is studied. Finally, the summary is given in section 5.

2. Communication system model

Figure 1 shows the model for the hybrid ARQ-based wireless transmission system under investigation. For real-time video applications, the raw video source is compressed and passed to the transport module. The transport module prepares the bitstream for delivery by segmenting the data into packets and adding the appropriate error protection. The packets are sent through a wireless link. When a packet error is detected, the retransmission is performed. Below, we describe some of the system components in more detail.

2.1. H.263 video compression standard

The video encoder uses the new ITU H.263 coding standard [8]. H.263 targets the transmission of video telephony through the Public Switched Telephone Network (PSTN) at very low date rates (less than 64 kbit/s). The ITU is adapting it for wireless applications because the low bitrate makes it well suited for current band-limited wireless networks.

H.263 supports two basic picture types, intra-coded (I) picture, encoded using information only from itself, and predictive-coded (P) picture, encoded using motion compensated prediction from a past reference frame. An optional mode is to use the PB-frame. A PB-frame consists of one P-picture and one B-picture. The P-picture is predicted from the last encoded P-picture (or I-picture). The B-picture is predicted from both the last encoded P-picture (or I-picture) and the present P-picture. The PB-frame can reduce the bitrate, but it costs at least one frame interval of encoding delay because a P-picture and a B-picture need to be coded in the same unit. For low bitrate and low frame rate applications, this delay is quite high. For example, if we consider that the frame interval is 67 ms (i.e., the frame rate is 15 frames/s). When the PB-frame is used, an additional 1 frame encoding delay (67 ms) is imposed. We assume that the size of a PB-frame is S_{PB} and that of a P-frame is S_{P}. Let $S_{\mathrm{PB}} = \alpha S_{\mathrm{P}}$. If $\alpha = 1.5$, an additional half-frame transmission delay is needed when using a constant bitrate transmission channel. As the display cannot be started until the entire frame has been received, the extra delay for a PB-frame then becomes 1.5 frame intervals (100 ms). Furthermore, the encoding and decoding processes for the PB-frames are much more complex, this increases the processing delay (this is processor speed dependent). Therefore, one needs to make a careful decision on whether or not the PB-frame mode should be used for a low bitrate interactive application.

The I-frames provide coding robustness for source coding distortion and transmission errors, and also serve as access points to any segment in a sequence. However, I-pictures are the least efficient as they exploit only spatial redundancies. P-pictures exploit both temporal and spatial redundancies. For video conferencing applications, the video motion is generally low. A significant amount of compression gain can be achieved using motion compensation. The I-frame size is much larger than the P-frame size for the same picture quality. The transmission delay for I-pictures is very high if the transmission is over a constant rate channel. It therefore makes sense to avoid using I-frames. A solution would be to only intracode the first frame and after that use only P-frames and/or PB-frames. However for robustness, some macroblocks (MBs) can be forced to be intraframe mode encoded (I-MBs) in a frame (called forced updated macroblocks, FUMBs) to stop possible error propagation. FUMBs are used to update portions of a frame incrementally to achieve a complete update. This approach helps maintain a smoother processing and bandwidth requirement between frames. The H.263 standard requires that each macroblock be encoded as an I-MB at least every 132 frames and this maximum insertion interval is defined as the forced update period. By shortening the forced update period, we increase error robustness. Of course, this scheme does not produce fixed access points and it may take a long time to completely recover the whole frame. However, it has been shown to work well for low-motion and low bitrate video conferencing applications.

2.2. Hybrid ARQ

In order to study the effect of retransmissions, we consider a type-II hybrid ARQ scheme with a selective repeat retransmission protocol [11,12]. A brief summary of the scheme is given here. A more complete description can be found in [12].

The hybrid ARQ scheme employs two codes, C_0 and C_1. C_0 is an (N, K) cyclic redundancy check (CRC) code, which is used as the error detection code. C_1 is a half-rate invertible shortened RS code $(2k, k, t)$ with m bits per symbol for both error detection and correction. For the invertible code, the information block and the parity block have the same length, and the information block can be obtained uniquely from the parity block by a simple inverting algorithm [11].

Information data is first divided into blocks with k symbols (km bits) per block. An information block D is encoded using the half-rate invertible RS code C_1. The k-symbol-long parity block P(D) corresponding to D is formed. (D, P(D)) is a codeword in C_1. After RS encoding, λ consecutive information blocks are transmitted in the initial transmission, and λ corresponding parity blocks are saved in the transmitter buffer for possible transmission at a later time. Before transmission, these λ consecutive information blocks are input row-by-row into a $\lambda \times k$-symbol interleaving buffer, and then read out column-by-column in symbols. These interleaved information symbols are converted to $K = \lambda \times k \times m$ bits. Based on the (N, K) CRC code C_0, $(N - K)$ CRC bits are attached to K information bits to form an N-bit information macroblock I. I is transmitted. A timer at the transmitter is set when I is transmitted.

Let $\tilde{\mathrm{I}}$ be the received version of I at the receiver. The CRC check is then performed on $\tilde{\mathrm{I}}$ and the CRC bits are removed. If no error is detected in the sequence $\tilde{\mathrm{I}}$, $\tilde{\mathrm{I}}$ is assumed to be error free and is accepted by the receiver (after removing the CRC bits and deinterleaving with a $\lambda \times k$-symbol de-interleaver). At the same time, a positive ACK is sent to the transmitter. If the presence of an error pattern is detected in $\tilde{\mathrm{I}}$, $\tilde{\mathrm{I}}$ is then deinterleaved and stored in the receiver buffer for possible reprocessing at a later time and no ACK is sent. At the transmitter, if an ACK is received before the timer expires, the transmitter knows that the initial transmission was successful and it discards the λ corresponding parity blocks. If not, it assumes that uncorrected errors occurred in the initially transmitted packet. The transmitter then interleaves the λ corresponding parity blocks, adds the CRC bits and forms a parity macroblock P(I). P(I) is sent to the receiver in the retransmission.

Let $\widetilde{\mathrm{P(I)}}$ be the received parity packet. After $\widetilde{\mathrm{P(I)}}$ is received, the CRC check is then performed and the CRC bits are removed. If no errors are detected in $\widetilde{\mathrm{P(I)}}$, after deinterleaving, the receiver inverts all the RS parity blocks in $\widetilde{\mathrm{P(I)}}$, denoted by $\mathrm{I}(\widetilde{\mathrm{P}})$, and accepts $\mathrm{I}(\widetilde{\mathrm{P}})$ as the original information data (since the RS code is invertible). If the presence of an error pattern is detected in $\widetilde{\mathrm{P(I)}}$, $\widetilde{\mathrm{P(I)}}$ is deinterleaved and combined with the deinterleaved erroneous data sequence $\tilde{\mathrm{I}}$ (stored in the receiver buffer) to form the λ rate $1/2$ RS codes. Error correction is then performed on the RS codes. If the errors are correctable by the RS codes or only one retransmission is allowed due to the delay constraint of real-time services, the RS decoded message is accepted.

If an uncorrectable error pattern is detected by the RS decoder and multiple retransmissions are allowed, the erroneous parity data $\widetilde{\mathrm{P(I)}}$ is saved in the receiver buffer, the old erroneous information sequence $\tilde{\mathrm{I}}$ is discarded and the retransmission of the information packet I is requested. When the new $\tilde{\mathrm{I}}$ is received, it is used to recover the information as described before. If this fails, the new erroneous information data $\tilde{\mathrm{I}}$ and the erroneous parity data (previously stored in the receiver buffer) are combined to form the λ rate

$1/2$ RS codes for error correction. If the errors are still not correctable, the old $\widetilde{\mathrm{P}}(\mathrm{I})$ is discarded and $\tilde{\mathrm{I}}$ is stored in the receiver buffer. The next retransmission will be the parity packet P(I). This process continues, i.e., alternating transmissions of the information packet I and the parity packet P(I), until the data is successfully accepted or the allowed maximum retransmission number is reached.

3. Adaptive source rate control

The goal of a real-time video transmission scheme is to efficiently use the available channel bandwidth and to ensure that all the frames correctly arrive at the receiver within the required delay bound. We believe that the application system (video source coder using source rate control) and the hybrid ARQ based error control module can work together to achieve this goal. The source coding rate should be adapted to the channel conditions. When the channel conditions are good, the source rate can be high and when the channel conditions are poor, the source rate is reduced and the extra bandwidth is used for retransmissions to reduce the channel errors. The change in source rate may result in the fluctuation of the encoded video quality. We first adjust the number of forced update macroblocks (FUMBs) in a frame. Only when the target source rate cannot be obtained by adjusting the number of FUMBs in a frame, is the quantization scale changed. This reduces the fluctuation in the value of the quantization scale during encoding so that a more uniform video quality is obtained. Based on the above observation, we propose an adaptive source rate control (ASRC) scheme.

The ASRC scheme can be decomposed into two phases: (1) bit allocation for each frame, and (2) bit assignment control in each frame. The first phase determines the target number of bits for each frame (target source rate). The second phase is to achieve the target bit allocation. We focus on the first phase in section 3.1 and discuss the second phase in section 3.2.

3.1. Bit allocation control

For a fading channel with a hybrid ARQ scheme, effective data rate (EDR) as seen by the source varies over time. We control the target number of bits for each frame based on the estimated EDR, the current transport buffer occupancy and the delay constraints. Before describing the algorithm, we first define a few terms:

- T_{tar}: target frame interval ($1/T_{\mathrm{tar}}$ is target frame rate);
- $D_{\max}$: frame delay bound in units of seconds (the maximum delay that the application can tolerate from the time instant that a frame is sent to the transport buffer to the time instant that it is received at the receiver);
- D_0: one-way channel delay;
- W: size of the sliding window to measure EDR;
- I_{p}: interval between two consecutive transmissions for a packet;
- S_n: number of correctly transmitted packets in the window before the nth frame is encoded;
- μ_n: estimated EDR before the nth frame is encoded (i.e., the rate at which the information bits are correctly transmitted, both the channel coding overhead and the retransmission overhead are excluded);
- R: channel information data rate if there are no retransmissions (the channel coding overhead is excluded);
- B_{p}: physical size of the transport buffer allocated for the connection;
- B_{h}: high buffer threshold for a current EDR, given the required frame delay bound;
- B_{l}: low buffer threshold;
- B_{tar}: target buffer size;
- B: current buffer occupancy before the frame is encoded;
- F_n: target frame size for the nth frame;
- $F_{\min}$: lower bound for the target frame size in order to guarantee minimal quality.

The target number of bits for a frame is computed dynamically from the average EDR within a sliding window with a window size of W. A smaller window size increases the response of the algorithm to changes in the channel conditions, but it may cause a larger variance in the target source rate. On the other hand, increasing the window size will improve the uniformity of the target source rate, but it may cause a larger buffer buildup and a longer delay. The sliding window is updated each frame. The frame interval does not have to be a multiple of the packet transmission interval. The EDR is determined before the nth frame is encoded. If each packet contains the same number of information bits, the EDR can be obtained based on how many packets were positively acknowledged among the last W packets in the window before their ARQ timers expire:

$$\mu_n = \frac{S_n}{W} R. \tag{1}$$

If the buffer occupancy is large, it should be brought down in order to accommodate a possible EDR reduction and fluctuations in the source encoding rate. We define a target buffer size. The target buffer size should be small and guarantee that no buffer underflow will occur. We choose the target buffer size to be the maximum number of bits that the channel can transmit during a frame interval. It can be obtained as follows:

$$B_{\mathrm{tar}} = \left\lceil \frac{T_{\mathrm{tar}}}{I_{\mathrm{p}}} \right\rceil I_{\mathrm{p}} R, \tag{2}$$

where $\lceil a \rceil$ is the minimum integer larger than or equal to a. The target size (in bits) for the nth frame is decided as follows:

$$F_n = \mu_n T_{\text{tar}} - \left\lceil \frac{\mu_n T_{\text{tar}} + B - B_{\text{tar}}}{\kappa} \right\rceil. \tag{3}$$

The second term in the equation attempts to maintain the buffer occupancy around B_{tar}. κ ($\kappa \geqslant 1$) is a tuning parameter which controls how quickly the buffer occupancy approaches B_{tar}. When κ is small, it converges very fast but may cause a large variance in the source rate between two consecutive frames. A larger κ results in better uniformity of the source rate, but at the cost of a longer period for changing the buffer occupancy, which may result in a larger buffer buildup and a longer delay. The target frame size should also have a lower bound too so that a minimal quality is guaranteed, therefore we have $F_n = \text{Max}(F_{\text{min}}, F_n)$, where F_{min} is chosen to be $RT_{\text{tar}}/4$.

Note that the high buffer threshold B_{h} also changes with the EDR. When the EDR is small, B_{h} should be small in order to guarantee that the data in the buffer can be transmitted within the required frame delay bound D_{max}, i.e., the frame delay is not above the frame delay bound. For an estimated value of μ_n, B_{h} is calculated as

$$B_{\text{h}} = (D_{\text{max}} - D_0)\mu_n. \tag{4}$$

To prevent the buffer occupancy from exceeding the high buffer threshold, we have that

$$F_n = \rho B_{\text{h}} - B, \quad \text{when} \quad F_n + B > \rho B_{\text{h}}, \tag{5}$$

where $0 \leqslant \rho \leqslant 1$ is used to compensate for the bit rate deviation from the target rate in the source encoding process (note that the encoding process does not exactly result in the target number of bits, as we discuss later). We let ρ equal 0.95 because as shown in the next subsection, our bit assignment control algorithm controls the source rate very tightly. When the buffer occupancy is close to the high buffer threshold, we have two options to ensure certain quality of the encoded video frame: (1) setting $F_n = \text{Max}(F_{\text{min}}, F_n)$, or (2) skipping this frame, i.e., changing the frame rate. We use the first option here.

We also want to prevent the buffer from underflow:

$$F_n = B_{\text{l}} + \mu_n T_{\text{tar}} - B, \quad \text{when} \quad F_n + B - \mu_n T_{\text{tar}} < B_{\text{l}}. \tag{6}$$

We choose B_{l} equal to zero in order to reduce the delay.

A physical buffer size of $B_{\text{p}} = (D_{\text{max}} - D_0)R$ is chosen to guarantee the delay and that no data loss occurs under all channel conditions. Note that a B_{p} greater than $(D_{\text{max}} - D_0)R$ is useless as the bits in the buffer cannot completely arrive at the receiver within the required frame delay bound D_{max} even if the channel is in a good condition and no retransmissions are required. To prevent buffer overflow we have the following limitation for a frame size:

$$F_n = \rho B_{\text{p}} - B, \quad \text{when} \quad F_n + B > \rho B_{\text{p}}.$$

3.2. Bit assignment control

Once the target number of bits for a frame has been decided upon, the next task is how to achieve that target number. Traditionally, the quantization scale (Q) is adjusted. There are two methods for adjusting the value of Q: (1) the quantization scale for the current frame or macroblock is set based on the previous bit count result, the previous quantization scale, and the estimated rate-quantization (R-Q) model [3,5,13], and (2) using an iterative algorithm to find the quantization scale for the current frame which is best matched to the target number of bits [2,16]. The former is a simple prediction and easy to implement. However it may result in uneven visual quality within a picture and large deviations from the target. The latter is more accurate and results in a relatively uniform visual quality within a picture. Although the iterative technique is more computationally intensive, it turns out that only two to three passes are required for a frame when using a good algorithm.

For the wireless channels we considered, the target number of bits varies from frame to frame with channel EDR. The visual quality may change rapidly between two consecutive frames if we only adjust Q to achieve the target number of bits. This is undesirable. Here we propose a new bit assignment control scheme which not only adjusts Q but also the number of forced update macroblocks (FUMBs) in a frame. Allowing the number of FUMBs to change based on the target number of bits can reduce the variance in Q and thus improve the uniformity of the picture quality. For adjusting Q, we use the iterative algorithm because:

(1) Some macroblocks are periodically forced to intra-mode in order to increase coding robustness. R-Q curves may change from frame to frame and it is difficult to find an accurate global model.

(2) We wish to maintain a tight rate control and relatively uniform quality within a frame.

(3) For low bitrate wireless video, the bandwidth is a precious resource whereas computational power is not as scarce.

Of course, the simple prediction algorithm can also be used here. Next we define several parameters and then describe the proposed bit assignment control algorithm.

- H_i: bit assignment for the headers of the current frame for the ith iteration including picture headers, GOB headers, and macroblock headers (minor changes may occur with Q);
- F_i: total bit assignment of the current frame for the ith iteration;
- F_{t}: target number of bits for the current frame obtained from 3.1;
- Q_i: quantization scale of current frame for the ith iteration;
- $\overline{Q}_{\text{pre}}$: average Q of the previous frame.

FUMBs are used incrementally in a frame in order to stop possible error propagation. $M_{\max}$ defines the maximum number of FUMBs per frame and $M_{\min}$ the minimum number of FUMBs per frame. The bit assignment algorithm is as follows:

(1) A frame is first coded using the average Q of the previous frame ($Q_0 = \overline{Q}_{\text{pre}}$). The bit assignment of the header and every macroblock is counted. The bits of the $M_{\max}$ would-be FUMBs are counted for two different coding modes: (a) inter-mode, (b) intra-mode. Then based on the target number of bits, we choose the number of FUMBs M_{c} ($M_{\min} \leqslant M_{\text{c}} \leqslant M_{\max}$) which gives a bit match that is the closest to the target. The resultant bit number of the encoded frame is F_0.

(2) We change the quantization scale if the target number of bits cannot be obtained when only adjusting the number of FUMBs, i.e., if $|F_{\text{t}} - F_0| > \varepsilon F_{\text{t}}$ ($0 \leqslant \varepsilon \leqslant 1$).

(a) Q_1 can be obtained from the first order R-Q curve,

$$Q_1 = Q_0 \frac{F_0 - H_0}{F_{\text{t}} - H_0}. \tag{7}$$

The frame is then coded using Q_1.

(b) If $|F_1 - F_{\text{t}}| < \varepsilon F_{\text{t}}, Q_1$ is the final quantization scale. If not, the following iteration is used:

$$\begin{aligned} &\text{if } F_i > F_{\text{t}}, \quad Q_{i+1} = Q_i + \delta \\ &\text{else} \qquad\quad\; Q_{i+1} = Q_i - \delta. \end{aligned} \tag{8}$$

The frame is then coded using Q_{i+1} ($i = 1, 2, 3, \ldots$).

(c) The above iteration is performed until we have $|F(Q) - F_{\text{t}}| < \varepsilon F_{\text{t}}$ or $F(Q-1) < F_{\text{t}}$ and $F(Q) > F_{\text{t}}$, where $F(Q)$ is the bit assignment for the quantization scale Q. If $|F(Q) - F_{\text{t}}| < \varepsilon F_{\text{t}}$, Q is used for the quantization scale. Otherwise, some MBs use $Q-1$ and others use Q so that the coded frame size F satisfies the condition $|F - F_{\text{t}}| < \varepsilon F_{\text{t}}$. Obviously the MBs with least visual impact, such as the edges, use Q.

Our experiments show that only two or three iterations are required for $\varepsilon = 5\%$. Once more, we can set a minimum quantization scale $Q_{\min}$ in order to guarantee the quality. Here we let $Q_{\min}$ be the minimum number that is allowed by the standard, i.e., 31. It is possible that the encoded frame size $F(Q_{\min})$ is greater than the target F_{t} even if $Q_{\min}$ is used. For this case, as before, we have two options: (1) using $Q_{\min}$ to encode the frame, (2) skipping this frame. If $F(Q_{\min})$ causes the overflow of the physical buffer, we skip the frame. Otherwise, we encode the frame using $Q_{\min}$ in hope that the channel will become good.

4. Performance results

We have developed the adaptive source rate control (ASRC) scheme. In this section, we present performance results of ASRC. The two most important parameters of the ASRC scheme are (1) the window size W, and (2) the parameter κ. We will first investigate the effect of these two parameters on the performance of ASRC and choose the appropriate values for the two parameters.

Real-time wireless services depend on the communication environments and the delay requirements. In this section, the performance of ASRC is studied under the different wireless communication environments and delay constraints. We also compare the performance of ASRC with that of conventional static constant bit rate (CBR) video transmission. CBR video transmission is considered for the following reasons: (1) CBR is widely used for video conferencing in wired networks because high motion and scene changes are rare for video conferencing applications and the quality of CBR video is acceptable; (2) most of the source rate control schemes were developed to realize CBR.

For the simulation we assume that the video is transmitted over a TDMA radio network. For a TDMA channel, time is divided into slots, where each slot is equal to the packet transmission duration. The transmitter and the receiver are synchronized with the slotted channel. In our system, the user data rate is 32 kb/s, the channel transmission rate is 2 Mb/s, and the average channel SNR is 20 dB. The total number of bits in a packet is 420, consisting of 400 data bits and 20 CRC bits. The data bits may be the original information data or the parity generated by the RS code. An shortened RS code $(8, 4, 2)$ with 4 bits per symbol is chosen as code C_1 in the hybrid ARQ scheme since this RS code is very easy to decode. The ARQ timeout is equal to the round-trip delay and the one-way channel delay is a half of the round-trip delay. A fading simulator based on Jack's model is used to simulate the radio channel [9]. The simulator generates Rayleigh distributed envelop for the received signal summing the output of several low-frequency oscillators with uniformly distributed phases. For the simulation, 30 oscillators are used to obtain the Rayleigh fading signal. DPSK is used as the modulation format with a carrier frequency of 1.9 GHz.

Simulation was carried out on the QCIF "Mother and Daughter" video sequence which contains typical video conference-like images and the frame interval is about 67 ms (15 frames/s). The encoded sequence with error protection is transmitted 40 times using different starting points in the fading simulator. Note that we do not assume that the frame interval is a multiple of the packet transmission interval. This is more realistic. Five or six packets may be transmitted during a frame interval in our experiments. The transport buffer is used to compensate for the difference in the transmission rate during a frame interval.

For the ASRC video transmission, the first frame is intracoded using a quantization scale of 16. After the first frame is sent out, the proposed adaptive source rate control scheme is employed. Maximum 6 macroblocks can be forced update macroblocks in a frame which is about a half of a GOB. The entire encoded video frame is assumed to be sent to the transport buffer by the encoder and taken from the receiver buffer by the decoder instantaneously. The frame delay is the interval from the time instant that a video frame is sent to the transport buffer to the time instant that it is completely received at the receiver. This includes the buffer delay, the transmission delay, the propagation delay and the ARQ delay. Video encoding and decoding delay are not considered here because they are processor related and we assume the processors are fast enough to handle encoding and decoding. The frame delay should be less than the frame delay bound which the application requires. Otherwise the frame is in error. The video data is transmitted using the hybrid ARQ scheme. If it cannot arrive at the receiver before the frame delay bound, the data is discarded at the transmitter. At the receiver, all received video data will be sent to the video decoder for decoding at its scheduled playout time (the scheduled playout time for a frame is the time at which the frame was sent into the transport buffer plus the frame delay bound).

4.1. Effect of window size and parameter κ

We first study the impact of the window size W on the performance of ASRC given a fixed parameter κ. An important measurement of the performance is the frame error rate (FER). If a frame cannot correctly arrive at the receiver before a given frame delay bound, we say that this frame is in error. Figure 2 shows the FER results as a function of window size for three values of the frame delay bound (FDB): 200 ms, 250 ms and 300 ms. The mobile speed is 2 km/h, the RTD is equal to 13 ms and κ is 10. In order to get some insights, in figures 3–6, we present the number of information bits that the channel can correctly transmit during a frame interval, the encoded frame size, the transport buffer occupancy and the frame delay when the window size is equal to 20 and 100, respectively. In figure 3, we see that there is a periodical fluctuation in the number of information bits correctly transmitted during a frame interval. This is because the frame interval is not a multiple of the packet transmission interval. Five or six packets may be transmitted during a frame interval in our experiments. The proposed ASRC scheme does not compensate for this normal periodic fluctuation. It only adjusts the source rate based on the channel conditions.

In figure 2, we notice that for a given frame delay bound, the FER decreases with decreasing window size. This is due to the fact that with a small value of W, the algorithm can quickly adjust the target number of bits to respond to the change in the channel EDR. We can see in figures 3–6 that the smaller window size brings the target number of bits down more quickly to compensate for the lower EDR due to retransmissions when a channel fade occurs so the buffer occupancy and delay are smaller. However this also results in a larger variance in the source rate. A larger window size gives a smoother target number of bits for each frame, but the buffer occupancy and delay are lager during a fading period. When the frame delay bound is small, the window size W must be small for quick response of the source rate control to the change in channel conditions. When the frame delay bound is large, we can increase the window size W to obtain more uniform allocation of the source rate and keep the FER low.

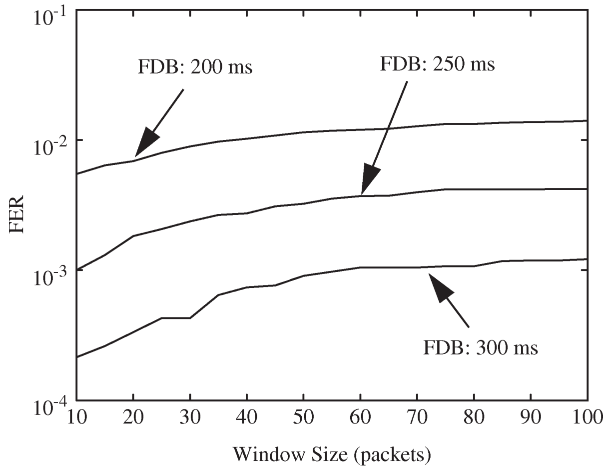

Figure 2. FER versus window size.

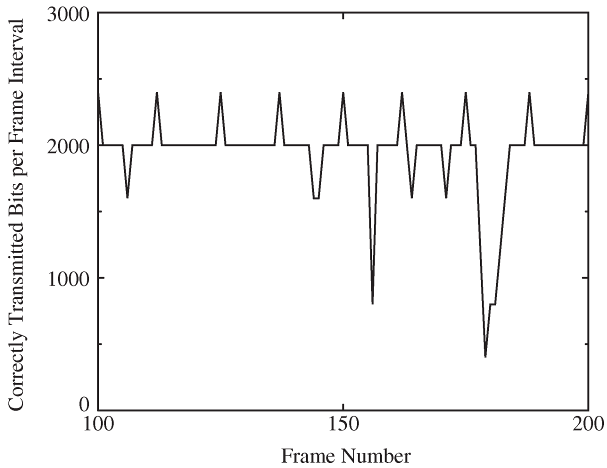

Figure 3. Number of information bits that the channel can correctly transmit during a frame interval (67 ms).

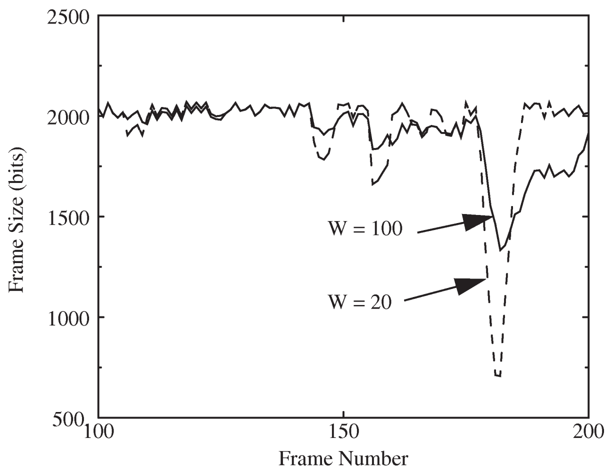

Figure 4. Encoded frame size with different window sizes.

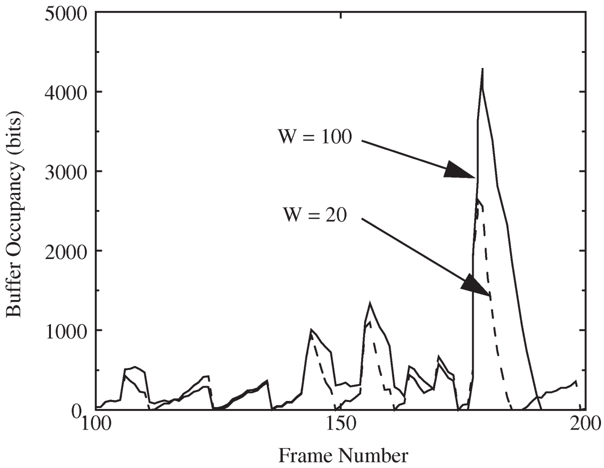

Figure 5. Buffer occupancy with different window sizes.

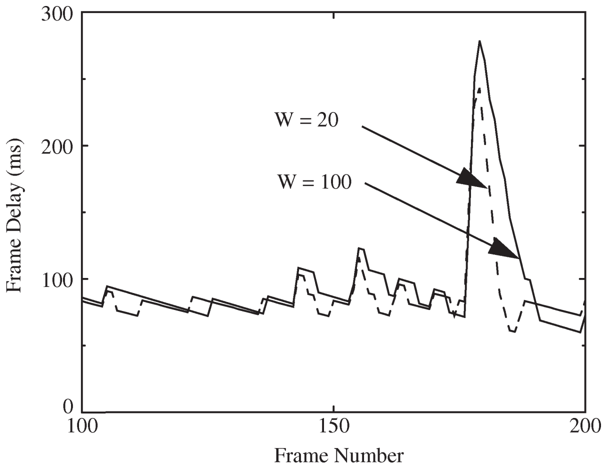

Figure 6. Frame delay with different window sizes.

Next we fix the window size and let κ vary over a reasonable range. We choose the value of W to be $\lfloor (D_{\max} - D_0)/I_{\mathrm{p}} \rfloor$ packets, where $\lfloor a \rfloor$ is the maximum integer equal to or less than a. This represents that the window size becomes larger as the frame delay bound increases.

Figure 7 shows the FER results as a function of κ for three values of frame delay bound: 200 ms, 250 ms and 300 ms. The mobile speed is 2 km/h and the RTD is equal to 13 ms. We notice that FER decreases as κ becomes smaller. This is because κ controls how quickly the buffer occupancy approaches B_{tar} in equation (3). When κ is smaller, it converges faster but may cause larger variance in the source rate. A larger κ results in less variance in the source rate, but at the cost of a longer period for changing the buffer occupancy, which may result in a larger buffer buildup and a longer delay.

So far we have observed the effects of W and κ. When the frame delay bound is large, we can increase the window size W and/or the parameter κ to obtain less variance in the source rate. When the frame delay bound is small, the window size W and the parameter κ must be small for quick response of the source rate control to the change of channel conditions. In later subsections, we will study the effect of

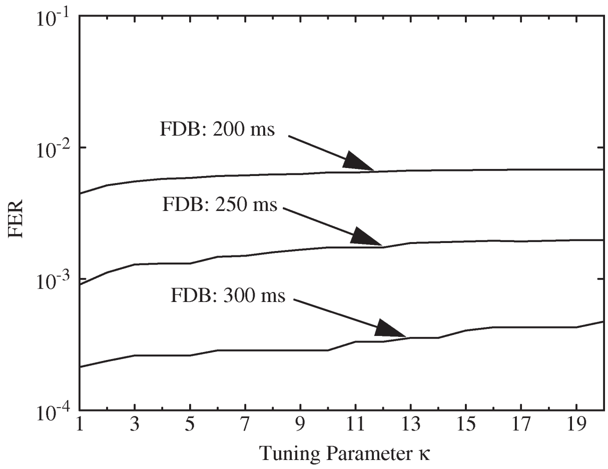

Figure 7. FER versus parameter κ.

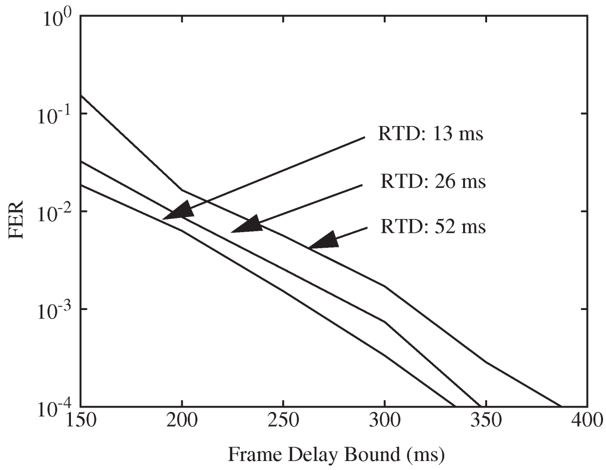

Figure 8. FER versus frame delay bound for ASRC over a slow fading channel.

the delay bound and the transmission characteristics on the performance by setting

$$W = \lfloor (D_{\max} - D_0)/I_{\mathrm{p}} \rfloor \quad \text{and} \quad \kappa = \lfloor (D_{\max} - D_0)/T_{\mathrm{tar}} \rfloor.$$

4.2. *Performance of adaptive source rate control scheme in a slow fading environment*

We first study the performance of ASRC in a slow fading environment. For slow fading, we set the mobile speed to be 2 km/h. For comparison, we also present the results of a static constant bit rate (CBR) video transmission.

Figure 8 shows the FER versus the frame delay bound for ASRC. Three ARQ round-trip delay values of 13 ms, 26 ms and 52 ms are considered, which correspond to 1 packet transmission interval, 2 packet transmission intervals and 4 packet transmission intervals for a user in a TDMA network, respectively. We also would like to know what throughput can be achieved using ASRC. The throughput is defined as the ratio of the average source rate over the channel rate. Figure 9 shows the throughput results. We can make the following observations:

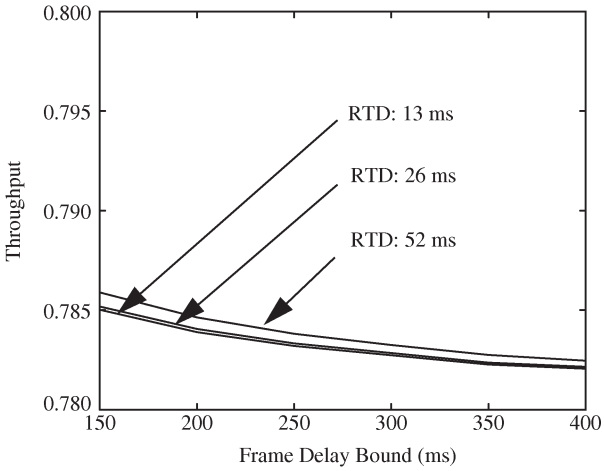

Figure 9. Throughput versus frame delay bound for ASRC over a slow fading channel.

(1) FER decreases with increasing frame delay bound. This is due to the fact that a longer frame delay bound allows more retransmissions. The throughput only slightly decreases as the frame delay bound increases. However for video conferencing, the frame delay bound must be less than 400 ms and it is preferable that the frame delay bound be below 200 ms [1].

(2) FER is greater with higher RTD. This is because it takes a longer time for retransmissions. For a given frame delay bound, the allowed number of retransmissions is smaller with higher RTD. The throughput does not change much as the RTD changes.

We would like to compare the performance of ASRC with that of conventional CBR. For CBR video, the first frame is also intracoded using a quantization scale of 16. After the first frame is sent out, the open-loop CBR source rate control is applied. Each frame is assigned a fixed target bit number, then the quantization scale is adjusted according to the iteration algorithm. The encoded bit number is very close to the target bit number and the small deviation is compensated for by slightly adjusting the target bit number for the later frames. The number of forced update macroblocks used in a frame is set to be the average number used in ASRC.

CBR generates a near-to-uniform number of bits for each frame (source rate). The source rate (throughput) determines the FER performance under certain delay requirements and channel conditions. Figure 10 presents FER versus throughput under various frame delay bounds for the above three RTD values. We notice that for a given frame delay bound (FDB), FER becomes larger with increasing throughput. Moreover, FER increases as the frame delay bound becomes smaller. When the frame delay bound is small (for example, 200 ms), the frame error rate is quite high with a large throughput. It is impossible to achieve a low FER and a high throughput simultaneously. For example, when the frame delay bound is 200 ms and the RTD is 13 ms, ASRC achieves a FER of 0.0063 with a throughput of 0.784. However, for CBR, the FER value is 0.36 when the throughput is 0.78. The throughput is only 0.58 to get a FER value of 0.0063. We have similar results with other values of RTD when the frame delay bound is low. The reason for this behavior can be explained as follows: for the slow fading channel, the error pattern is very bursty. The channel is in a good state (no retransmission is necessary) most of time so the EDR is high, but the EDR dramatically drops due to retransmissions during a long deep fade. In other words, the retransmissions result in a very bursty

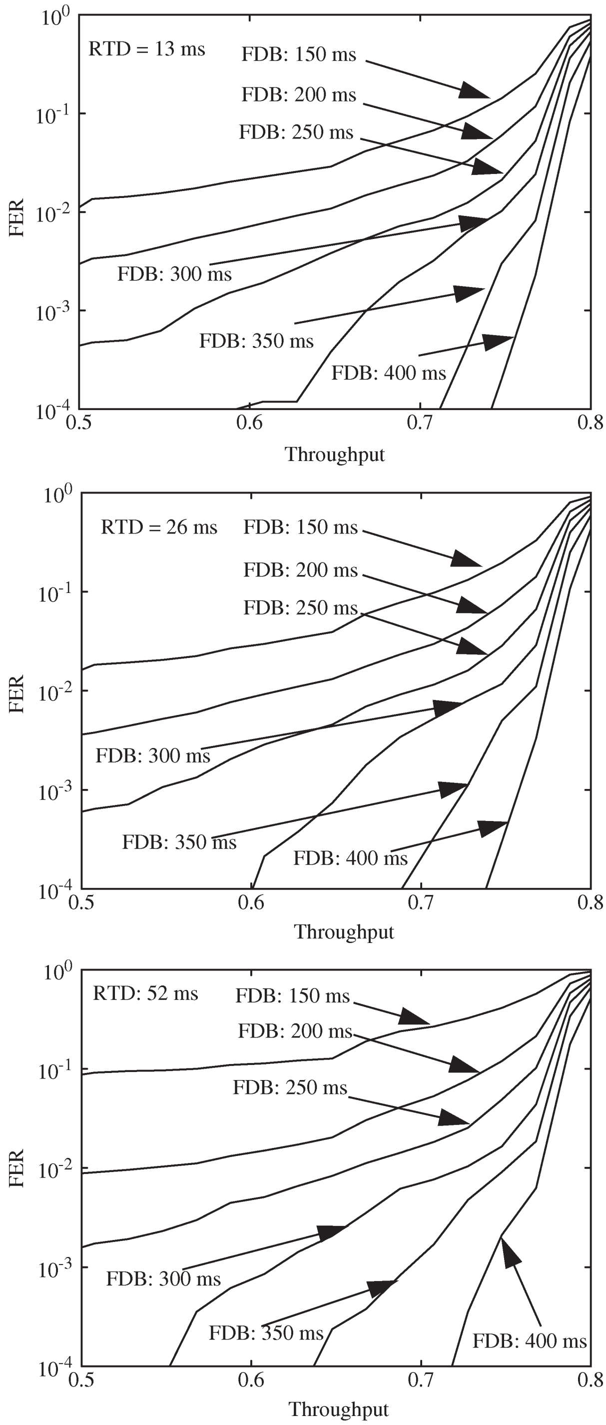

Figure 10. FER for CBR over a slow fading channel.

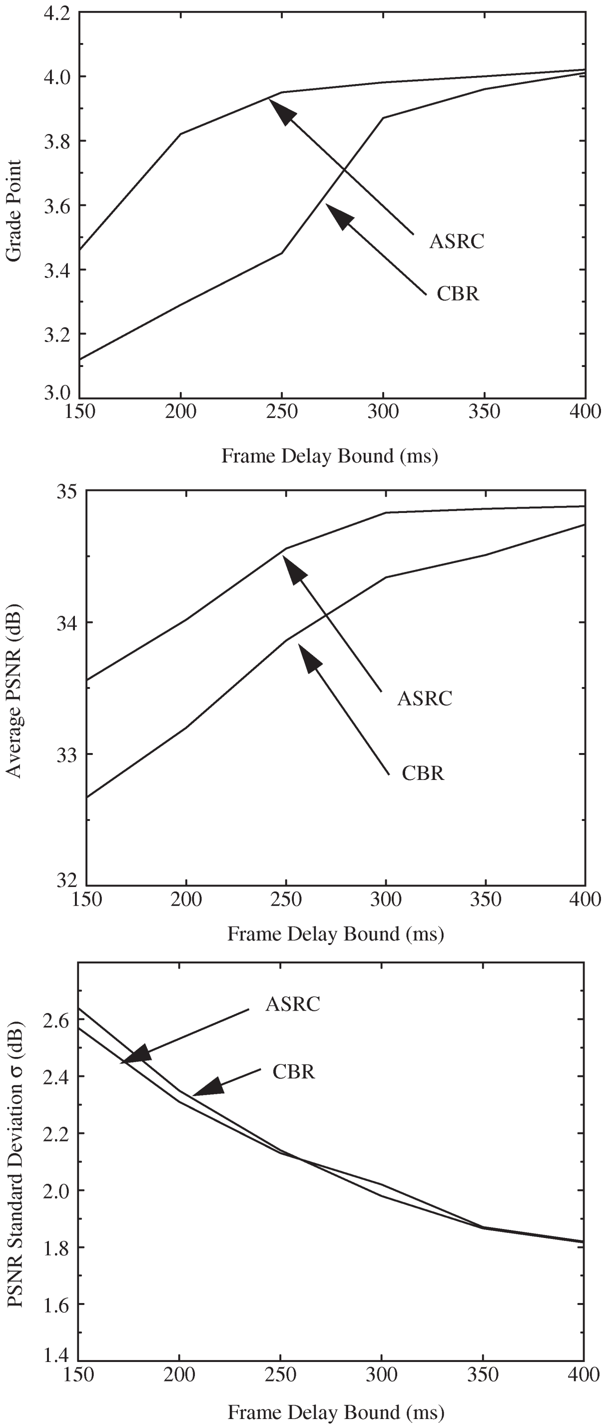

Figure 11. Video quality over a slow fading channel.

EDR when the channel fading is slow. CBR video does not consider the channel conditions. The traffic produced by a CBR video source is near to constant, then when it is transmitted over a channel with bursty EDR, it suffers a high FER if the source rate is high. On the other hand in order to obtain a low FER, it must always keep a very low source rate. ASRC adapts the source rate to the effective channel rate. It can therefore achieve a low FER and a high throughput.

In order to compare the video quality of CBR and ASRC transmission, three metrics are used in this paper: average peak signal-to-noise ratio (PSNR), PSNR standard deviation σ, and an objective video quality assessment scheme based on the human visual system [17] which uses a grade point (GP) system. The average PSNR gives the general quality of a video sequence after transmission, the σ indicates the degree of variation in picture quality in the video sequence, and the GP represents the overall viewing quality of a video sequence as perceived by a human being. The objective video quality assessment scheme gives a grade point ranging from 1.0 to 5.0, with 5.0 meaning excellent quality, 4.0 meaning good quality, 3.0 meaning acceptable quality, 2.0 meaning bad quality and 1.0 meaning absolutely unacceptable quality.

Figure 11 shows the quality of the received ASRC video sequence and the best quality for the CBR video transmission. Based on PSNR and GP, the video quality of ASRC outperforms that of conventional CBR. The smaller the delay bound, the larger performance difference between ASRC and CBR. This is because CBR either suffers from a high FER or results in a very low throughput for slow fading channel. Thus the quality of CBR is greatly degraded. When the frame delay bound increases, not only are more retransmissions allowed but the variance in channel conditions are smoothed out. The performance difference between CBR and ASRC becomes smaller. There is not much difference in σ between ASRC and CBR. This proves that the bit assignment algorithm in which the variance in the target number of bits due to the change in the channel conditions is compensated for by adjusting the number of forced update macroblocks within a frame works very well.

We have shown that in the slow fading environment, adaptive source rate control can achieve much better video quality for real-time video transmission. In the next subsection we study the performance of ASRC in fast fading environment.

4.3. Performance of adaptive source rate control scheme in a fast fading environment

For the fast fading channel, we set the mobile speed to be 100 km/h. The error pattern is more random in fast fading environment. The more random the error pattern, the better the performance of the FEC codes. The probability that a packet can not be correctly received after many retransmissions becomes very small. We expect that the FER performance is better in a fast fading environment. Figures 12 and 13 show the FER and the throughput for ASRC with a RTD of 13 ms. FER quickly decreases when the frame delay bound increases. The change in the throughput is very small. Compared to the slow fades, the fast fades result in a much lower FER as expected. The throughput however becomes slightly lower. This is because random errors results in more initially transmitted packets in error so that more packets require retransmissions.

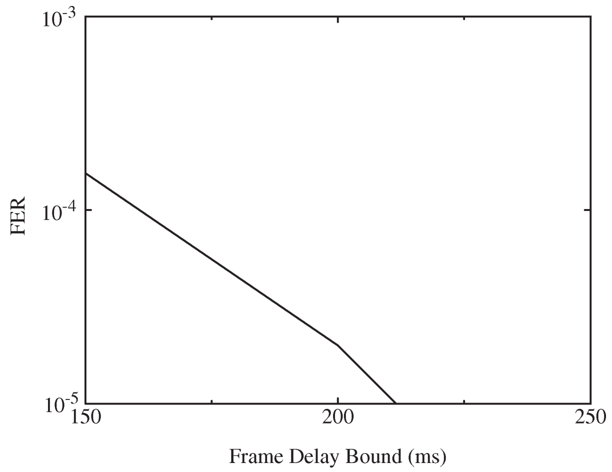

Figure 12. FER versus frame delay bound for ASRC over a fast fading channel.

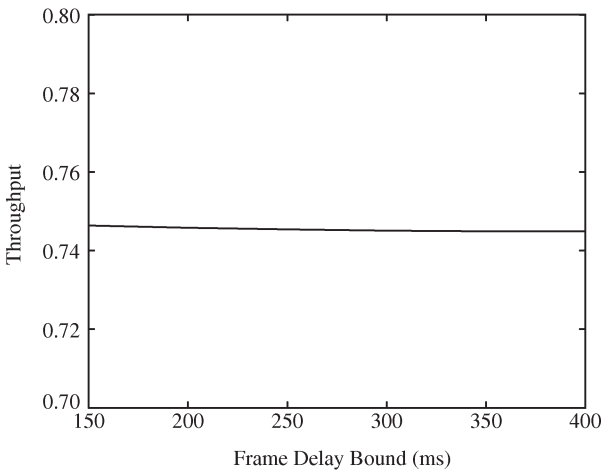

Figure 13. Throughput versus frame delay bound for ASRC over a fast fading channel.

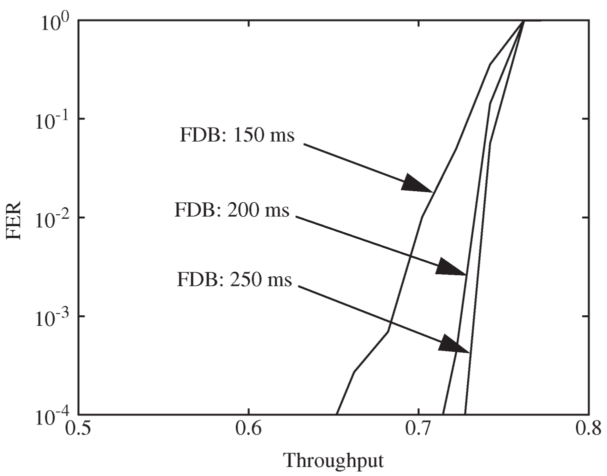

Figure 14. FER for CBR over a fast fading channel.

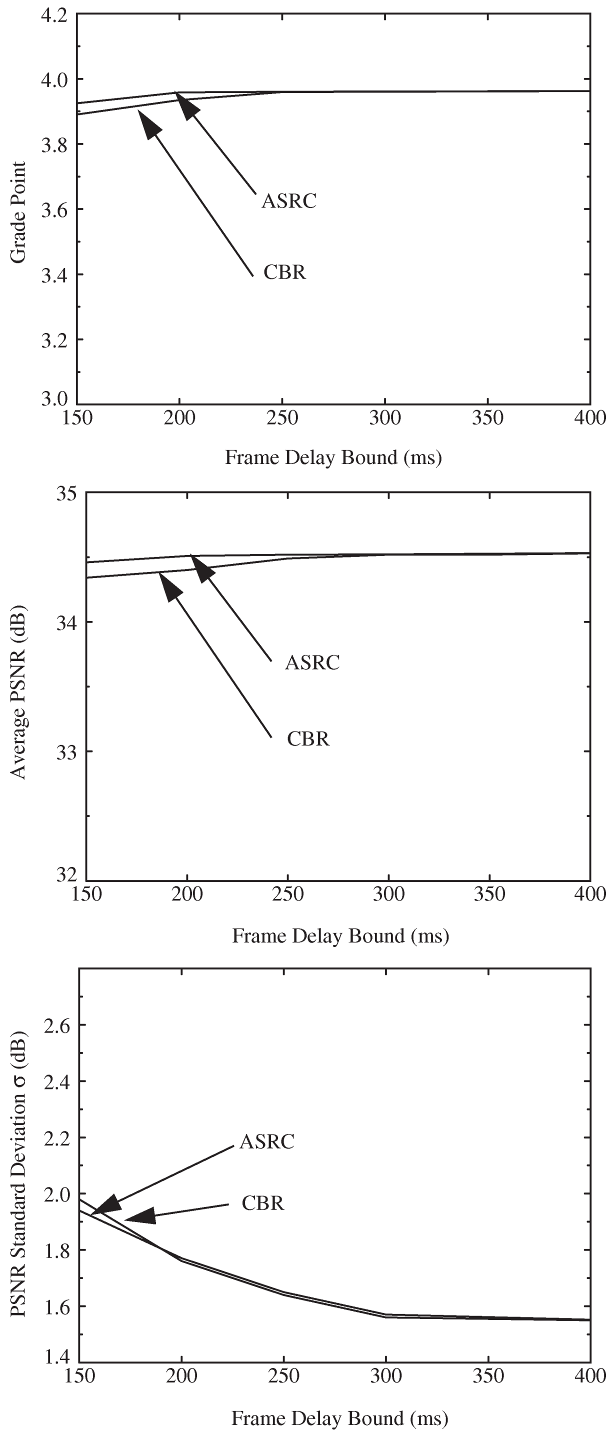

Figure 15. Video quality over a fast fading channel.

The FER of the CBR scheme is shown in figure 14. CBR achieves a much better performance in the fast fading environment than in the slow fading environment because the channel is more random and the effective data rate is more uniform for fast fading. Figure 15 shows the quality of the received ASRC video sequence and the best quality for the CBR video transmission. Both ASRC and CBR can achieve good video quality. The difference in the two approaches is very small.

5. Conclusions

In this paper we proposed an adaptive source rate control scheme and demonstrated that it can work together with the hybrid ARQ error control scheme to achieve efficient transmission of real-time video with low delay and high reliability. The proposed ASRC scheme dynamically allocates the target source rate based on the channel condition, the transport buffer occupancy and the delay constraints so that the available channel bandwidth is efficiently utilized and the encoded video data can correctly be transmitted within the delay bound imposed by the applications. It achieves the allocated target source rate by adjusting both the number of the forced update macroblocks and the quantization scale. The number of the forced update macroblocks used in a frame is first adjusted so that the fluctuation of the quantization scale with the change in the channel conditions during encoding is reduced and the uniformity of the video quality is improved. The simulation results showed that the proposed ASRC scheme performs very well for both slow fading and fast fading channels.

References

[1] R. Cox and P. Kroon, Low bitrate speech coders for multimedia communication, IEEE Commun. Mag. 34(12) (December 1996) 34–41.

[2] W. Ding and B. Liu, Rate control of MPEG video coding and recording by rate-quantization modeling, IEEE Trans. Circuits and Systems for Video Tech. 6(1) (February 1996) 12–20.

[3] ISO-IEC/JTC1/SC29/WG11, MPEG2 Test model 5 Draft (April 1993).

[4] ISO-IEC/JTC1/SC29/WG11, MPEG-4 syntax description language specification (MSDL), version 1.3 (September 1996).

[5] ITU-T, H.263 TMN 5 (1995).

[6] ITU-T Recommendation H.324, Terminal for low bitrate multimedia communication (November 1995).

[7] ITU-T Draft Recommendation H.223/Annex A, Multiplexing protocol for low bitrate mobile multimedia communication (November 1996).

[8] ITU-T Recommendation H.263, Video coding for low bitrate communication (1996).

[9] M. Jeruchim, P. Balaban and K. Shanmugan, *Simulation of Communication Systems* (Plenum Press, New York, NY, 1992).

[10] M. Khansari, A. Jalali, E. Dubois and P. Mermelstein, Low bit-rate video transmission over fading channels for wireless microcellular systems, IEEE Trans. Circuits and Systems for Video Tech. 6(1) (February 1996) 1–11.

[11] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[12] H. Liu and M. El Zarki, Performance of video transport over wireless networks using hybrid ARQ, in: *Proc. of ICUPC '96*, Boston (October 1996).

[13] W. Luo and M. El Zarki, Quality control for MPEG-2 video transmission over ATM-based networks, to appear in IEEE JSAC (1997).

[14] J. Padgett, C. Gunther and T. Hattori, Overview of wireless personal communications, IEEE Commun. Mag. 33(1) (January 1995) 28–41.

[15] D. Raychaudhuri, Wireless ATM networks: architecture, system design and prototyping, IEEE Personal Commun. 3(4) (August 1996) 42–49.

[16] L. Wang, Bit rate control for hybrid DPCM/DCT video codec, IEEE Trans. Circuits and Systems for Video Tech. 4(5) (October 1994) 509–517.

[17] A. Webster, C. Jones, M. Pinson, S. Voran and S. Wolf, An objective video quality assessment system based on human perception, in: *Proc. Human Vision, Visual Processing and Digital Display TV*, Vol. 1913 (SPIE, San Jose, CA, February 1993).

[18] H. Xie, R. Yuan and D. Raychaudhuri, Data link control protocols for wireless ATM access channels, in: *Proc. ICUPC '95*, Tokyo, Japan (November 1995).

Hang Liu received the B.S. from Tianjin University, China, in 1985, and the M.S. from the University of New Orleans, New Orleans, LA, in 1992. He is currently a Ph.D. candidate in the Department of Electrical Engineering, University of Pennsylvania. His research interests include video compression and communications, wireless networking, ATM based networks and digital signal processing.

Magda El Zarki received the B.E.E. degree from Cairo University, Egypt, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, in 1981 and 1987, respectively. She worked from 1981 to 1983 as a communication network planner in the Department of International Telecommunications at Citibank in New York. She joined Columbia University in 1983 as a research assistant in the Computer Communications Research Laboratory where she was involved in the design and development of an integrated local area network testbed called MAGNET. In 1988 she joined the faculty of the Department of Electrical Engineering of the University of Pennsylvania, teaching courses and conducting research in the field of telecommunications, where she currently serves as an associate professor. She also holds a secondary appointment in the Department of Computer and Information Sciences. In January 1993, she was appointed as a part-time professor of Telecommunication Networks in the Faculty of Electrical Engineering at Delft University of Technology, in Delft, The Netherlands. Dr. El Zarki is a member of the ACM, IEEE and Sigma Xi. She is actively involved in many of their sponsored conferences and journals. She was the Technical Program Chair of IEEE INFOCOMM '94.