Real-Time Scheduling Supporting VCR Functionality For Scalable Video Streaming

Kui Gao[1,2], Yuan Zhang[2,3] , Wen Gao[1,2], Simin He[1]
[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate School, Chinese Academy of Sciences, Beijing 100039, China
[3] Beijing Broadcasting Institute, Beijing 100024, China
{kgao, yzhang, wgao }@jdl.ac.cn, simin.he@ieee.org

Abstract—**Digital video cassette recording (VCR) functionality enables quick and user friendly browsing of multimedia content, thus is highly desirable in streaming video application. In this paper, we propose a layer-based implementation scheme and a layer-based soft real-time scheduling algorithm for efficient implementation of scalable streaming video system to provide full VCR functionality over a dynamic wireless network. Scalable streams provide a layer-based representation for transmitting media contents. Besides the real-time constraint, unequal priorities of scalable streams at different layers and frames are taken into account to support VCR functionality in the proposed algorithms. Thus, the better usage of available bandwidth and smoother playback can be achieved in the streaming system. Simulated results show that, with the proposed algorithms, the playback quality can be efficiently improved.**

I. Introduction

Real-time streaming of multimedia over the broadband wireless networks has evolved as one of the major technical fields in recent years. Due to the wide variation of available bandwidth and transmitted errors over the wireless networks and the variety of end user devices, it is very desirable to have a streaming media coding scheme that can adapt to the channel conditions and user devices. The server should be designed to support a broad range of capacities and dynamically adaptation to time-varying network conditions. To improve the performance of streaming applications, many research works have developed various media coding schemes and data delivery algorithms with scalability. For instance, FGS (Fine Granular Scalable) [1][2] is adopted by the MPEG-4 standard. An FGS encoder compresses a raw video sequence into multiple layer streams, i.e., a base layer bit-stream and one or more enhancement bit-streams. PFGS (Progressive Fine Granularity Scalable) coding scheme proposed in [3] is an improvement over the FGS scheme.

It is highly desirable that scalable video streaming systems support effective and quick browsing. A key technique that enables quick and user friendly browsing of video streaming is to provide full VCR functionality. The set of effective VCR functionality includes forward, backward, step-forward, step-backward, fast-forward, fast-backward, random access, pause, stop and resume, and allows the users to have complete controls over the session presentation. However, the implementation of full VCR functionality with the scalable video streaming is not a trivial task.

Traditional VCR operations impose additional resource requirement on the VOD server in terms of storage space, retrieval throughput and network bandwidth. Chen et al. [4] proposed a segment skipping method. Dey-Sinclair et al. [5] proposed an increased playback rate method. Wee et al. [6] presented a method which is used to convert the I-P frames in the compressed domain. Lin et al. [7] proposed to add a reverse-encoded bitstream in the server. But those techniques may not applicable to scalable video streaming for supporting VCR functionality.

In this paper, we propose a layer-based implementation scheme and a layer-based soft real-time scheduling algorithm to implement full VCR functionality in a scalable streaming system over the wireless networks. The implementation scheme combines skipping method and increased playback rate to support arbitrary playback rate. Different layer and frame have different effort to the quality reconstructed. The real-time scheduling algorithm is an efficient and simple scheme to select the packets from the scalable steaming for supporting the VCR functionality, which can improve the playback quality greatly in client.

The rest of the paper is organized as follows. Section 2 briefly introduces the framework of FGS video coding and the architecture of scalable streaming system. In section 3, we describe our proposed implementation scheme and a real-time scheduling algorithm for supporting full VCR functionality for scalable video streaming. Section 4 shows the simulation of the proposed methods. Section 5 concludes this paper.

II. Framework of FGS Video Coding and the Architecture of Scalable Streaming System

A. The Basic Framework of FGS

Fine granularity scalable (FGS) video coding [1][2] has been accepted as an amendment to the traditional non-scalable MC-DCT approach by MPEG-4 for streaming video profile. The basic idea of FGS video streaming is to code a video sequence into a base layer and multiple enhancement layers. The base layer uses non-salable coding to reach the lower bound of the bit-rate range. The enhancement layer is to code the difference between the original picture and the reconstructed picture using bit-plane coding of the DCT coefficients, with each enhancement layer corresponding to one bitplane. The bitstream of the FGS enhancement layers may be truncated into any number of bits per picture after encoding is completed. The decoder should be able to reconstruct an enhancement video from the base layer and the truncated enhancement-layer bitstreams. The enhancement-layer video quality is proportional to the number of bits decoded by the decoder for each picture.

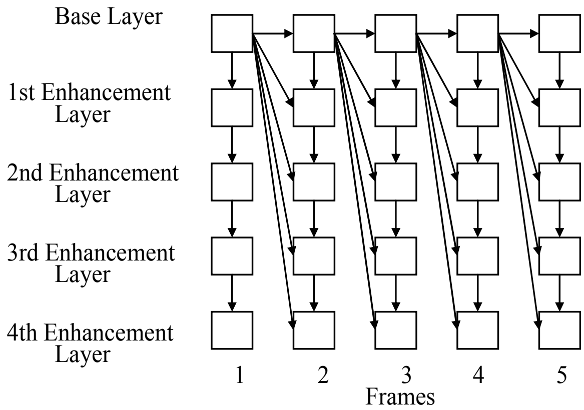

Figure 1. The FGS framework

Figure 1 shows a sample of FGS scalability structure. FGS coding scheme is that the base layer and all enhancement layers in predicted frame are always predicted from the reconstructed version of the base layer in the reference frame.

B. The Architecture of scalable streaming system

A typical streaming system consists of clients and servers on a network. Figure 2 shows the architecture of the scalable streaming media system. The client requests are sent to the server via network connections which also serve for transmission of media data. The buffers in each client are used to provide some tolerance on variations in network delay as well as data consumption rates. To satisfy the performance requirements and VCR interactive control of each client, a scalable streaming server must employ a user manager (i.e. admission control) to determine whether the server can guarantee the QoS (Quality of Service) requirements of a new client without violating the performance requirements of the clients already being serviced. If a new request is admitted, the server will receive the control command and read the data from the storage devices, packetize them, and feed them into the server's transmission buffers. The server selects one packet at a time from those buffers and sends it over the lossy channel. The real-time scheduler in the server controls the packet size and sequence, manages the server transmitted buffer and packets via the network to the clients' buffers.

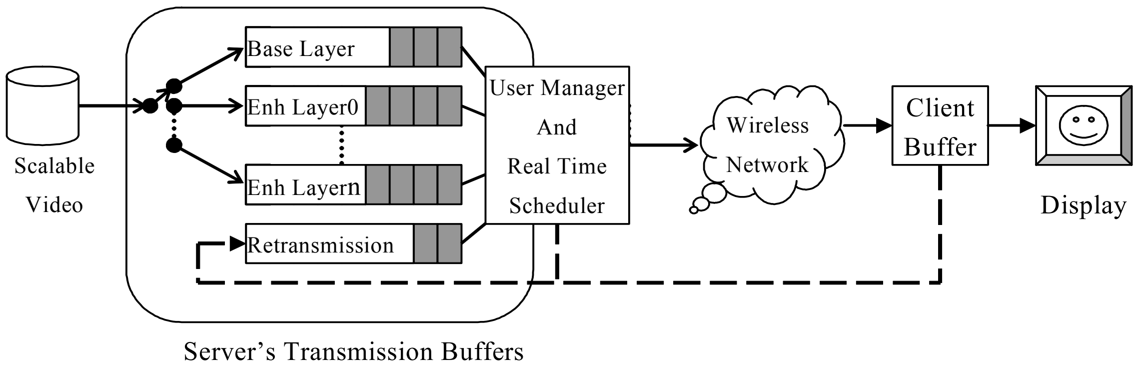

Figure 2. The architecture of the FGS streaming system

III. Real-time Scheduling algorithm support VCR Functionality

Different layer in a frame have different importance to the playback quality reconstructed and the base layer is always predicted from the previous frame at the base layer, whereas each frame at an enhancement layer is predicted from the previous frame at the base layer. Therefore, we proposed layer-based implementation scheme and soft real-time scheduling algorithm for supporting VCR functionality over the wireless networks.

A. Layer-based implementation scheme for supporting VCR functionality

In general, the implementation schemes for support VCR functionality either increased playback rate [4] or skipping frames [5][8] methods. In an increased playback rate method, fast-forward at n-times the normal playback rate requires n-times as many frames to be retrieved n (as compared to the normal playback), yielding an n-fold increase in the load on the server and the network bandwidth. In a skipping frames scheme, fast-forward at n-times the normal playback rate is achieved by transmitting every n^{th} frame to the client site. But such frame skipping schemes may not be directly applicable for scalable video streams that are encoded using motion compensated predictive coding between successive frames in base layer.

Each layer has different dependency, so we proposed an integrated implementation scheme for supporting VCR functionality of scalable video streaming. The base layer in predicted frame is always predicted from the reconstructed version of the base layer in the reference frame, so the increased playback rate is adopted in the base layer. Each frame at an enhancement layer is only predicted from the previous frame at the base layer and not dependent the bits of the enhancement layer, therefore, we use the skipping frame scheme in the enhancement layers. When the request of the fast-forward operation at n-times the normal playback rate is received at the streaming server, the server will sent the n-times as many frames of normal playback in the base layer and sent every n^{th} frame in the enhancement-layer which be truncated according as the network condition. Our proposed scheme is not only simple and efficient but also accommodates to the variation of network bandwidth.

B. Layer-based soft real-time scheduling algorithm for supporting VCR functionality

It is important to select and schedule packets of delivery of scalable video streaming to support VCR functionality. Scalable video streaming consist of a sequence of frames which be decoded and presented continuously in time. If the data does not arrive in time, the playout process will pause, which annoying to human ears and eyes. Due to the variation of the network bandwidth and the delivery delay constraint, not all packets can be transmitted. However, if the server schedules a packet to be sent and the packet arrives the receiver earlier than its playback time, this packet will can be decoded and display in the client. If a frame is not available at its expected display time at the receiver, it misses its deadline. Since the bitplane coding produces an embedded bitstream with fine granularity scalability, the bitstream of enhancement layer can be arbitrarily truncated to fit in the available channel bandwidth. Some packets in a frame cannot be received in the client, but the playback quality is accepted by the user. So the soft real-time packet scheduling algorithm [9] can be used in the packet scheduling policy. Different layer in a frame have different effort to the playback quality reconstructed in the client and the bitstreams of VCR is more complicated than the normal playback, so we proposed the layer-based soft real-time scheduling (LB-SRT) algorithm for supporting the VCR functionality during a streaming session, in such a way as to improve the playback quality in client.

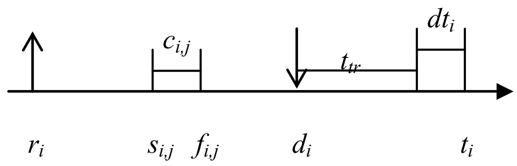

Figure 3. The parameters of a transmit packet

Let $p_{i,j}$ denote the packet of the j^{th} layer in frame i. The packets are put into the transmission buffers according to the decoding order. The release-time $r_{i,j}$ is the earliest time at which the packet $p_{i,j}$ becomes ready for scheduling in transmission buffer. Deadline d_i is the latest time at which all packets of frame i should be sent to the client, otherwise it is too late for playback. We assume that different layers in a frame have the same deadline. The schedule-time $s_{i,j}$ is the time at which the scheduler sends packet $p_{i,j}$ to client. The round-trip-time (*RTT*) is defined as the interval from the time a packet is sent from the server to the time the server gets feedback of this packet from the client. In general, the transmit time of the packet from the server to the client t_{tr} = RTT/2. The size of the packet $p_{i,j}$ is $b_{i,j}$. The current channel bandwidth is B_{cur}. The processing time of the packet $p_{i,j}$ is $c_{i,j} = b_{i,j} / B_{cur}$. The fulfill-time of a packet $p_{i,j}$ is $f_{i,j} = s_{i,j} + c_{i,j}$. The decoding time is dt_i and playback time is t_i. The deadline $d_i \leq t_i - dt_i - t_{tr}$. Some of the parameters defined above are illustrated in Figure 3.

A packet $p_{i,j}$ is ready for scheduling if the following conditions are satisfied: the current time t_{cur} is later than its release-time $r_{i,j}$, and its fulfill-time $f_{i,j}$ is earlier than its deadline, *i.e.*, $r_{i,j} \leq t_{cur}$ and $t_{cur} + c_{i,j} \leq d_i$.

In the session of VCR, the packets of different layer have different importance, so we set the different arrive time $r_{i,j}$ to the different packet in the same frame. More important packet has earlier arrive time. Thus the packets of base layer and important enhancement layers have more chance than other enhancement layers and the scheduler no longer take into the layer importance. If the rate of the fast-forward is n-times the normal playback rate, all the packets of base layer will be sent and the packets of enhancement layers will be sent every n^{th} frame.

The layer-based soft real-time scheduling algorithms for supporting VCR functionality of scalable streaming media over a bandwidth variation network are proposed as follows.

Layer-based soft real-time scheduling algorithm:

Step 1: compare the current time t_{cur} with the deadline d_i of all the packets in the server transmit buffer.

If $t_{cur} > d_i$, move the packet from the server transmit buffer.

Step 2: let T = set of tasks with the lowest layer in the server transmit buffers.

Step 3: **If** $|T| = 1$, (this is only a single packet in T), service that packet; go to Step 1;

Else let t_l be the first packet in T with the lowest layer.

Step 4: Service t_l, $t_{cur} = t_{cur} + b_{i,j} / B_{cur}$. Then move t_l from the buffer.

Go to Step 1.

The layer-based soft real-time scheduling algorithm depends on the decoding order and the importance of the packet. The server transmits the lower (more important) layer packet in the transmit buffer as possible as soon. The scheme not only improves the utility of the bandwidth but also smoothes the playback quality.

IV. SIMULATION RESULTS

The Microsoft H.26L-PFGS encoder/decoder is used in the simulation [10][11]. H.26L-based PFGS is an efficient scalable coding scheme with fine granularity scalability, where the base layer is encoded with H.26L, and the enhancement layer is encoded with PFGS coding. Extensive simulations have been performed to test the performance of the proposed layer-based implementation scheme and soft real-time scheduling algorithm. The sequence Foreman in QCIF format is used in the simulation. It is encoded with 30 frames per second. The maximum level of bitplane is 6 in the sequence Foreman, so there are 6 layers in the Enhancement layer. Different bitplane has different frame size. The enhancement layer bitplane 0 (EL0) is the smallest in size, but it is the most significant layer. The enhancement layer bitplane 5 (EL5) is the largest in size, but it is the least significant layer. The average rate of video data with all enhancement layers is 3,804.094 Kbps. The average rate of base layer is 138.139 Kbps. The sizes of the enhancement layers in the sequence are shown in Figure 4. The packet size is 512 bytes. The frame rate is 30 Hz. The Fast-forward rate is 3 times of the normal playback rate. The playback quality is measured by PSNR of the video frames reconstructed at the client end based on all available packets.

Figure 5 shows comparison results of PSNR for Foreman sequence using different implementation schemes and scheduling algorithms under various bandwidths. The FB-SRT combines the layer-based implementation scheme and frame-based soft real-time scheduling algorithm (not taking into account the importance of the layer). Increased rate is the increased playback rate scheme. Normal playback is the normal playback quality in the same bandwidth. It can be seen that, overall, LB-SRT scheduling schemes outperforms FB-SRT and increased rate schemes and LB-SRT smoothes the playback quality.

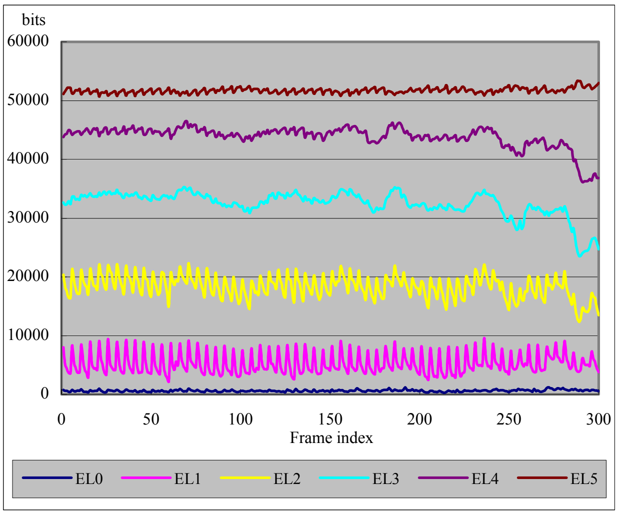

Figure 4. The bits used in each enhancement layer.

V. CONCLUSION

In this paper, we propose a layer-based implementation scheme and a layer-based real-time scheduling algorithm for supporting VCR functionality of scalable video streaming over the wireless networks. The layer-based implementation scheme achieves both efficiency and robustness by considering the layer concept in the scalable video streaming. The LB-SRT scheduling algorithm is efficient and simple to improve the playback quality for supporting VCR functionality of scalable streaming media over a dynamic wireless network and smoothes the playback quality. The simulation result shows that our proposed algorithms outperform the frame-based scheduled algorithm and increased playback rate method with different bandwidth. The low complexity of the proposed algorithms also enables them to be applied in the real-time applications

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China under grant No. 69983008; the National High Technology Development 863 Program of China under grants No. 2001AA112100 and No. 2002AA118010; the Knowledge Innovation Program in Chinese Academy of Sciences grant No. KGCXZ-103.

REFERENCES

[1] "Text of ISO/IEC 14496-2/FPDAM4", ISO/IEC JTC1/SC29/WG11 N3518, Beijing, July 2000.

[2] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Transactions on

Circuit and System for Video Technology, vol. 11, no. 3, pp. 301-317, March 2001.

[3] F. Wu. “A Framework for Efficient Progressive Fine Granularity Scalable Video Coding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 3, pp. 332-344, March 2001.

[4] Ming-Syan Chen, Dilip D. Kandlur and Philip S. Yu, “Storage and Retrieval Methods to Support Fully Interactive Playout in a Disk-Array-Based Video Server,” ACM Multimedia Systems, Vol. 3, no. 3, pp.126-135. July 1995.

[5] J. Dey-Sinclair, J. Saleni, J. Kurose, and D. Towsley, “providing VCR Capabilities in Large-scale Video Servers,” Proceedings ACM Multimedia, pp.25-32, San Francisco, CA, October, 1994.

[6] S.J. Wee and B. Vasudev, “Compressed-domain reverse play of MPEG video streams,” Proceedings SPIE Conference Multimedia Systems and Applications, pp.237-248, November 1998.

[7] C. Lin, J. Zhou, J. Youn and M. Sun, “MPEG Video Streaming with VCR Functionality”, IEEE Transactions and System for Video Technology, Vol 11, no.3, pp415-425, March 2001.

[8] P. Shenoy and H. Vin, “Efficient Support for Scan Operations in Video Servers”, Proceedings ACM Multimedia, San Francisco, CA, pages 131-140, November 1995.

[9] G. Buttazzo, “Hard Real-time Computing Systems”, Kluwer Academic Publishers, 1997.

[10] Y. He, F. Wu, S. Li, Y. Zhong, S. Yang, "H.26L-based fine granularity scalable video coding", IEEE International Symposium on Circuits and Systems (ISCAS 2002), Scottsdale, Arizona, May 2002.

[11] Y. He, R. Yan, F. Wu, S. Li, "H.26L-based fine granularity scalable video coding", ISO/IEC MPEG 58th meeting, M7788, Pattaya, Thailand, December, 2001.

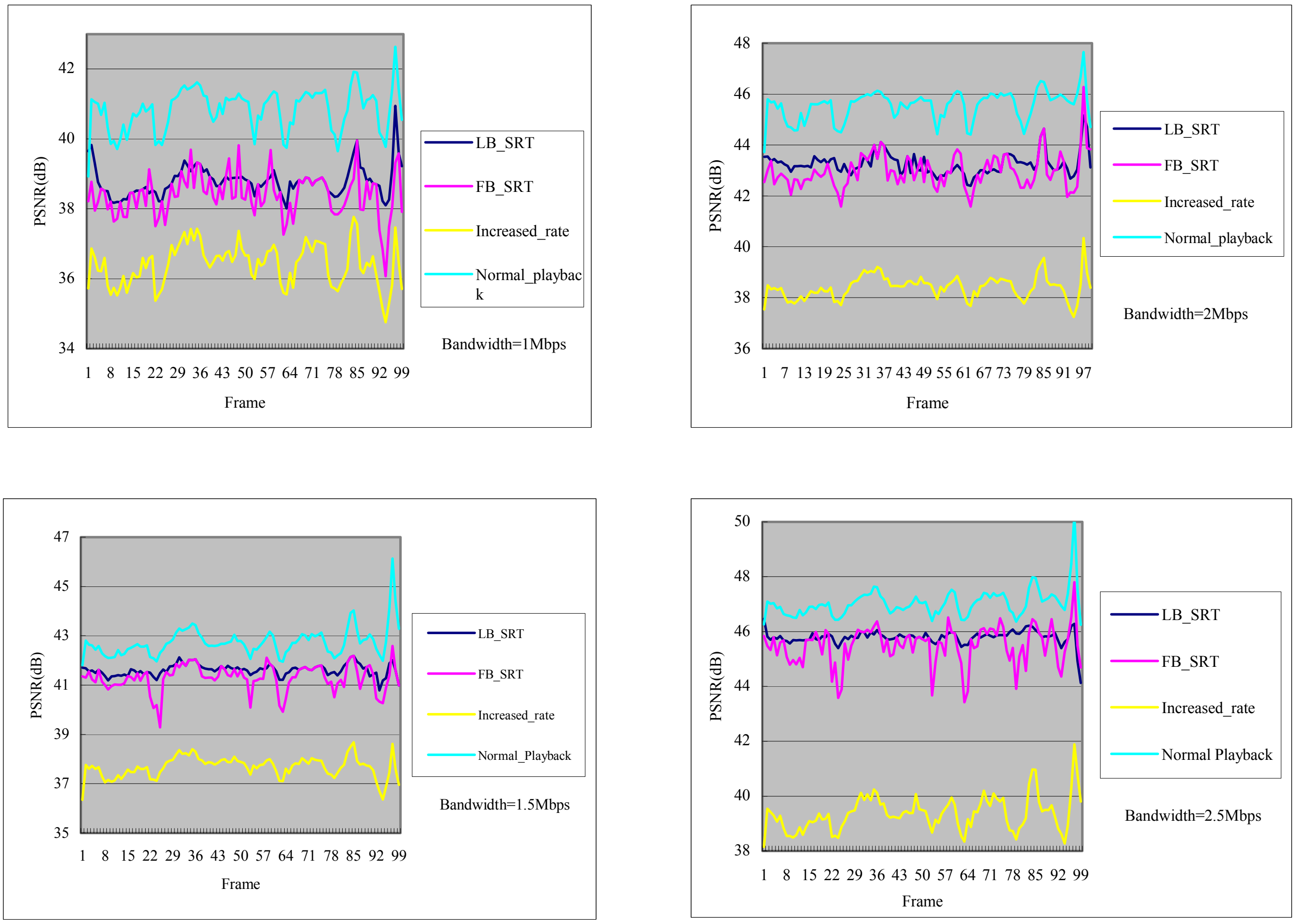

Figure 5. Playback quality under various bandwidths.