LOW DELAY, ERROR ROBUST WIRELESS VIDEO TRANSMISSION ARCHITECTURE FOR VIDEO COMMUNICATION

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

DSP/IC Design Lab, Graduate Institute of Electronic Engineering,
Department of Electrical Engineering, National Taiwan University

ABSTRACT

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 FEC coding and ARQ[1] 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.

We use H.263[2] as the experiment platform for our architecture. Experiment shows that average luminance PSNR decreases only 0.35db for "Foreman" sequence under a random error condition of 10^{-3} error probability.

1. INTRODUCTION

The goal of the future wireless communication system is video communication. But one of the major difficulties of wireless video communication is the error sensitive nature of the compressed bitstream. Because compression algorithm often uses VLC codes, errors affect not only the symbol located at the error point, but also the succeeding symbols. Motion compensation procedure also propagates errors and makes video quality unacceptable.

There are many previous arts trying to solve this problem. On the transmission layer, channel coding techniques like Forward Error Correction (FEC), Automatic Repeat reQuest (ARQ) 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[3], Synchronization Marker[4], Reversible Variable Length Codes (RVLC)[5], Error Resilience Entropy Coding[6], etc. Error concealment techniques can be found in [7].

In [8], a combination of ARQ, FEC, error tracking is shown to be the best choice of error resilience tools. We use these error resilience tools plus data partition to enhance the error performance. Error tracking function operates at video proxy server at base station, and other error resilience tools are implemented at the wireless terminal. This configuration facilitates ARQ operation and reduces hardware requirement of the wireless terminal. Error concealment is not considered since new concealment method could be easily implemented on this architecture.

2. 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(a). Two terminals are linked together by a network with errors.

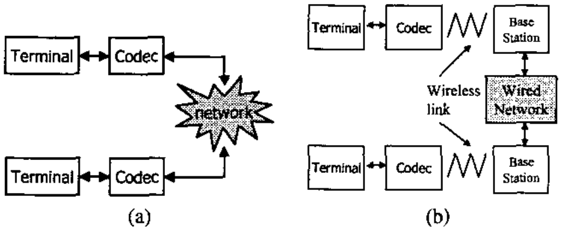

Figure 1: (a) point to point network model (b) wireless network model

In a practical wireless systems, base stations connected to wireless terminals are built and maintained by service providers. 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 Fig. 1(b).Since almost all the errors come from the wireless link, the wired link in Fig. 1(b) 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.

3. DELAY ANALYSIS

Delay constraint is necessary for real-time video communication system. The total delay from encoder to decoder is given in Eq.1. T_{enc} is the encoding time. T_{net} is the network delay from transmitter to receiver. 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 Eq.1 is the ARQ delay. 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. So it is important to analyze and reduce the ARQ delay.

$$T_{total} = T_{enc} + T_{net} + T_{buf} + T_{dec} + T_{ARQ} \quad (1)$$

$$\begin{aligned} T_{ARQ} &= p \times T_{round-trip} + p^2 T_{round-trip} + \dots \\ &= \frac{p}{1-p} T_{round-trip} \quad (2) \end{aligned}$$

The ARQ delay is modeled by Eq. 2[9], where p is the packet error probability. And $T_{round-trip}$ is the delay penalty for resending the packet. Packet error probability can be controlled by the FEC technique. Another way to reduce ARQ delay is to decrease $T_{round-trip}$. $T_{round-trip}$ is shown in Eq.3. It can be divided into three parts. T_{net} is the net propagation delay. T_{tr} is the transmitter processing delay. T_{rev} is the receiver processing delay. If the distance between transmitter and receiver is large, T_{net} will dominate $T_{round-trip}$.

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

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 and the packet delay from wireless terminal to 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 transmitter and receiver.

$$T_{ARQ} = T_{ARQw1} + T_{ARQg} + T_{ARQw2} \tag{4}$$

The ARQ delay model of Fig. 1(b) with ARQ proxy at the base station is now changed to Eq.4. The total ARQ delay is cut into three parts. $T_{ARQw1} + T_{ARQw2}$ is the ARQ delay of wireless links. T_{ARQg} is the ARQ delay of the wired link at ground. The ARQ delay terms in Eq.4 follow Eq.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 Eq.5 and Eq.6. The approximation of Eq.5 holds when the error probabilities are small. In practical case, p_g is small enough to make good approximation of Eq.5.

$$\begin{aligned} p &= p_{w1} + p_g + p_{w2} - p_{w1}p_g - p_g p_{w2} \\ &\quad - p_{w2}p_{w1} + p_{w1}p_g p_{w2} \\ &\approx p_{w1} + p_g + p_{w2} \qquad (5) \\ T_{net} &= T_{w1net} + T_{gnet} + T_{w2net} \qquad (6) \end{aligned}$$

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

$$\begin{aligned} T_{ARQ} &= \frac{2p_{w1}}{1-p_{w1}} T_{w1net} + \frac{2p_g}{1-p_g} T_{gnet} \\ &\quad + \frac{2p_{w2}}{1-p_{w2}} T_{w2net} \end{aligned} \tag{7}$$

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[9]. The wired network delay is assumed to be 200 ms. From the assumption above, we set $T_{w1net} = 10ms$, $T_{gnet} = 200ms$, $T_{w2net} = 10ms$. The average ARQ delay of the proxy model is 0.82 ms, which is calculated by Eq.7. For the same situation as above, the average ARQ delay of point to point model is 18.33 ms as obtained from Eq.2,where the packet error rate p is calculated by Eq.5 and $T_{round-trip}$ is assumed to be $2T_{w1net} + 2T_{gnet} + 2T_{w2net}$. 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.

4. VIDEO ARQ PROXY SERVER

The simple ARQ proxy server discussed in Section 3 only reduces ARQ delay. It can not achieve optimal performance. Resending data increases not only ARQ delay, but also buffer delay because the rate control unit on the transmitter side has slow response to packet loss. 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 total delay. Then time slack for ARQ data resending will reduce 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. By 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 PSNR contribution than DCT residual data. If a DCT residual data packet is discarded to support redending a motion vector data packet, PSNR will be much higher than 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.

4.1. Video ARQ Proxy Behavior

The behavior of the proposed video ARQ proxy is illustrated in Fig. 2. If there is no error on the wireless link, the video ARQ proxy acts like a router.The forward bitstream buffer stores all incoming packets and drops them after the packet acknowledgements from wireless terminal are received. The video ARQ proxy always parses the incoming video packets to trace their symbol types. If an error occurs on the wireless link, the wireless terminal will detect this error and send NACK signal to 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 (macroblock type, motion vectors, intra-block DC coefficient) located in front of other DCT coefficients will be resent. The bandwidth loss caused by packet resending is covered by dropping an unimportant packet (DCT coefficients) at the end of this frame.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. The unimportant packet chosen to be dropped from the bitstream buffer is then decoded into the error frame buffer, which is

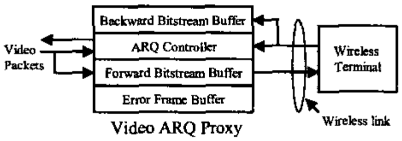

Figure 2: Illustration of video ARQ proxy behavior

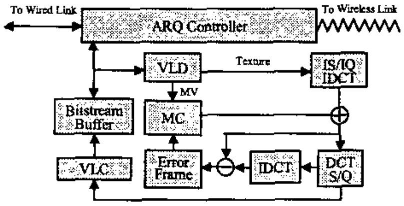

Figure 3: Block Diagram of Video ARQ proxy server

the same size as the frame buffer.

No matter whether the bitrate 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 the bitstream under the real-time constraint, the bitstream not transmitted will be decoded and added to the error frame buffer.

4.2. Video ARQ Proxy Functions

The block diagram of the video ARQ proxy server is shown in Fig. 3. Video packets first come into the ARQ controller. If the packet contains motion vectors or header, 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 motion compensation (MC) unit to compensate error frames. The output of MC is then added to the texture part which is decoded by the 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.

4.3. 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 bitrate = 64 kbit/s, the computation complexity is calculated as shown in Table 1. DCT and IDCT cores can easily handle pixel rate up to 100 Mpixel/sec in today's technology. 6.4Mbit/sec VLC and VLD are also easy to be implemented. S, IS, Q and IQ are simple operations and form no the bottle neck in the whole system. So a reasonable cost implementation could be achieved in today's technology.

Table 1: 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

4.4. 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, QCIF picture format is most widely used in practical situation. One user must have two frame buffers to store essential information because of motion compensation procedure. Thus, one user occupies $176(W) \times 144(H) \times 1.5(YUV420) \times 2 = 76032$ 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 80k bytes memory. The memory requirement of a 100 users proxy is thus 8M bytes.

4.5. Memory Bandwidth Analysis

Memory bandwidth is always a big problem in a digital video system because data rate is high. It is also related to the hardware implementation style. From Table 1, VLC and VLD needs 16Mbytes /sec bandwidth, while motion compensation needs 38Mbytes /sec. If the IDCT/DCT loop is implemented by hardware or the macroblock is cached in software implementation, the loop needs only one read and one write action to the memory. In this case, 76Mbytes /sec bandwidth is needed. Total bandwidth is 130Mbytes /sec, which is available using today's SDRAM.

5. EXPERIMENTAL RESULTS

A software version of the video ARQ proxy is built for performance evaluation. The wired link is assumed to be error free and only one wireless link with error is simulated. The error model used here are random error and burst error with BCH code. Table 2 shows the error conditions used in the experiment. C1 to C4 are random error conditions and C5 to C8 are burst error conditions. The packet size throughout C1 to C8 is 511 bits. C9 to C14 are wideband CDMA 64kb error patterns from ITU-T. The packet size of these patterns is 640 bits and each packet has a 16-bit CRC error detection code. Three video sequences are used in the experiment. They are "Foreman", "Carphone" and "Grandma" in QCIF format. Each is coded by TMN 3.0 from UBC without any option modes. The frame rate is set to be 10fps.

The performancse under error conditions in Table 2 are listed in Table 3. The first number is the PSNR under the error condition, and the second one is under error free condition. For a predetermined error condition (C1-C8), proper FEC coding can greatly re-

Table 2: Channel conditions used in the experiments

Channel	Error Condition	BCH	BR	PER
C1	1×10^{-4}	9	61k	0.13%
C2	1×10^{-3}	18	60k	1.52%
C3	5×10^{-3}	54	55k	1.55%
C4	1×10^{-2}	81	52k	3.52%
C5	1×10^{-3}, 5ms	18	60k	0.51%
C6	1×10^{-3}, 50ms	18	60k	0.31%
C7	1×10^{-2}, 5ms	81	52k	4.81%
C8	1×10^{-2}, 50ms	81	52k	1.88%
C9	1.35×10^{-3}, 3km/h	0	64k	6.73%
C10	1.26×10^{-3}, 40km/h	0	64k	4.93%
C11	9.73×10^{-4}, 120km/h	0	64k	4.61%
C12	8.17×10^{-5}, 3km/h	0	64k	0.72%
C13	1.21×10^{-4}, 40km/h	0	64k	0.81%
C14	9.37×10^{-5}, 40km/h	0	64k	0.69%

Table 3: PSNR(db) of Luminance Component

Channel	Foreman	Grandma	Carphone
C1	31.45/31.49	38.53/38.55	33.68/33.76
C2	31.08/31.43	38.17/38.49	32.75/33.69
C3	30.77/31.10	37.82/38.17	32.53/33.35
C4	30.07/30.85	37.28/37.85	31.65/33.10
C5	31.34/31.49	38.38/38.49	33.47/33.69
C6	31.29/31.49	38.29/38.49	33.50/33.69
C7	29.64/30.85	37.07/37.85	31.47/33.10
C8	30.32/30.85	37.39/37.85	32.47/33.10
C9	29.03/31.68	36.93/38.73	30.55/33.99
C10	30.62/31.68	37.86/38.73	31.97/33.99
C11	29.48/31.68	36.73/38.73	30.89/33.99
C12	31.30/31.68	38.25/38.73	33.05/33.99
C13	31.58/31.68	38.57/38.73	33.42/33.99
C14	31.38/31.68	38.51/38.73	33.65/33.99

duce packet error rate and the PSNR degradation is no more than 1.2 db for the Foreman sequence.

PSNR(Y) of Foreman sequence in the channel C9 is shown in Fig. 4. Compared with that of only error concealment approach, the PSNR is much higher. The DCT residual and the error may be canceled out because the DCT residual can be viewed as zero mean errors. The errors can thus be recovered using the least overhead bits.

6. CONCLUSIONS

In this paper, a low-delay and error robust video wireless communication system is presented. An 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. With extensive computer simulation, the proposed system over H.263 is demonstrated to work well under various error conditions. With predetermined error condition of 1×10^{-3} error probability. The average PSNR degration is about 0.35 db for the Foreman sequence.

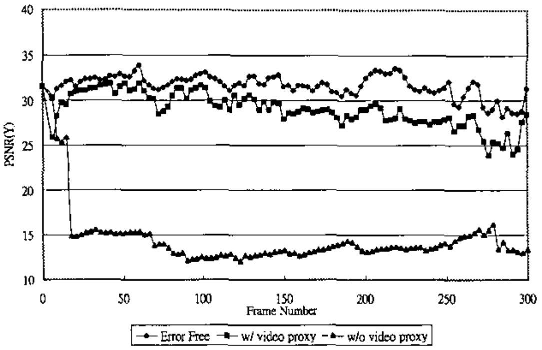

Figure 4: Performance under channel C9

7. REFERENCES

[1] S.Lin, D.J.Costello, and M.J.Miller, "Automatic repeat error control schemes," *IEEE Communication Magazine*, vol. 22, pp. 5–17, 1984.

[2] *ITU-T Recommendation H.263: Video coding for low bit rate communication, version 2*, Jan. 1998.

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

[4] Guy Côté, Shahram Shirani, and Faouzi Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal On Selected Areas in Communications*, vol. 18, no. 6, pp. 952–965, June 2000.

[5] J. Wen and John D. Villasenor, "Reversible variable length codes for efficient and robust image and video coding," in *Proceedings of Data Compression Conference*, 1998, pp. 471–480.

[6] David W. Redmill and Nick G. Kingsbury, "The EREC: An error-resilient technique for coding variable-length blocks of data," *IEEE Transaction on Image Processing*, vol. 5, no. 4, pp. 565–574, Apr. 1996.

[7] Y. Wang and Q. F. Zhu, "Error control and concealment for video communications:a review," *Proceeding of IEEE*, vol. 86, pp. 974–997, May 1998.

[8] Eckehard Steinbach, Niko Färber, and Bernd Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, no. 6, pp. 872–881, Dec. 1997.

[9] Naoto Matoba, Yasushi Kondo, Hiroyuki Ohtsuka, and Toshiaki Tanaka, "Performance evaluation of 32kbit/s real-time and dual-direction video communication system for wireless channels," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 664–672, May 2001.