Development of a Network Adaptive H.264/AVC Medical Video Transmission System

Yi-Fei Xu and Hai-Bo Hu

***Abstract*—Network high definition video applications such as video calls and videoconferences have been increasingly developed recent years. When congestions occur and network environment becomes unstable, the network applications have to deal with such issues like delay, packet loss and bandwidth constraints. In this paper, we describe a high definition H.264/AVC video transmission system with network adaptive function based on real-time transport protocol (RTP) and real-time transport control protocol (RTCP). The adaptive system is composed of three components: 1) transmission management (TM) which focuses on network state monitoring, bandwidth allocating and transmission rate control; 2) a feedback-based adaptive forward error correction (AFEC) scheme using Reed Solomon (RS) code; 3) adaptive video encoding (AVE) that adjusts the bitrate of video streaming according to the bandwidth constraint information offered by TM. In our paper, delay, packet loss and bandwidth constraints are generated by the network bottleneck and congestions. Hence, the main idea is to enhance the bitrate of video streaming adapt to the bandwidth constraints. In addition, we integrate AFEC with RTCP to feedback bandwidth status and make the adaptive video streaming more robust under packet loss environment. Here we call it interactive adaptive forward error correction (IAFEC). The test result shows IAFEC can be adaptive to strict bandwidth constraints and high packet loss rates.**

***Index Terms*—H.264, RTP, RTCP, streaming, reed solomon code, AFEC**

I. INTRODUCTION

In recent years, with the booming usage of high resolution endoscopies, video calls, videoconferences, computer assisted surgery systems and surgical robots, there emerge great demands on the real-time transmission of high definition video streaming. The demands are mainly on data compression, error correction and TM. Firstly, data compression attempts to compress the video data under a limited code rate in order to satisfy the bandwidth and delay constraints. One choice is the H.264/AVC compression standard which has become the most commonly used format for recording and compression of high definition video since 2003 [1]. This standard is flexible and efficient so that more and more network applications have adopted it. Secondly, error correction is conducive to making the video stream transmission more robust in unstable channels. In previously developed applications, people have made many researches about AFEC [8]-[10] and their schemes really work well. However, many of them ignored that the bitrate of a video streaming is not always the same and varies considerably. Hence, we design the interactive adaptive forward error correction (IAFEC) scheme to deal with this problem. Lastly, TM controls the transmission rate of video streaming and allocates the bandwidth by monitoring the real-time channel state. We aim to develop a robust and efficient embedded real-time medical high definition video transmission system integrated with the above three techniques.

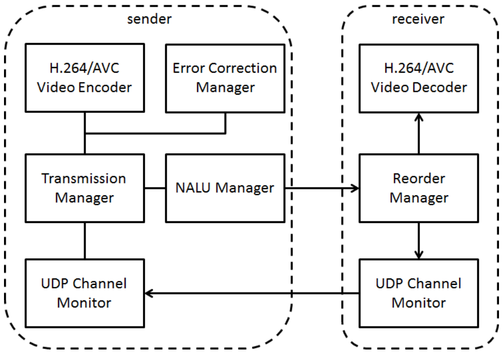

Fig. 1. The architecture of the system.

In Section II, we introduce the architecture of the video transmission system. In Section III, we describe the network adaptive approaches. Then the test result is illustrated in Section IV and the conclusion is given in Section V.

II. TRANSMISSION SYSTEM DESIGN

As shows in Fig. 1, the system can be divided into 2 parts: the sender and the receiver. The sender includes 5 components and they are described as follows:

- H.264/AVC video encoder: Compressing the YUV raw image data streaming into a H.264/AVC network abstraction layer unit (NALU) streaming. Base-line H.264/AVC standard is used here;
- NALU Manager: Making NALUs into fragmentation units, encapsulating the units to RTP source packets and sending them. The fragmenting scheme FU-B is defined in [RFC6184] [6];
- Error Correction Manager: Generating redundant packets using Reed Solomon code (RS-FEC packets) to protect source packets and adjusting the level of redundancy;
- Transmission Manager: Analyzing the monitoring data incoming from channel monitor, controlling the transmission rate of video streaming and allocating the bitrate of the video encoder and the error correction manager;
- UDP Channel Monitor: Monitoring the transporting state particularly the available bandwidth of the

Manuscript received August 6, 2012; revised September 8, 2012
The authors are with the Biomedical Engineering Institute Shanghai Jiao Tong University, Shanghai, China. 200240 (email: huhaibo@sjtu.edu.cn)

DOI: 10.7763/IJFCC.2013.V2.170

RTP-UDP channel by the means of SR (sender report RTCP packet), RR (receiver report RTCP packet) and APP (application-defined RTCP packet) reports over a reliable TCP (transmission control protocol) connection [2].

On the other hand, the architecture of the receiver is relatively simple so that three components are included:

- Reorder Manager: Receiving packets, reordering them according to their sequence number and dropping the packets whose time stamps are dated. If redundant packets are distinguished, the manager will use them to repair lost or delayed RTP packets;
- H.264/AVC video decoder: Decoding the NALUs;
- UDP Channel Monitor: Making the statistics of UDP transmission and error correction performance which are used in RTCP RR and APP reports.

The hardware is based on TMS320DM6467 of Texas Instrument[TM] and the software is based on LIVE555 streaming media. Hence, we should fulfill our obligations under the LGPL that make the modified source code available.

III. Network Adaptive Approach

A. RTP FEC with Reed Solomon Code

Our approach referred to [RFC3550], [RFC5510] and [draft] [2], [5], [7] which is a new draft proposed to the IETF. This RS code based RTP FEC scheme is a RTP packet level scheme. When k source packets, also called the source block, have been transmitted, $n - k$ redundant packets, or called repair packets, will be generated from them. At the receiver side, it can repair all the k source packets if any k ones of the n packets are received. The format of source packets and repair packets is shown in Fig. 2.

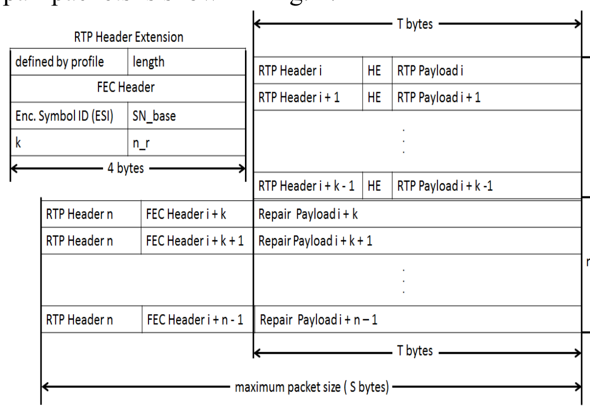

Fig. 2. The format of RTP packets and RS-FEC packets.

RTCP Header (pt, length, etc.)
RTCP Header (SSRC)
name (ASCII)
source packets received
repair packets received
source packets repaired
available bandwidth
← 4 bytes →

Fig. 3. The format of RTCP APP packets.

Unlike the other schemes of [8]-[11] which can only be used in their own system, we raise two designs to make the system more compatible. Firstly, a RTP header extension is used to record the size of a whole source packet so that there is no need to make padding at the tails of source packets unlike that of [9]. The size of the repair packets is related to the Maximum Transmission Unit (MTU) of the working network and always the same so that the header extension is not necessary to them. As a result, we can save a little bandwidth due to the abolishment of padding and header extensions of repair packets. The other point is about the payload type of repair packets. We make source packets and repair packets transporting through the same UDP channel to simplify the bitrate allocating of TM. Their payload types are set to different values, so that the receiver can distinguish repair packets from source packets easily. Other RTSP network applications that cannot identify the payload type value will just drop the repair packets.

B. Channel State Monitor

To make sure the system could be adapted to various kinds of networks, we should monitor transmission channel state and feeds back them to the sender. In [RFC 3550], RTCP protocol is defined to monitor the transmission state. Through analyzing RR reports from the receiver, the system will be able to calculate channel state parameters [2]. They are bandwidth, round-trip delay, one-way delay, jitter and packet loss rate (PLR).

Usually, AFEC researchers need some more information like average burst loss length to help determining the level of redundancy in their schemes [10]. Similarly, we use customized RTCP APP reports to establishing an interaction between the sender and the receiver and feedback the extension information. The format of APP packets is shown in Fig. 3. How to calculate and make use of them will be introduced in section III.C and III.D:

- source packets received (N_{sp}): the number of RTP source packets received between the APPs;
- repair packets received (N_{rp}): the number of RS-FEC repair packets received between the APPs;
- source packets repaired (N_{repair}): the number of source packets repaired timely between the APPs;
- available bandwidth (ABW): the available bandwidth for one video flow which is estimated between the APPs.

C. Network Adaptive TM

Firstly, we should define some additional essential symbols as follows:

- B: the bitrate of RTP-UDP source packets or the output bitrate of the H.264/AVC encoder;
- PLR_{target}: the target packet lost rate we want to achieve;
- $\begin{pmatrix} i \\ j \end{pmatrix}$: the number of ways selecting i things out of j things without repetition.

The main idea of the transmission manager is to analyze the channel state and allocate the bitrate of the H.264/AVC video encoder according to ABW which can be calculated as follows:

$$B = \lfloor ABW \rfloor$$

$$ABW = RBW + SABW$$

$$RBW = \left\lfloor \frac{N_{sp} \times S \times 8/1000000}{T_{APP(i)} - T_{APP(i-1)}} \right\rfloor$$

RBW (mbps) is the real-time useful bitrate of video streaming at time $T_{APP(i)}$ which represents the time when a RTCP APP report responses to the sender. S represents the packet size of RTP-UDP packets. Moreover, SABW (mbps) is the available bandwidth for the whole channel left which is not concerned here. The estimation of SABW is not involved here.

Then we should determine the redundancy level for the video streaming. In real-time network applications, the quality of service (Qos) will be decreased by packet lost and delay. In this paper, the two parameters are end-to-end and only caused by network bottleneck and congestions what means it has nothing to do with equipment troubles etc. Hence, the redundancy is only to make the video streaming robust.

Redundancy information (RI) is defined to measure the performance of RS-FEC scheme like that in [8] and it can be calculated as follows:

$$RI = (n-k)/k$$

We assume that in one RTP-UDP transmission channel the original packet lost rate is PLR. Then the average packet lost rate can be calculated as follows:

$$\overline{PLR} = \sum_{i=n-k+1}^{n} \binom{i}{n} (PLR)^i (1-PLR)^{(n-i)}$$

To make the average packet lost rate under PLR_{target}, n and k should meet the requirements as follows and make RI minimum:

$$PLR_{target} > \sum_{i=k+1}^{n} \binom{i}{n} (PLR)^i (1-PLR)^{(n-i)}$$

Generally, the redundancy level is determined with a feedback-based method like above [8]-[10]. However, this algorithm ignored that the bitrate of a video streaming and its redundancy varies considerably and the PLR cannot be estimated correctly. The real-time bitrate is sometimes two times than ABW and sometimes much less than it due to I pictures. Therefore, we design IAFEC to enhance the feedback based algorithm.

D. Inactive Adaptive Forward Error Correction

The goal of AFEC is to adjust the level of redundancy, increasing or decreasing it according to the network performance [11]. In order to simplify the algorithm and improve efficiency, a parameter based on packet loss rate and RTP FEC repair rate is applied to measure the performance of a video streaming with redundancy.

The program of IAFEC is shown in Fig. 4. At first, an incoming H.264/AVC NALU will be encapsulated to one or several RTP source packets according to [RFC6184] [6]. Every time when a source packet is ready to be transmitted, the system will check the encoding conditions as follows:

- The fragmentation units of one NALU have been entirely transmitted and the number of units is not very small;
- The number of source packets which have been transmitted is larger than a threshold value;
- The time consumed is more than half of the latency constraint.

If any one of the above three conditions is satisfied, the system will begin to generate repair packets (k is equal to the number of source packets stored). Another parameter n can be calculated as follows:

$$n = k + round((\sum_{i=1}^{k} weight_i) \times A)$$

According to our statistics, about five kinds of NALUs are used in network applications as shows in Table I. In IAFEC scheme, every kind of NALU has its own weight and the system decides the level of redundancy on the basis of their weights.

TABLE I: NAL UNIT TYPE OF H.264 BASELINE

NAL_UNIT_ type	Content of NAL unit
1	Coded slice of a non-IDR picture
5	Coded slice of an IDR picture
6	Supplemental enhancement information (SEI)
7	Sequence parameter set (SPS)
8	Picture parameter set (PPS)

The initial value of index A is an experience value related to B. We define RPI (repair performance index) to measure the performance of video streaming and repairing as follows:

$$RPI = \frac{N_{rp}}{N_{sp} \times PLR - N_{repair}}$$

Once a RTCP APP packet is received and identified, the system will calculate the RPI value and compare it with the last RPI value. Then index A will be adjusted according to the result of comparison. After that, RS encoder will work.

E. Adaptive Video Encoding

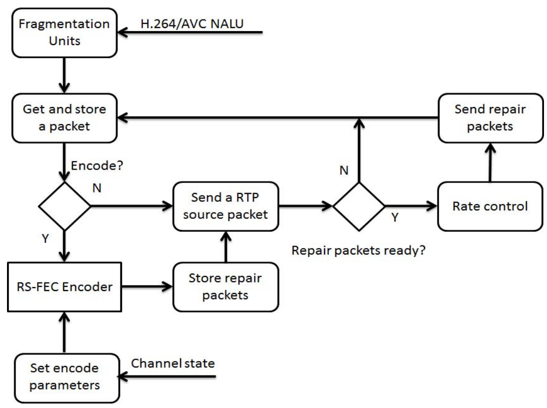

Fig. 4. The flow diagram of IAFEC

The H.264/AVC video encoder applies adaptive video encoding (AVE) technique while compressing data. Once the transmission manager passes in a new set of parameters, AVE will help the encoder producing proper video streaming which meets the bandwidth requirement of the working network.

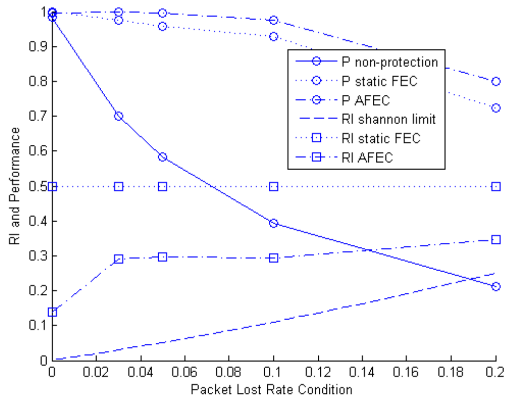

Fig. 5. The performance and RI of the three schemes.

Though many parameters of the encoder may affect the quality of the video streaming, we only decide to adjust the bitrate B as shown in III.C. This method is simple and effective.

IV. Test Result

The conditions, requirements and assumptions are shown in Table II as follows:

TABLE II: Network Conditions

Bandwidth:	about 10mbps
Bitrate of video streaming:	about 7mbps
Latency Constraint: DT	100ms
Maximum RTP packet size: S	1250 bytes
Packet lost rate conditions: PLR	0%, 3%, 5%, 10%, 20%
Target packet lost rate: PLR_{target}	0.0001%

The test is only about the performance of IAFEC under congested network environment. Generally, we should use PSNR to measure the performance but its value cannot be calculated in real-time. Therefore, the rate of useful video source packets received and repaired is used instead.

The packets transport between the sender and the receiver are routed via an emulator called WANem which can emulate a wide area network. By means of this method, we can test the system under different kinds of packet lost rate conditions. While testing, we will also use other tools such like iperf to monitor the real-time network status to ensure the emulator works normally.

The test video is encoded with H.264/AVC baseline. It has a resolution of 1280x720 and duration about one minute. The system runs in three modes: non-protection, static FEC and IAFEC with TM. The test result is shown in Fig. 5. In this figure, P means the performance of the three modes and RI represents the cost. Certainly, there is no cost for the non-protection mode. The RI line of Shannon limit represents the theoretical minimum cost for all FEC schemes.

According to the research of [8], when the original PLR is 10%, normal RTP FEC scheme need a RI between 0.4 and 0.5 to satisfy the quality demands. However, our test shows that a smaller RI about 0.3 is better because the former scheme will lead to a video flow whose bitrate is much more than the channel bandwidth and exacerbate the degree of network congestion.

In addition when the original PLR becomes greater such as 20%, the performance of IAFEC is still better than the other two schemes. It shows that a greater RI is no more useful under congested packet loss network and the RI 0.3 may be the balance between performance and congestion under the particular test environment. Therefore, we can conclude that IAFEC is adaptive to packet loss and bandwidth limitations caused by congestions.

V. Conclusion

In this paper, we designed a network H.264/AVC video transmission system prototype and tested it for applications in adaptive forward error correction (AFEC). A new AFEC scheme based on Reed Solomon code working with RTCP APP packets is developed and validated in laboratory. The test results demonstrate the system's performance under strict bandwidth constraints and high packet loss rates. Future work is to make more research in the transmission characteristic between the streaming characteristic and improve the performance of network video transmission system.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video coding standard," *IEEE Trans. Circuit and Systems for VideoTechnology*, vol. 13, no. 7, pp. 560-576, July 2003.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 3550, Tech. Rep., 2003.

[3] H. Schulzrinne and S. Casner, "RTP profile for audio and video conferences with minimal control," RFC 3551, Tech. Rep., 2003.

[4] A. Li, "RTP payload format for generic forward error correction," RFC 5109, Tech. Rep., 2007.

[5] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo, "Reed-Solomon forward error correction (FEC) schemes," RFC 5510, Tech. Rep., 2009.

[6] Y. K. Wang, R. Even, T. Kristensen, and R. Jesup, "RTP payload format for H.264 video," RFC 6184, Tech. Rep., 2011.

[7] S. Galanos, O. Peck, and V. Roca, *RTP payload format for reed Solomon FEC draft-galanos-fecframe-rtp-reedsolomon-03*, 2011.

[8] G. Tan and T. Herfet, "Optimization of an RTP Level Hybrid Error Correction Scheme for DVB Services Over Wireless Home Networks Under Strict Delay Constraints," *IEEE Trans.* Broadcasting, vol. 53, no. 1, pp. 279–307, March 2007.

[9] M. Mazzotti, E. Paolini, M. Chiani, B. Gadat, C. Bergeron, and R. Fracchia, "Analysis of packet-level forward error correction for video transmission," presented at 2011 Int. Conf. Vehicular Technology.

[10] X. Zhang, X. Peng, S. Fowler, and D. Wu, "Robust H.264/AVC video transmission using data partitioning and unequal loss protection," presented at 2010 Int. Conf. Computer and Information Technology.

[11] K. Park and W. Wang, "AFEC: an adaptive forward error correction protocol for end-to-end transport of real-time traffic," presented at 1998 Int. Conf. Computer Communications and Networks.