Low-Delay and Error-Robust Wireless Video Transmission for Video Communications

Tu-Chih Wang, Hung-Chi Fang, and Liang-Gee Chen, *Fellow, IEEE*

***Abstract*—Video communications over wireless networks often suffer from various errors. In this paper, a novel video transmission architecture is proposed to meet the low-delay and error-robust requirement of wireless video communications. This architecture uses forward-error-correction coding and automatic repeat request (ARQ) protocol to provide efficient bandwidth access from wireless link. In order to reduce ARQ delay, a video proxy server is implemented at the base station. This video proxy not only reduces the ARQ response time, but also provides error-tracking functionality. The complexity of this video proxy server is analyzed. Experiment shows that about 8.9% of the total macroblocks need to be transcoded under a random-error condition of 10^{-3} error probability. Because H.263 is the most popular video coding standard for video communication, we use it as an experiment platform. A data-partition scheme is also used to enhance error-resilience performance. This architecture is also suitable for various motion-compensation-based standards like H.261, H.263 series, MPEG-1, MPEG-2, MPEG-4, and H.264. For "Foreman" sequence under a random-error condition of 10^{-3} error probability, luminance peak signal-to-noise ratio decreases only 0.35 dB, on average.**

***Index Terms*—Error resilience, H.263, video communications, video proxy, wireless video.**

I. INTRODUCTION

MANY popular networks cannot provide a guaranteed quality of service (QoS). For example, wireless networks always suffer from different kinds of fading and multipath interference. Thus, packet loss or delay is inevitable. With advances in communication technology, we can expect that video communications will be made available in the near future. But one of the major difficulties in achieving wireless video communication is that compressed video bitstream is very sensitive to errors. Because a compression algorithm often uses variable-length coding (VLC) codes, errors affect not only the symbol located at the error point, but also the succeeding symbols. The motion-compensation (MC) procedure also propagates errors and makes video quality unacceptable. Video communication standards like H.261 [1] and H.263 [2] define some methods for dealing with errors. But the assumptions of error conditions in H.261 and H.263 are under a wired network (ISDN for H.261, PSTN for H.263). The error probability in wireless networks is usually higher than that in a wired network. Consequently, the techniques described in the standards often fail in a wireless network, even if they perform well in a wired network.

There have also been many previous papers trying to solve this problem. On the transmission layer, channel coding techniques like forward error correction (FEC), automatic repeat request (ARQ) [8] try to maximize the throughput of correct packets under a specific channel error condition. On the source-coding layer, error-resilience and error-concealment techniques are used to reduce the damage of errors.

Error-resilience coding techniques include data partition [10], synchronization marker [11], reversible variable-length codes (RVLC) [12], [13], error resilience entropy coding (EREC) [14], multiple-description coding (MDC) [15], etc. These techniques insert redundancy into the bitstream or reorder the symbols to increase the video quality in the error-prone environment.

FEC is widely used in communication to detect and correct errors. It is proved to be efficient if the type of errors is known and the errors do not exceed the maximum correction capacity of FEC. For example, H.261 and H.263 use a (511, 493) Bose–Chaudhuri–Hochquenghem (BCH) FEC checksum which can correct 2 bits of random errors per packet [3]. One problem of FEC is that it cannot efficiently handle burst errors. Some systems use frame interleaving [9] to solve this problem, but a frame-interleaving technique will introduce a large delay, which is not suitable for real-time video communication systems.

FEC techniques can be enhanced by taking into account the importance of symbols in different locations of video bitstream. According to the contribution of each kind of symbols to the peak signal-to-noise ratio (PSNR), important symbols are protected by applying more FEC checksum bits. This is called unequal error protection (UEP) [16], [17].

The ARQ technique is also widely used in communication. This technique can efficiently recover packet loss and burst errors. Transmitters must have an ARQ buffer to hold the sent-out packets until the receiver acknowledges receipt of correct data. If the data are corrupt, the transmitter will resend the data packet until the delay constraint cannot be held. ARQ and FEC techniques can be combined to get a higher data throughput under a predetermined error condition. The main problem of ARQ is that it needs a feedback channel. Thus, broadcast applications like digital TV broadcasting cannot use this technique. Another problem is that retransmission fails when the round-trip delay time is long.

Another technique, called error tracking [18], [19], uses lost packet information from ARQ to track the decoder's behavior and stop error propagation by intra-block update. This technique can reduce error propagation efficiently. The main drawback is

Manuscript received August 15, 2001; revised February 27, 2002. This paper was recommended by Associate Editor W. Zhu.

The authors are with the DSP/IC Design Lab, Graduate Institute of Electronics Engineering, Department of Electrical Engineering, National Taiwan University Taipei 106, Taiwan, R.O.C. (e-mail: eric@video.ee.ntu.edu.tw; honchi@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2002.806807

the large buffer and computation requirement on the encoder side. Moreover, the performance is closely related to round-trip delay.

The data-partition technique is used in MPEG-4 [6] and H.263++ (Annex V). Without data partition, the bitstream syntax is formed at different levels, from picture level to block level. All the information of a macroblock (MB), including motion vectors and block coefficients, is placed nearby. Since errors could cause symbol synchronization problem, any data after the first error are useless. If important data, like motion vectors, can be placed at the front side, the video quality will be much better. Data partition can be applied with ARQ and FEC techniques, since it only reorders the bitstream symbols.

The synchronization marker is used to regain symbol synchronization when error occurs. Traditionally, it is placed at the beginning of the MB rows. MPEG-4 can insert the synchronization marker at the beginning of any MB. It can be optimized to get a better video quality in error-prone environment by using the rate-distortion synchronization marker insertion scheme.

Reversible variable-length codes (RVLCs) are variable-length codes that can be decoded from the opposite side. If an error occurs in the middle of two synchronization points, the decoder can decode from the other side and reduce the size of corrupt data. The RVLC technique is also included in the MPEG-4 standard. The drawback of the RVLC technique is that it often results in longer codewords, and thus reduces compression efficiency. Another constraint of RVLC is that DPCM symbols like motion vector differences (MVDs) cannot use this technique if more than one predictor are chosen.

An error-resilience entropy coding (EREC) technique is used to achieve symbol synchronization at the start of the fixed-length packet. Unlike the synchronization marker, the EREC imposes little overhead on compression efficiency and it is proved to be efficient in wireless environment.

MDC uses multiple video streams to describe a video sequence. If only one stream is received, the decoder can decode an acceptable quality video sequence, but if all streams are received, the decoder can decode a full-quality video. This technique is suitable for a no-feedback channel situation, but the overhead of MDC is quite high and is not suitable for low-bit-rate applications.

Error-concealment techniques [22] can be divided into spatial concealment and temporal concealment. Most of them could be combined with error-resilience techniques to form a better visual quality, but the computational complexity of these techniques would be of great concern on the portable devices.

In [18], a combination of ARQ, FEC, and error tracking is shown to be the best choice of error-resilience tools. We use these error-resilience tools plus data partitioning to enhance the error performance. An error-tracking function operates at the video proxy server at the base station, and other error-resilience tools are implemented at the wireless terminal. This configuration facilitates ARQ operation and reducesthe hardware requirement of the wireless terminal. Error concealment is not considered in this paper, since a new concealment method could be easily implemented on this architecture.

This paper is organized as follows. Network models used in this paper and delay analysis are described in Section II. The low-delay video transmission architecture proposed in this paper is described in Section III. Section IV shows the experimental environment and discusses the experiment result. Finally, a conclusion is given in Section V.

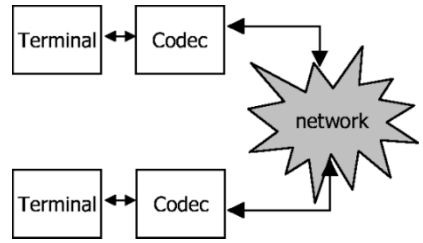

Fig. 1. Traditional network model for point-to-point network.

II. Network Model and Delay Analysis

A. Network Model

Most of the research on error control of video transmission today uses a point-to-point network model. This model is shown in Fig. 1. Two terminals are linked together by a network with errors. Video input goes into the encoder part of codec to form the bitstream and is then transmitted to the network. The video bitstream also comes from the network, is passed to the decoder, and is then displayed on the terminal. In this network model, the network is treated as a black box. The error probability and delay of the network are essential parameters for error simulation. This point-to-point network model fits internet video communications since end-users have no privilege altering the configurations of the network which affect error performance. But in a practical wireless system, the network model can be slightly altered to increase error resilience for video transmission.

In a practical wireless system like the global system for mobile communications (GSM) or code-divison multiple-access (CDMA) system, base stations connected to wireless terminals are built and maintained by service providers. Service providers have the incentive to alter the base stations if better QoS can be achieved. So, we can assume that the base stations can be modified in the wireless network model. The network between base stations is usually more reliable than the wireless link, and the bandwidth is usually larger than the wireless link. Under this assumption, a typical wireless network model is illustrated in Figs. 2 and 3. The gray parts of Figs. 1 and 2 represent network nodes that cannot be changed. Since almost all the errors come from the wireless link, the wired link in Fig. 2 can be viewed as almost error-free. With this model, some alterations can be made at the base stations to increase error resilience and decrease the delay.

B. Delay Analysis

Delay constraint is necessary for real-time video communication system. The total delay from encoder to decoder is given in (1). T_{enc} is the encoding time, T_{net} is the network delay from transmitter to receiver, and T_{buf} is the buffer delay, which is proportional to data in the transmitter buffer and is affected by rate-control algorithm. T_{dec} is the time interval from receiving all data to displaying them on screen, which can also be viewed as the decoder's latency. The last item of (1) is the ARQ delay.

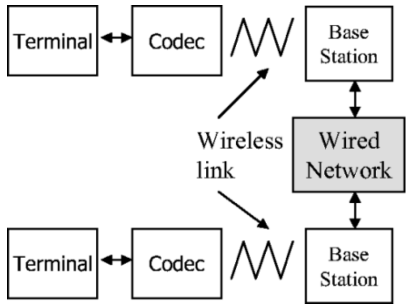

Fig. 2. Practical network situation for wireless video communication.

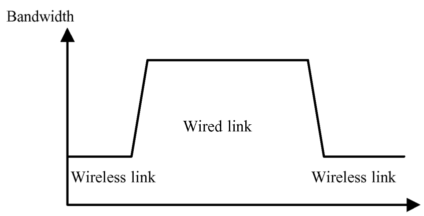

Fig. 3. Bandwidth condition for wireless video communication system.

Since it depends on the error pattern, only an expected value is available. The ARQ delay is the largest part of the total delay if the network condition is bad. Thus, it is important to analyze and reduce the ARQ delay

$$T_{\text{total}} = T_{\text{enc}} + T_{\text{net}} + T_{\text{buf}} + T_{\text{dec}} + T_{\text{ARQ}} \tag{1}$$

$$\begin{aligned} T_{\text{ARQ}} &= p \times T_{\text{round-trip}} + p^2 T_{\text{round-trip}} + \cdots \\ &= \frac{p}{1-p} T_{\text{round-trip}} \end{aligned} \tag{2}$$

The ARQ delay is modeled by (2) [20], where p is the packet-error probability and $T_{\text{round-trip}}$ is the delay penalty for resending the packet. The packet-error probability can be controlled by the FEC technique. Another way to reduce ARQ delay is to decrease $T_{\text{round-trip}}$. $T_{\text{round-trip}}$ is shown in (3). It can be divided into three parts: T_{net} is the net propagation delay, T_{tr} is the transmitter processing delay, and T_{rev} is the receiver processing delay. If the distance between transmitter and receiver is large, T_{net} will dominate $T_{\text{round-trip}}$

$$T_{\text{round-trip}} = 2 \times T_{\text{net}} + T_{\text{tr}} + T_{\text{rev}}. \tag{3}$$

T_{net} can be reduced if an ARQ proxy server is added on the path of transmitter to receiver. The ARQ proxy server stores data from transmitter and respondes to ARQ messages from the receiver by resending data from the local buffer. T_{net} in this network condition will be shortened to become the network propagation delay from receiver to ARQ proxy.

The best location to place the ARQ proxy server is at the base station. It is the nearest point from the wireless terminal; the packet delay from a wireless terminal to the base station is a constant. If an ARQ proxy server is added to the base station, the ARQ delay will no longer be related to the distance between the transmitter and receiver, and the number of times for resending the data under the real-time delay constraint will increase

$$T_{\text{ARQ}} = T_{\text{ARQ}\,w1} + T_{\text{ARQ}\,g} + T_{\text{ARQ}\,w2}. \tag{4}$$

The ARQ delay model of Fig. 2 with ARQ proxy at the base station is now changed to (4). The total ARQ delay is cut into three parts: $T_{\text{ARQ}\,w1} + T_{\text{ARQ}\,w2}$ is the ARQ delay of wireless links and $T_{\text{ARQ}\,g}$ is the ARQ delay of the wired link at ground. The ARQ delay terms in (4) follow (2) with different packet-error probabilities. The relation of packet-error probabilities and time delay between the point-to-point network model and ARQ proxy model is listed in (5) and (6). The approximation of (5) holds when the error probabilities are small. In a practical case, p_g is small enough to make good approximation of (5)

$$\begin{aligned} p &= p_{w1} + p_g + p_{w2} - p_{w1}p_g - p_g p_{w2} \\ &\quad - p_{w2}p_{w1} + p_{w1}p_g p_{w2} \\ &\approx p_{w1} + p_g + p_{w2} \end{aligned} \tag{5}$$

$$T_{\text{net}} = T_{w1\,\text{net}} + T_{g\,\text{net}} + T_{w2\,\text{net}}. \tag{6}$$

Assume that T_{tr} and T_{rev} are small enough compared with T_{net}. The ARQ delay of (4) is

$$T_{\text{ARQ}} = \frac{2p_{w1}}{1-p_{w1}} T_{w1\,\text{net}} + \frac{2p_g}{1-p_g} T_{g\,\text{net}} + \frac{2p_{w2}}{1-p_{w2}} T_{w2\,\text{net}} \tag{7}$$

It is much smaller than that in the original point-to-point model. For example, if the packet-error rates of both wireless links are 2%, and the packet-error rate of the wired link is 10^{-6}, then $p_{w1} = 2 \times 10^{-2}, p_g = 1 \times 10^{-6}, p_{w2} = 2 \times 10^{-2}$. The packet-error rate of 2% is large for wireless link if proper FEC code is used. The packet-error rate of the wired link in today's technology is usually less than 10^{-6} if there is no congestion node on the network. The delay of the wireless link is set to be 10 ms, which is two times of the frame length in a typical PHS system [20]. The wired network delay is assumed to be 200 ms. From the assumption above, we set $T_{w1\,\text{net}} = 10$ ms, $T_{g\,\text{net}} = 200$ ms, and $T_{w2\,\text{net}} = 10$ ms. The average ARQ delay of the proxy model is 0.82 ms, which is calculated by (7). For the same situation as above, the average ARQ delay of point to point model is 18.33 ms as obtained from (2), where the packet-error rate p, is calculated by (5) and $T_{\text{round-trip}}$ is assumed to be $2T_{w1\,\text{net}} + 2T_{g\,\text{net}} + 2T_{w2\,\text{net}}$. The point-to-point model delay is 22 times larger than the proxy model. Note that this value is only an expectation value. Delay in the worst case will be much larger than this value.

III. Low-Delay Video Transmission

A. Architecture

From the delay analysis of Section II, we proposed a low-delay video transmission architecture. A proxy server is located at the base stations for handling ARQ requests and tracking errors. It not only reduces delay, but also improves coding efficiency of error recovery if errors occur. The real-time delay constraint is also managed by the proxy server. The power consumption of the wireless terminal will benefit from this architecture because computation load of error control is located at the base stations.

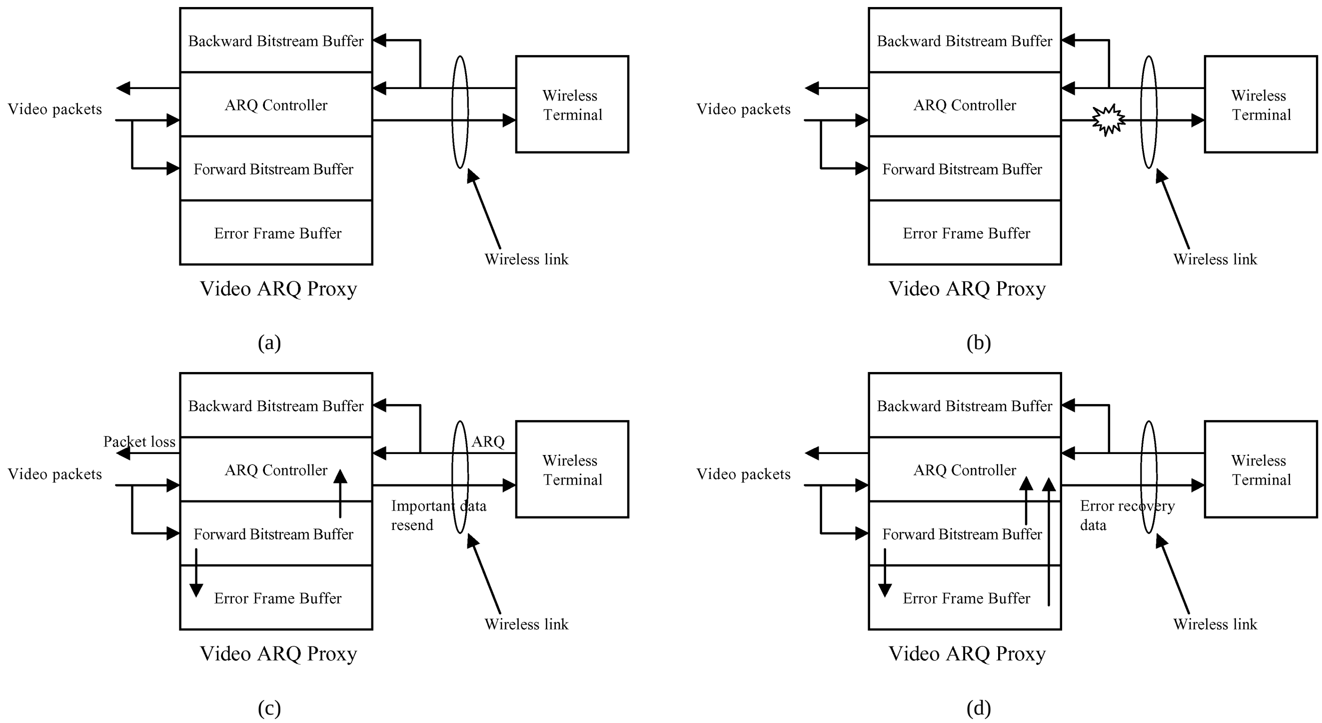

Fig. 4. Illustration of video ARQ proxy behavior. (a) Normal condition. (b) An error occurs on the wireless link. (c) Important data resent and texture residual decoded into error-frame buffer. (d) Recovery of texture errors.

B. Video ARQ Proxy Server

The simple ARQ proxy server discussed in Section II-B only reduces ARQ delay. It cannot achieve optimal performance. Resending data increases not only ARQ delay, but also buffer delay because the rate-control unit on the transmitter side has a slow response to packet loss. A rate-control algorithm which considers packet loss effect could be found in [23]. Although an ARQ proxy can minimize the relation of delay and the distance between transmitter and receiver, the response of rate control on the encoder side is still proportional to the distance. When the distance is large, excess video packets will be stocked on the ARQ proxy server. T_{buf} will dominate the total delay. Then, time slack for ARQ data resending will be reduced and cause failure of the ARQ mechanism.

Another problem with the ARQ proxy server is the symbol dependency of video packets. Not knowing this property results in bandwidth wasting on the wireless link, which is crucial to video quality. For example, if motion vectors were lost due to errors, texture data are useless.

The third problem with the ARQ proxy server is the unequal importance of different kinds of symbols. Resending important symbols instead of unimportant symbols yields better video quality at the decoder side within the same bandwidth constraint. For example, motion vectors have much more peak signal-to-noise ratio (PSNR) contribution than discrete cosine transform (DCT) residual data. If a DCT residual data packet is discarded to support redending a motion vector data packet, the PSNR will be much higher than in the original condition.

For these reasons, an video ARQ proxy server which understands video packet content is the optimal solution to the wireless video transmission problem. The behavior of the video ARQ proxy server is discussed as follows.

1) Video ARQ Proxy Behavior: The behavior of the proposed video ARQ proxy is shown in Fig. 4. If there is no error on the wireless link, the video ARQ proxy acts like a router that routes incoming packets to wireless link, which is illustrated in Fig. 4(a). The forward bitstream buffer stores all incoming packets and drops them after the packet acknowledgment from wireless terminal are received. The video ARQ proxy always parses the incoming video packets to trace their symbol types. The delay introduced by the video ARQ proxy is of one packet length, since the proxy must check whether the incoming packet is correct or not. This delay is small compared with real-time delay constraint in a practical wireless system. For example, a personal handy-phone system (PHS) video communication system described in [20] has a 64-kbit bandwidth on the wired link (ISDN). If a 320-bit packet is used in this system, the delay time introduced by the video ARQ proxy is 5 ms, which is unnoticed to human eyes.

If an error occurs on the wireless link, the wireless terminal will detect this error and send a NACK signal to the video ARQ proxy. The proxy will resend the packet if the bandwidth budget of this frame is enough. Since the data partition scheme is used, important data (MB type, motion vectors, intra-block DC coefficient) located in front of other DCT coefficients will be resent. At the same time, the video proxy server sends a packet loss indication to control the encoder's rate-control unit. This indication only tells the encoder to reduce bit rate by one packet length. It is not a resending request and has no effect on the delay. The reduction in bit rate by one packet length on the encoder side will facilitate error recovery thereafter. The bandwidth loss caused by packet resending is covered by dropping an unimportant packet (DCT coefficients) at the end of this frame, which is shown in Fig. 4(b) and (c). Because the parser of the

TABLE I
SYMBOL BIT COUNTS OF H.263 BITSTREAM. SEQUENCES: FOREMAN 10 FPS, QCIF FORMAT. ENCODING PARAMETERS: $\mathrm{QP(I)} = 13$, INITIAL $\mathrm{QP(P)} = 13$, BIT RATE $= 64$ k

Picture Type	Symbol	Bits
Intra Picture(1)	Header	51
	MCBPC	145
	CBPY & DCT Coefficients	18180
Total		18376
Inter Picture(98)	Header	53
	MCBPC	204
	Vectors	840
	COD	99
	DQUANT	56
	CBPY & DCT Coefficients	4751
Total		6003

video ARQ proxy parses all incoming packets, the video ARQ proxy will know which packet the frame ended on. The dropping action occurs when a new frame symbol is received. This action can eliminate the delay introduced by bandwidth loss, and gather the error together at the bottom of the frame. One restriction of the dropping policy is that important data cannot be dropped. If there is not enough bandwidth to transmit important data, a buffer delay on the video ARQ server is generated. This situation rarely happens since important data (MB type, motion vectors, intra block DC coefficient) are often less than DCT coefficients. The unimportant packet chosen to be dropped from the bitstream buffer is then decoded into the error-frame buffer, which is the same size as the frame buffer.

No matter whether the bit rate on the encoder side is reduced or not, the video ARQ proxy enters an error-recovery phase when the location of the current picture has error on the error-frame buffer. The error in the error-frame buffer is coded and combined with incoming bitstream to form a new bitstream packet. If the bandwidth is not enough to transmit all of the bitstream under the real-time constraint, the bitstream not transmitted will be decoded and added to the error-frame buffer. This error-recovery method will track the errors and get a better video quality, which is illustrated in the experiment section.

The behavior of the video ARQ proxy is slightly different from the mechanism in [18] and [21]. The video ARQ proxy resends important data, even if the real-time constraint cannot be achieved. The data will be useless for display because the real-time constraint can not be held, but it is useful to prevent intra-MB updating, which has less coding efficiency than inter-MB coding. Table I provides the bit counts in a H.263 bitstream. It shows that the bit rate of a P picture is almost one third of a I picture. Thus, preserving motion vectors will result in higher coding efficiency than forcing an intra update on the encoder side.

2) Video ARQ Proxy Functions: The block diagram of the video ARQ proxy server is shown in Fig. 5. Video packets first come into the ARQ controller. If the packet contains motion vectors or headers, the packet is sent out to the wireless link and stores in the bitstream buffer. The packet is decoded by the variable-length code decoder (VLD) unit. Then, motion vectors are passed to the MC unit to compensate error frames. The output of MC is then added to the texture part which is decoded by the

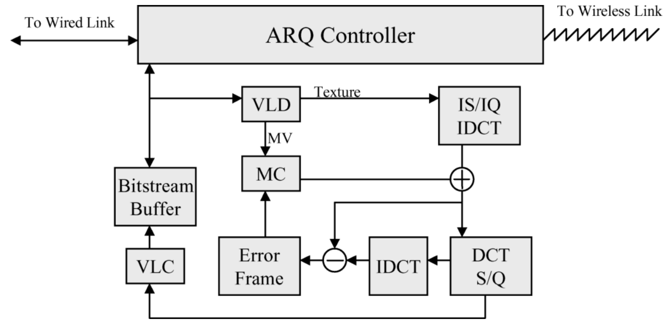

Fig. 5. Block diagram of video ARQ proxy server.

TABLE II
COMPUTATION COMPLEXITY OF A 100 USERS' VIDEO PROXY SERVER

Function	Computation
DCT	38.016 Mpixels/sec
Scan	38.016 Mpixels/sec
Motion Compensation	38.016 Mpixels/sec
Quantization	38.016 Mpixels/sec
Inverse Scan	38.016 Mpixels/sec
Inverse Quantization	38.016 Mpixels/sec
IDCT	76.032 Mpixel/sec
VLD	6.4 Mbit/sec
VLC	6.4 Mbit/sec

VLD/IS/IQ/IDCT path and form the final DCT residual. The residual is then coded by the DCT/S/Q/VLC path to replace texture packet in the bitstream buffer, and sent out to the wireless link. The final error frame is reconstructed by the quantization errors of DCT residual that are not coded by the VLC.

3) Computation Complexity Analysis: The computation complexity is the same as a transcoder without motion estimation, in the worst case. For a base station which supports concurrently 100 users with QCIF picture size, frame rate $= 10$ fps and bit rate $= 64$ kbit/s, the computation complexity is calculated as shown in Table II. DCT and IDCT cores [29], [30] can easily handle pixel rates of up to 100 Mpixels/s in today's technology; 6.4-Mbit/s VLC and VLD are also easily implemented. S, IS, Q, and IQ are simple operations and do not form a bottle neck in the system. A reasonable cost implementation thus could be achieved in today's technology.

In a typical condition when error is not propagated to the whole error-frame memory, the MBs with no error do not need to pass through the whole DCT/IDCT loop. The computation load could be greatly reduced and may be implemented by software if the processor's computation power is enough. Experiment shows that about 8.9% of the MBs need to be transcoded under a random-error condition of 10^{-3} error probability (Channel C2 in Table III). In the burst-error condition of 10^{-3} with a burst length of 5 ms (Channel C5 in Table III), 3.4% of the MBs need to be processed, on average.

4) Memory-Requirement Analysis: The memory requirement is proportional to the number of concurrent online users and picture format. Because the bandwidth is usually very low in wireless communication, the QCIF picture format is most widely used in practical situation. One user must have two frame buffers to store essential information because of MC procedure. Thus, one user occupies $176 \times 144 \times 1.5 \times 2 = 76\,032$

TABLE III
CHANNEL CONDITIONS USED IN THE EXPERIMENTS

Channel	Error Type	BCH parity bits	Packet length	Source rate	PER
C1	1×10^{-4} Random Error	9	511	60.87k	0.126%
C2	1×10^{-3} Random Error	18	511	59.74k	1.5177%
C3	5×10^{-3} Random Error	54	511	55.23k	1.552%
C4	1×10^{-2} Random Error	81	511	51.85k	3.5225%
C5	1×10^{-3} Burst Error, Burst length=5ms	18	511	59.74k	0.51%
C6	1×10^{-3} Burst Error, Burst length=50ms	18	511	59.74k	0.31%
C7	1×10^{-2} Burst Error, Burst length=5ms	81	511	51.85k	4.81%
C8	1×10^{-2} Burst Error, Burst length=50ms	81	511	51.85k	1.88%
C9	WCDMA 64kb,error=1.35×10^{-3}, speed=3km/h	0	640	64k	6.73%
C10	WCDMA 64kb,error=1.26×10^{-3}, speed=40km/h	0	640	64k	4.93%
C11	WCDMA 64kb,error=9.73×10^{-4}, speed=120km/h	0	640	64k	4.61%
C12	WCDMA 64kb,error=8.17×10^{-5}, speed=3km/h	0	640	64k	0.72%
C13	WCDMA 64kb,error=1.21×10^{-4}, speed=40km/h	0	640	64k	0.81%
C14	WCDMA 64kb,error=9.37×10^{-5}, speed=40km/h	0	640	64k	0.69%

TABLE IV
PERFORMANCE [PSNR (dB)] OF FOREMAN SEQUENCE

Channel	PSNR(Y)	PSNR(U)	PSNR(V)
C1	31.45/31.49	36.62/36.63	37.46/37.47
C2	31.08/31.43	36.29/36.59	36.95/37.49
C3	30.77/31.10	36.24/36.43	36.68/37.15
C4	30.07/30.85	35.83/36.30	35.85/36.94
C5	31.34/31.49	36.55/36.59	37.35/37.49
C6	31.29/31.49	36.55/36.59	37.37/37.49
C7	29.64/30.85	35.70/36.30	35.66/36.94
C8	30.32/30.85	36.15/36.30	36.46/36.94
C9	29.03/31.68	35.37/36.76	35.41/37.62
C10	30.62/31.68	36.37/36.76	36.48/37.62
C11	29.48/31.68	35.55/36.76	36.01/37.62
C12	31.30/31.68	36.65/36.76	37.18/37.62
C13	31.58/31.68	36.69/36.76	37.60/37.62
C14	31.38/31.68	36.52/36.76	37.12/37.62

TABLE V
PERFORMANCE [PSNR (dB)] OF CARPHONE SEQUENCE

Channel	PSNR(Y)	PSNR(U)	PSNR(V)
C1	33.68/33.76	38.35/38.36	39.01/39.13
C2	32.75/33.69	38.16/38.34	38.11/39.08
C3	32.53/33.35	37.97/38.14	38.36/38.84
C4	31.65/33.10	37.55/37.93	37.74/38.68
C5	33.47/33.69	38.30/38.34	38.87/39.08
C6	33.50/33.69	38.26/38.34	38.99/39.08
C7	31.47/33.10	37.38/37.93	37.69/38.68
C8	32.47/33.10	37.77/37.93	38.23/38.68
C9	30.55/33.99	37.31/38.49	36.90/39.32
C10	31.97/33.99	37.70/38.49	38.02/39.32
C11	30.89/33.99	37.45/38.49	37.05/39.32
C12	33.05/33.99	38.41/38.49	39.04/39.32
C13	33.42/33.99	38.09/38.49	39.32/39.32
C14	33.65/33.99	38.41/38.49	39.19/39.32

bytes for error-frame buffers. The bitstream buffer and status variables are much less than the error-frame buffers. By a coarse estimation, one user needs 80 kB of memory. The memory requirement of a 100-user proxy is, thus, 8 Mbytes.

5) Memory-Bandwidth Analysis: Memory bandwidth is always a large problem in a digital video system because

TABLE VI
PERFORMANCE [PSNR (dB)] OF GRANDMA SEQUENCE

Channel	PSNR(Y)	PSNR(U)	PSNR(V)
C1	38.53/38.55	41.44/41.45	41.80/41.81
C2	38.17/38.49	41.33/41.41	41.74/41.79
C3	37.82/38.17	41.07/41.16	41.56/41.63
C4	37.28/37.85	40.67/40.98	41.17/41.34
C5	38.38/38.49	41.38/41.41	41.78/41.79
C6	38.29/38.49	41.31/41.41	41.70/41.79
C7	37.07/37.85	40.50/40.98	40.98/41.34
C8	37.39/37.85	40.74/40.98	41.29/41.34
C9	36.93/38.73	40.28/41.57	40.99/42.01
C10	37.86/38.73	41.14/41.57	41.66/42.01
C11	36.73/38.73	39.37/41.57	40.38/42.01
C12	38.25/38.73	41.42/41.57	41.87/42.01
C13	38.57/38.73	41.55/41.57	41.99/42.01
C14	38.51/38.73	41.55/41.57	41.94/42.01

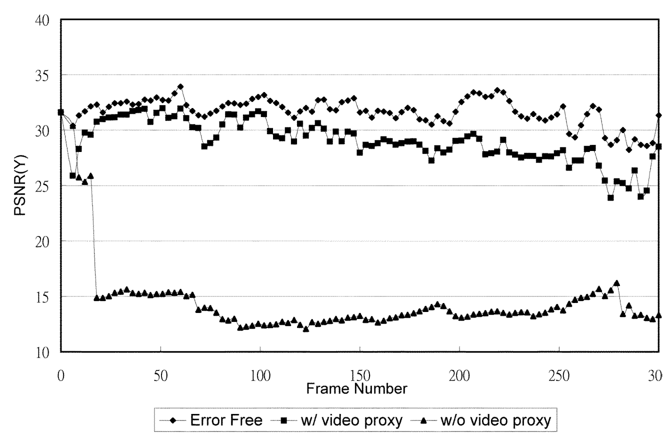

Fig. 6. Performance of the Foreman sequence under channel C9.

the data rate is high. It is also related to the hardware-implementation style. From Table II, VLC and VLD needs a 16 Mbytes/s bandwidth at worst case, while MC needs 38 Mbytes/s. If the IDCT/DCT loop is implemented by hardware or the MB is cached in software implementation, the loop needs only one read and one write action to the memory. In this case, 76 Mbytes/s bandwidth is needed. Total bandwidth

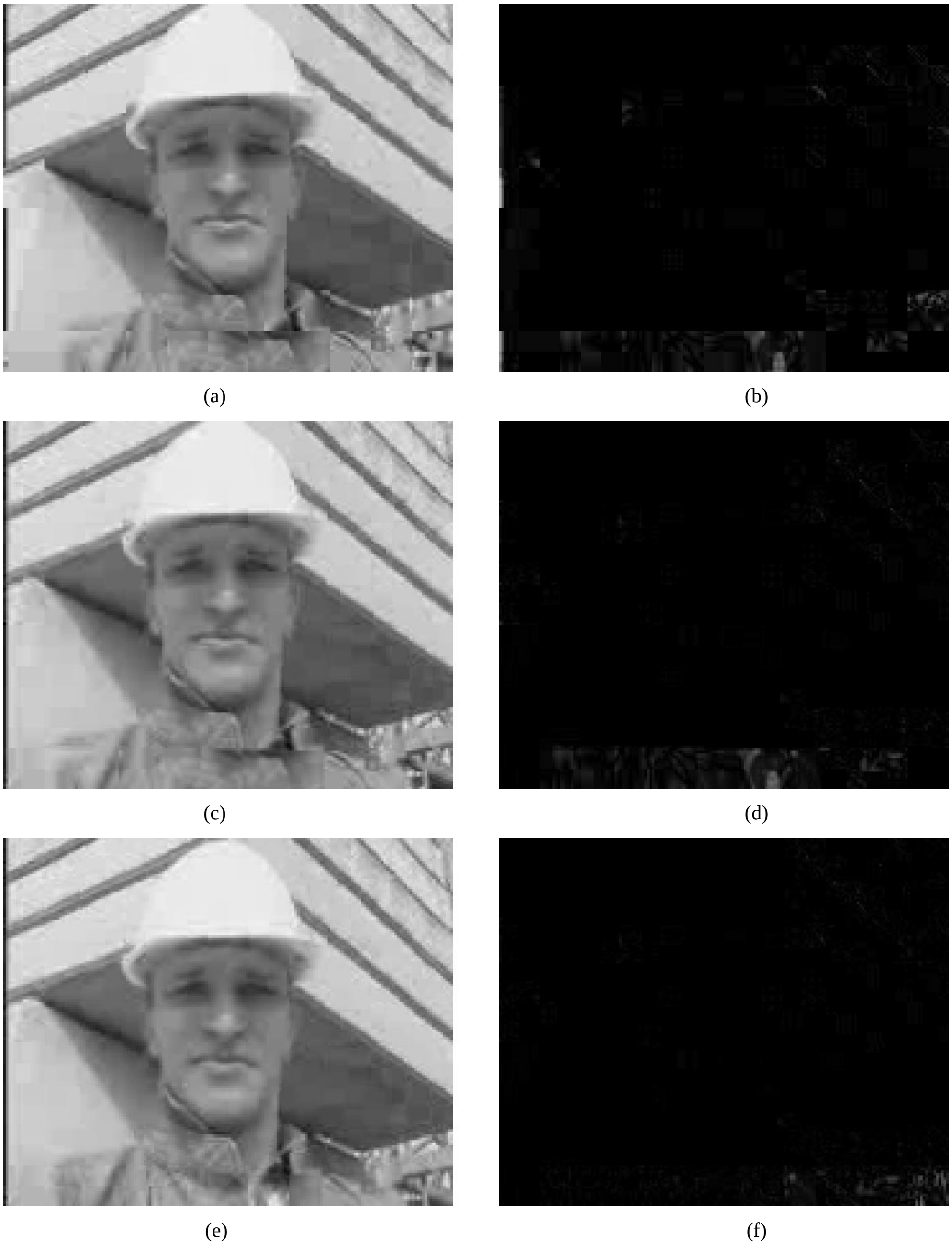

Fig. 7. Illustrations of error-recovery effects. (a) Frame 1 of the Foreman sequence, error condition C9. The error is large because the error packet in the I frame is resent and the bandwidth loss is accumulated into Frame 1. (b) Error frame of frame 1. (c) Frame 2 of the Foreman sequence. (d) Error frame of frame 2, error is less than frame 1. (e) Frame 3 of the Foreman sequence. (f) Error frame of frame 3. More errors are recovered and new errors are generated at the bottom-right MBs.

is 130 Mbytes, which is available using today's SDRAM. Additionally, for a random-error rate of 10^{-3}, only about 8.9% of the MBs need to be processed. The bandwidth can be greatly relieved if the error probability is low.

IV. Experiments and Performance Evaluation

A. Experiment Environment

A software version of the video ARQ proxy is built for performance evaluation. The wired link is assumed to be error free and the video packets can be sent to the proxy without error. In order to get a clear idea of the relation between error and video quality degradation, only one wireless link with error is simulated. Rate control which can handle packet loss on the encoder side is not implemented, and so the loss of bandwidth is absorbed by the video ARQ proxy. The error models used here are random error and burst error. In order to get maximum performance, the BCH code is used in these two models. The optimal BCH code configuration is examined in Appendix II.

Table III shows the error conditions used in the experiment. C1–C4 are random-error conditions with errors ranging from 10^{-4} to 10^{-2}, which are most frequently used in practical conditions. Burst-error patterns with lengths of 5 and 50 ms are used in C5–C8. C9–C14 is a wideband CDMA error pattern from ITU-T [25] that are typical error conditions in wideband CDMA. The packet size of these patterns is 640 bits and each

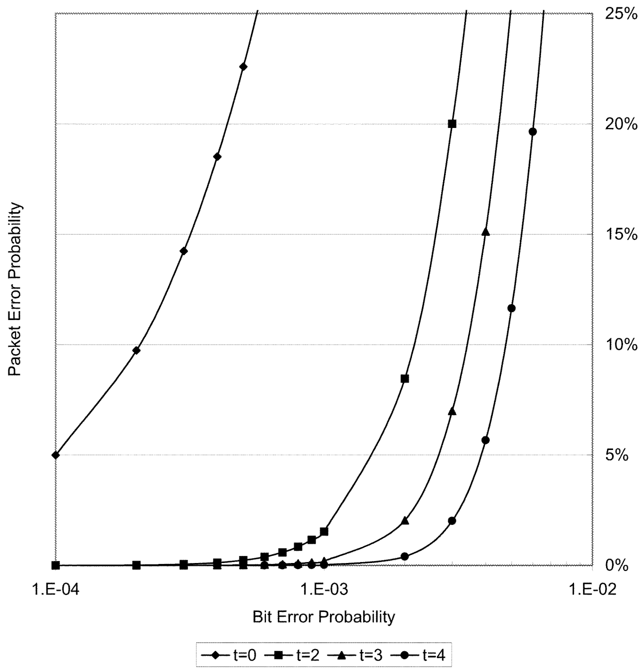

Fig. 8. Packet error under various error conditions and error-correction codes. Packet length $= 511$ bit.

packet has a 16-bit CRC error-detection code attached, and so the packet error can be assumed to be detected. Since channel coding is already implemented in the error patterns, no further FEC coding is used in these error patterns.

Three video sequences are used in the experiment. They are "Foreman," "Carphone," and "Grandma" in QCIF format. Each sequence is coded by TMN 3.0 from UBC at a bit rate specified in Table III. The coding parameter is the H.263 baseline without any option modes. The encoding time for these sequences is 10 s and the frame rate is set to be 10 frames per second (fps).

B. Performance Evaluation

The performances under error conditions in Table III are listed in Tables IV–VI. The first number is the PSNR under error conditions, and the second number is the PSNR under error-free conditions. For a predetermined error condition (C1–C8), proper FEC coding can greatly reduce packet-error rate and the PSNR degradation is no more than 1.2 dB for the Foreman sequence.

The PSNR(Y) of the Foreman sequence in channel C9 is shown in Fig. 6. Compared with that of the only error-concealment approach, the PSNR value is much higher. The PSNR is lower in the first P frame using the video ARQ proxy approach because errors in I frame occupy the bandwidth of the successive P frame. The video ARQ proxy recovers errors by error tracking throughout the sequence. The snapshot of frames 1–3 with its corresponding error frame is shown in Fig. 7. The error tracking and recovery ability can be easily observed in Fig. 7. Note that the bit rate increases little while encoding thr error with current texture. The textured part and error may be cancelled out, since the DCT residual after motion estimation can

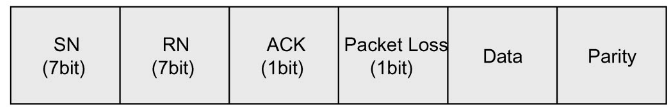

Fig. 9. Typical packet format.

be modeled as zero-mean error. The errors can thus be recovered using the least overhead bits.

From Tables IV–VI, we find that the proposed video transmission architecture handles errors efficiently due to error tracking, and large motion sequences (Foreman) degrade more compared with small motion sequences (Grandma).

Another observation from Tables IV–VI is that PSNR degradation is related to packet-error rate. Moreover, PSNR decreases rapidly when the packet-error rate grows. Under a predetermined error condition (C1–C8), good FEC coding can efficiently reduce packet errors to improve video quality. When the error is larger than the capacity of FEC coding, the packet errors increase, thus reducing video quality. Video quality declines rapidly when packet-error rate increases because the errors will propagate through frames.

V. Conclusions and Discussion

In this paper, a low-delay and error-robust video wireless communication system is presented. A video ARQ proxy server which handles ARQ response and recovers errors is placed at the base station. This network configuration not only reduces delay, but also enhances video quality when error occurs. An FEC coding technique and data-partition technique are also used in the experiment. With extensive computer simulation, the proposed system over H.263 is demonstrated to work well under various error conditions. With a predetermined error condition of 1×10^{-3} error probability, the average PSNR degration is about 0.35 dB for the Foreman sequence. This system is not restricted to the H.263 algorithm. Other DCT-based video standards could be easily adapted to it.

The video ARQ proxy algorithm complexity is also analyzed. Platform-based architecture with a MB-level transcoding engine is recommended in the hardware design because of its flexibility. The flexibility not only reduces design time, but also unnecessary operations. In short, the proposed video transmission architecture is practical and suitable for wireless video communication.

The video ARQ proxy can be made by software or hardware. If the hardware solution is used, the hardware architecture affects performance such as the ARQ response speed and maximum number of concurrent users. The video-processing part of the proxy acts like a transcoder. Many transcoder architectures were proposed [25]–[27]. Most of them are implemented in ASIC and use dedicated control and datapath to speed up the transcoding speed. But in our video ARQ proxy, most of the video data need only parsing operations. Only MBs with errors from a previous frame need to be transcoded. In this situation, dedicated control which needs regular dataflow is not optimal in the hardware design. A platform design with a MB-level transcoding engine may be more suitable for the proxy's hardware implementation.

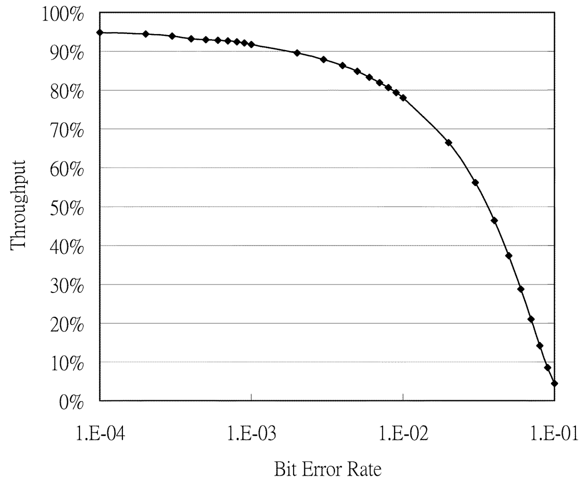

Fig. 10. Maximum throughput using BCH code under various error conditions. Packet length =511 bits.

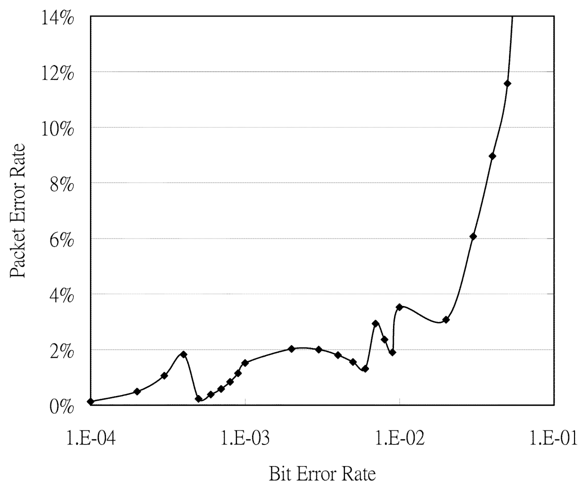

Fig. 11. Packet error using BCH under various error conditions when maximum throughput is achieved. Packet length =511 bits.

APPENDIX I
ERROR MODEL

The error model discussed in this paper is a packet loss rate. In a real wireless system, the relation between bit-error rate and packet-error rate in the random-error model can be calculated. Suppose a random-error probability p, packet length l, and packet error $\mathrm{PER}_{\mathrm{no}}$ without any error-correction procedure is

$$\mathrm{PER}_{\mathrm{no}} = 1 - (1-p)^l.$$

If the error-correction code with maximum error capacity t is used, packet error $\mathrm{PER}_{\mathrm{ec}}$ is

$$\mathrm{PER}_{\mathrm{ec}} = 1 - \sum_{i=0}^{t} \binom{l}{i} p^i (1-p)^{l-i}.$$

A typical packet-error probability trend versus random bit-error probability is given in Fig. 8. The packet length is fixed at 511 bits. This figure shows the effectiveness of FEC coding if the error type and error rate are known.

TABLE VII
BCH CODE CONFIGURATION FOR MAXIMUM THROUGHPUT

BER	Parity bits	Throughput	PER
1×10^{-4}	9	0.949878	0.1260%
5×10^{-4}	18	0.931329	0.2287%
1×10^{-3}	18	0.919297	1.5177%
5×10^{-3}	54	0.849620	1.5520%
1×10^{-2}	81	0.781637	3.5225%
5×10^{-2}	279	0.373775	11.5745%

The packet error in burst-error condition cannot easily be modeled by a single equation because the error bit distribution is not uniform. Computer simulation is used to get the final result. The Gilbert model is used for burst-error generation in this paper.

APPENDIX II
MAXIMUM-THROUGHPUT CHANNEL-CODING DESIGN

In order to get maximum performance in an erroneous channel environment, careful design of the channel coding scheme is important. In this section, the BCH code is investigated under random-error conditions.

A typical ARQ packet is shown in Fig. 9. The header of an ARQ packet needs 16 bits. This could be a big overhead in short packets. Considering the delay which is proportional to packet size, the packet length is chosen to be 511 bits. This packet length is also fitted with packet-length restrictions of BCH code.

The error capacity of the BCH code is nine parity bits per error bit for a 511-bits packet. The maximum throughput is calculated as follows:

$$\mathrm{PER}_{\mathrm{BCH}} = 1 - \sum_{i=0}^{t} \binom{511}{i} p^i (1-p)^{511-i}$$
$$\mathrm{Throughput} = \mathrm{PER}_{\mathrm{BCH}} \times (511 - 16 - 9 \times t).$$

Symbol p is the bit error rate and $\mathrm{PER}_{\mathrm{BCH}}$ is the packet-error rate under error-correction capacity of t symbols. The goal is to obtain t under determined p for maximum throughput. Using a simple computer program to search all possible t, we got the result shown in Figs. 10, 11, and Table VII.

REFERENCES

[1] *Video Codec for Audiovisual Services at $p \times 64$ kbit/s*, Draft ITU-T Recommendation H.261, 1993.
[2] *Video Coding for Low Bitrate Communication*, Draft ITU-T Recommendation H.263, 1997.
[3] *Extensions of Recommendation H.223 for Highly Error-Prone Environment*, Draft ITU-T Recommendation H.223/Annex C, 1998.
[4] *Coding of Moving Pictures and Associated Audio for Digital Storage Media Up to About 1.5 Mbit/s*, ISO/IEC IS 11172(MPEG-1), 1993.
[5] *General Coding of Moving Pictures and Associated Audio Information*, 1994. ISO/IEC IS 13818(MPEG-2).
[6] *Very-Low Bitrate Audio-Visual Coding*, ISO/IEC DIS 14 496(MPEG-4), Oct. 1998.
[7] *Working Draft Number 2, Revision 2 (WD-2),JVT-B118r2*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Mar. 2002.
[8] S. Lin, D. J. Costello, and M. J. Miller, "Automatic repeat error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5–17, 1984.
[9] W. E. Ryan and P. Conoval, "A method of analysis for interleaved Reed-Solomon coding with erasure decoding on burst error channels," *IEEE Trans. Commun.*, vol. 41, pp. 430–434, Mar. 1993.

[10] R. Talluri, I. Moccagatta, T. Nag, and G. Cheung, "Error concealment by data partitioning," *Signal Processing: Image Commun. Mag.*, vol. 14, pp. 505–518, May 1999.

[11] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 952–965, June 2000.

[12] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, pp. 158–162, Apr. 1995.

[13] J. Wen and J. D. Villasenor, "Reversible variable length codes for efficient and robust image and video coding," in *Proc. Data Compression Conf.*, 1998, pp. 471–480.

[14] D. W. Redmill and N. G. Kingsbury, "The EREC: An error-resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 565–574, Apr. 1996.

[15] A. A. El-Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 851–857, 1982.

[16] A. Andreadis, G. Benelli, A. Garzelli, and S. Susini, "FEC coding for H.263 compatible video transmission," in *Proc. ICIP*, Oct. 1997, pp. 579–581.

[17] W. R. Heinzelman, M. Budagavi, and R. Talluri, "Unequal error protection of MPEG-4 compressed video," in *Proc. ICIP*, Oct. 1999, pp. 530–534.

[18] E. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 872–881, Dec. 1997.

[19] P.-C. Chang and T.-H. Lee, "Precise and fast error tracking for error-resilient transmission of H.263 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 600–607, June 2000.

[20] N. Matoba, Y. Kondo, H. Ohtsuka, and T. Tanaka, "Performance evaluation of 32 kbit/s real-time and dual-direction video communication system for wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 664–672, May 2001.

[21] J. Vass, S. Zhuang, and X. Zhuang, "Scalable, error-resilient, and high performance video communications in mobile wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 833–847, July 2001.

[22] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.

[23] S. Aramvith, I. M. Pao, and M. T. Sun, "A rate-control scheme for video transport over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 569–580, May 2001.

[24] W. W, Peterson, *Error Correcting Codes*. Cambridge, MA: M.I.T. Press, Mar 1961.

[25] "Document no. Q11-F-05, WCDMA Error Patterns at 64 kb/s," Ericsson, Lund, Sweden, 1998. ITU-T.

[26] T. Shanableh and M. Ghanbari, "Transcoding architectures for DCT-domain heterogeneous video transcoding," in *Proc. ICIP*, vol. 1, 2001, pp. 433–436.

[27] J. Youn, J. Xin, and M.-T. Sun, "Fast video transcoding architectures for networked multimedia applications," in *Proc. ISCAS*, vol. 4, 2000, pp. 25–28.

[28] L. Bortot, A. Chimienti, M. Lucenteforte, D. Pau, and M. L. Sacchi, "An innovative low complexity MPEG-2 video transcoder operating in the DCT domain," in *Proc. ICCE*, 2000, pp. 312–313.

[29] A. Madisetti and A. N. Willson Jr., "A 100 MHz 2-D 8×8 DCT/IDCT processor for HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 158–165, Apr. 1995.

[30] Y.-P. Lee, T.-H. Chen, L.-G. Chen, M.-J. Chen, and C.-W. Ku, "A cost-effective architecture for 8×8 two-dimensional DCT/IDCT using direct method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 459–467, June 1997.

Tu-Chih Wang was born in Taipei, Taiwan, R.O.C., in 1975. He received the B.S. and M.S. degrees in electrical engineering in 1997 and 1999, respectively, from the National Taiwan University, Taiwan, R.O.C., where he is currently working toward the Ph.D. degree.

His main research interests include video coding technology, DSP architecture, and media processor architecture.

Hung-Chi Fang was born in I-Lan, Taiwan, R.O.C., in 1979. In 2001, he received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., where he is currently working toward the Ph.D. degree.

His research interests are video coding technology and DSP architecture.

Liang-Gee Chen (F'01) was born in Yun-Lin, Taiwan, R.O.C., in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University in 1979, 1981, and 1986, respectively.

He was an Instructor (1981–1986) and an Associate Professor (1986–1988) in the the Department of Electrical Engineering, National Cheng Kung University. During his military service in 1987 and 1988, he was an Associate Professor in the Institute of Resource Management, Defense Management College. In 1988, he joined the Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C. During 1993 to 1994, he was a Visiting Consultant with the DSP Research Department, AT&T Bell Labs, Murray Hill, NJ. In 1997, he was a Visiting Scholar of the Department of Electrical Engineering, University of Washington, Seattle. Currently, he is a Professor with National Taiwan University. His research interests are DSP architecture design, video processor design, and video coding systems.

Dr. Chen served as Associate Editor of both the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1996 to present) and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS (1999 to present). He was the Associate Editor of the *Journal of Circuits, Systems, and Signal Processing* and has served as the Guest Editor of *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*. He is the Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING. He was the General Chairman of the 7th VLSI Design Computer-Aided Design (CAD) Symposium and the1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He was elected as the IEEE Circuits and Systems Distinguished Lecturer during 2001–2002. He has received numerous awards, including the Best Paper Award of the 1992 Asia-Pacific Conference on Circuits and Systems in the VLSI design track (1992), the Annual Paper Award of Chinese Engineer Society (1993), the Best Paper Award from the ROC Computer Society (1990 and 1994), the Outstanding Research Award from the National Science Council, the Dragon Excellence Award from Acer (1996), and Long-Term (Acer) Paper Awards annually from 1991 to 1999. He is a member of Phi Tan Phi.