REAL-TIME SCHEDULING ON SCALABLE MEDIA STREAM DELIVERY

Kui Gao[1,2], *Wen Gao*[1,2], *Simin He*[1], *Peng Gao*[2], *Yuan Zhang*[2,3]
[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

ABSTRACT

Scalable streams provide a layered representation for transmitting media contents over a channel with packet losses and variable delay. In general, real-time streaming scalable media to audience has timing constraints, and different layers in a frame have different importance to the playback quality reconstructed in client. In this paper, we proposes an efficient and simple real-time scheduling algorithm, Layer-Based Earliest Deadline First (LB-EDF) algorithm, which decides how to transmit/re-transmit video packets at any given time for delivery of scalable streaming media over a lossy channel. Besides real-time constraint, unequal priorities of scalable source bitstream in different layers are taken into account in the proposed algorithm. This guarantees the better usage of available bandwidth and the smoother playback. Simulated results show that, with the proposed approach, the playback quality has a greatly improvement.

1. INTRODUCTION

Real-time streaming of multimedia over the Internet 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 Internet 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. Different from an SNR-scalable encoder, an FGS encoder using the motion-compensated DCT transform coding, which can be compatible to other standards, such as MPEG2, MPEG4, H.263 and H.26L, etc, generates a base-layer video as the lowest quality layer. And then the encoder uses bitplane coding to represent the enhancement stream. With bitplane coding, an FGS encoder is able to achieve continuous rate control for the enhancement stream. This is because the enhancement bit stream can be truncated anywhere to achieve the target bit rate and easily adapt to the channel bandwidth fluctuations. The base layer can be decoded by itself; while the higher layer can be decoded, with the presence of the entire base layer, to obtain enhanced visual quality. PFGS (Progressive Fine Granularity Scalable) coding scheme proposed in [3] is an improvement over the FGS scheme.

Scalable streaming media have timing constraints because of their sensitivity to delay and jitter. Retransmission can be used to recover the lost packets over a best-effort network. However a retransmitted packet typically has an extra delay of one or more RTTs (round-trip-time), and cannot be guaranteed to arrive at the client on time. In addition, even if there is still time to retransmit at a given time, a decision needs to be made on whether this packet should be retransmitted or not. Higher (less important) layers can be dropped to provide more chance for lower (more important) layers to arrive at the client on time.

Optimized scheduling of layered streaming media delivery was raised by Podolsky *et al.* [4], who used Markov chain analysis to find the optimal packets transmission and retransmission policies. Chou and Miao also addressed the same problem with a rate-distortion analysis [5][6].

In this paper, we propose an efficient and simple scheduling algorithm, Layer-Based EDF scheduling algorithm, for delivery of scalable streaming media over lossy channel. It is a priority-based real-time scheduling scheme to select the packets to be transmitted at a given time. The algorithm can improve the playback quality greatly in the client.

The rest of this paper is organized as follows. Section 2 presents the streaming system architecture. Section 3 describes the scalable streaming packet scheduling problem and the proposed algorithm. Section 4 gives some experimental results to verify the performance of the proposed scheduling algorithm. Section 5 concludes this paper.

2. THE SYSTEM ARCHITECTURE

A typical streaming system consists of clients and servers on a network. Figure 1 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. The scheduler in the server controls the packet size and sequence, manages the server transmitted buffer and packets via the network to the clients' buffers.

The media sequence consists of many frames, which are compressed into several layers. The layers are packetized and fed into the server's transmission buffer. These are the packets waiting to be scheduled for transmission. There are also packets into server's buffer that are waiting for retransmission because the client reports them lost. The server's scheduler selects one packet at a time from those buffers and sends it to the lossy channel. Some packets may be lost, damaged or delayed (delayed packets are also considered lost if they exceed their playback delay). At the client end, the lost or damaged packets are reported to the server via a feedback channel.

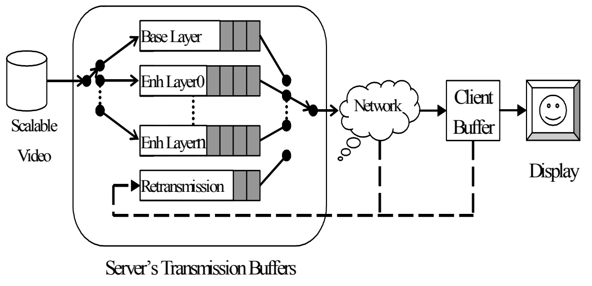

Figure 1. The architecture of PFGS video streaming system.

3. LAYER-BASED EDF SCHEDULING ALGOROTHMS

It is important to select and schedule packet delivery of scalable streaming media over a lossy network. Due to the fixed delay constraint, not all lost packets can be recovered by retransmission. However, if the server schedules a packet to be sent much earlier than its playback time, this packet will have more chances to be retransmitted before it is too late for display. If a packet is not available at its expected display time at the receiver, it will miss its deadline. We want to find a packet transmission policy to select the packets to be transmitted or retransmitted at any give time during a streaming session, in such a way as to improve the playback quality in client.

Let $p_{i,j}$ denote the packet of the jth 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. The packet loss probability over the lossy channel is ε. Information related to RTT and channel error ε can be received by server via client's feedback. The size of the packet $p_{i,j}$ is $b_{i,j}$. The current channel bandwidth is B. The processing time of the packet $p_{i,j}$ is $c_{i,j} = b_{i,j} / B$. 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.

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.*, $r_{i,j} \leqslant t_{cur}$ and $t_{cur} + c_{i,j} \leqslant d_i$.

The real-time scheduling algorithms for delivery of scalable streaming media over a lossy network are proposed as follows.

3.1 Frame-Based EDF Scheduling Algorithm

One basic metric to decide the sending order is the deadline d_i (*i.e.*, the latest time they can be sent before they are no longer useful to the receiver) of the packets. So the server sends the packets with earliest deadline first (EDF) [7] real-time scheduling algorithm. Because different layers in a frame have the same deadline, we called the algorithm Frame-Based EDF (FB-EDF). The precise description of the algorithm is given below.

Frame-Based EDF scheduling algorithm:

```
Step 1: Compare the current time $t_{cur}$ with the deadline $d_i$ of the
        packets in the server transmit buffer.
        If $t_{cur} > d_i$, remove the packet from the server transmit
        buffer.
Step 2: Let $T$ equal to the set of ready packets with the earliest
        deadline. Select the packet of the lowest layer from $T$ and
        send it to the client via the network. And set the timeout of
        this packet as $t_{cur} + RTT$.
Step 3: If the server gets ACK of a packet from the client,
        remove the packet from the buffer; else if a packet reaches
        its timeout, move it from transmission buffer into
        retransmission buffer, and set its release time as the current
        time $t_{cur}$;
        Go to Step 1.
```

In this Frame-Based EDF case, the stay time of packets in the receiver's buffer is minimized, and as a result, we achieve minimum required buffer size at the receiver. If the channel is in good condition without errors and channel bandwidth is enough, it is suitable to use Frame-Based EDF algorithm to send packets in continuous order to obtain minimum average queue length in the client's buffer.

There are many higher layer packets that cannot be decoded when they transmit over a lossy channel with the Frame-Based EDF scheduling algorithm, because the lower layer packet may be lost or damaged. The bandwidth is not fully utilized and the playback quality has a wide variation.

3.2 Layer-Based EDF Scheduling Algorithm

It is instinctive to add another metric with scalable video coding, namely, the relative "importance" of the data in the scalable video stream. For example, the loss of lower enhancement layers within a frame impacts on video quality much greater than that of the higher enhancement layers. Thus it is desirable to send the more important parts of video prior to the less important ones in order to ensure more opportunities for retransmissions in case of packet losses. In fact, the importance of the packet is consistent with the distortion introduced by the loss of the packet. It is quite complex to compute and compare the distortion of every packet. So we adopt a rough estimation of the corresponding distortion at each layer instead. So we proposed the Layer-Based EDF (LB-EDF) scheduling algorithm, which combines importance and priority of the layer.

As packets at different layers have different effects on the playback quality, we set the higher priority to the lower (more important) layer packets and set the lower priority to the higher (less important) layer packets. Thus, important layer packets should be transmitted earlier, with more chances to be retransmitted in case of loss. Packets in the same layer are served according to EDF. The detailed description of the algorithm is given as follows.

Layer-Based EDF scheduling algorithm:

Step 1: compare the current time t_{cur} with the deadline d_i of the packets in the server transmit buffer.
If $t_{cur} > d_i$, remove the packet from the server transmit buffer.

Step 2: let T = set of ready packets with the lowest layer in the server transmission buffers. Select the packet with the earliest deadline from T and send it to the client via the network. And set the timeout of this packet with $t_{cur} + RTT$.

Step 3: **If** the server gets ACK of a packet from the client, remove it from the buffer;
else if a packet reaches its timeout, move it from transmission buffer into retransmission buffer, and set its release time as the current time t_{cur}.
Go to Step 1.

Because of the dependency between layers in a frame, the higher layer packet $p_{i,j}$ cannot be decoded if the lower layer packet $p_{i,j-1}$ does not arrive at the decoder on time. The LB-EDF 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 soon as possible. The scheme not only improves the utility of the bandwidth but also smoothes the playback quality.

4. SIMULATIONS

We use a two-state Markov model to simulate the Internet channel as in Figure 2 [8]. This model can characterize the error sequences generated by data transmission channels. In good state (G) errors occur with low probability while in bad state (B), they occur with high probability. The errors occur in cluster or in burst with relatively long error-free intervals (gaps) between them. The state transitions are shown in Figure 2 and summarized by its transition probability matrix:

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}.$$

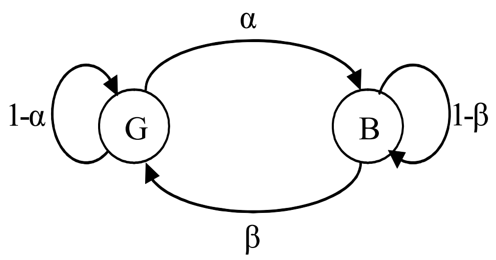

Figure 2. Two-state Markov model for the network simulation.

The average packet loss rate under the two-state Markov model is

$$\varepsilon = \frac{\alpha}{\alpha+\beta}.$$

The Microsoft H.26L-PFGS encoder/decoder is used in the simulation [9][10]. 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. For comparison performance of the scheduling algorithms in the enhancement layer bitstreams, we assume that the base layer can get all the data without any packet losses. Extensive simulations have been performed to test the performance of the proposed real-time scheduling algorithms. The sequence Foreman in QCIF format is used in the simulation. It is encoded with 25 frames per second and 300 frames are encoded and transmitted. 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 sizes of the enhancement layers in the sequence are shown in Figure 3. The enhancement layer 0 (EL0) is the smallest in size, but it is the most significant layer. The enhancement layer 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 playback frame rate is 25 Hz. The RTT is set to 200 ms and the streaming session has an initial delay of 2 seconds. The decoding time is 40ms. The channel packet loss rate varies from 0.05 to 0.5. The channel bandwidth varies from 0.5 Mbps to 3.5 Mbps. The playback quality is measured by PSNR of the video frames reconstructed at the client end based on all available packets.

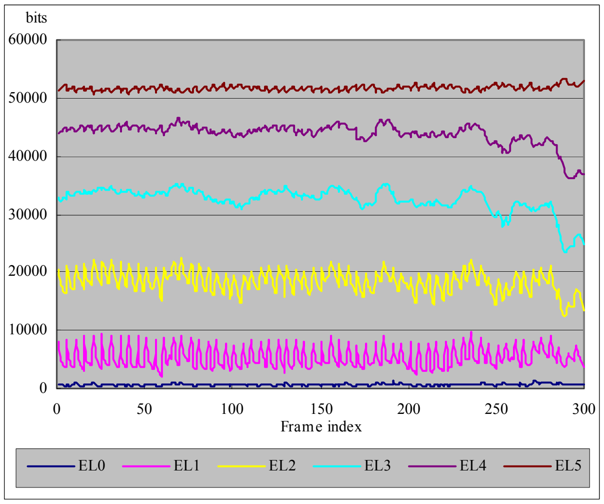

Figure 3. The bits used in each enhancement layer.

Figure 4 and Figure 5 show comparisons between Frame-Based EDF and Layer-Based EDF scheduling algorithms with different parameters. Figure 4 shows playback quality under various packet loss ratios. Figure 5 shows playback quality under various bandwidths. With the Layer-Based EDF scheduling algorithm, the simulation results of the playback quality improves about 2 dB or more over a Frame-Based EDF delivery algorithm.

5. CONCLUSION

In this paper, we propose an efficient and simple real-time scheduling algorithm, Layer-Based EDF scheduling algorithm for delivery of scalable streaming media over a lossy network. It solves the problem of scheduling the packets of scalable streaming media in the server transmission buffers. The simulation result shows that Layer-Based EDF scheduling algorithm outperforms the Frame-base EDF scheduling algorithm in various situations with different bandwidths, channel error ratios, etc. The low complexity of the scheduling algorithm also enables it to be applied in real-time applications.

6. 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 grant No. 2001AA112100; the Knowledge Innovation Program in Chinese Academy of Sciences under grant No. KGCXZ-103.

REFERENCES

[1] W. Li, "Fine granularity scalability in MPEG-4 for streaming video", *IEEE International Symposium on Circuits and Systems* (ISCAS2000), vol. 6, Geneva, May 2000, pp. 29-32.

[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, March 2001, pp. 301-317.

[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, March 2001, pp. 332-344.

[4] M. Podolsky, S. McCanne and M, Vetterli. "Soft ARQ for layered streaming media", *Journal of VLSI Signal Processing Systems, Special issue on multimedia signal processing*, vol. 27, no. 1-2, February 2001, pp. 81-97.

[5] Z. Miao and A. Ortega. "Expected Run-time Distortion Based Scheduling for Delivery of Scalable Media", *The 12th International Packet Video Workshop* (PVW 2002), Pittsburgh, PA, April 2002.

[6] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, February 2001, submitted.

[7] C. Liu and J. layland, "Scheduling Algorithms for Multiprogramming in Hard-Real-Time Environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, January 1973. pp. 46-61.

[8] J.R. Yee and E. J. Weldon. "Evaluation of the performance of the error-correcting codes on a Gilbert channel," *IEEE Transactions on Communications*, vol. 43, no. 8, August 1995, pp. 2316-2323.

[9] 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.

[10] 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.

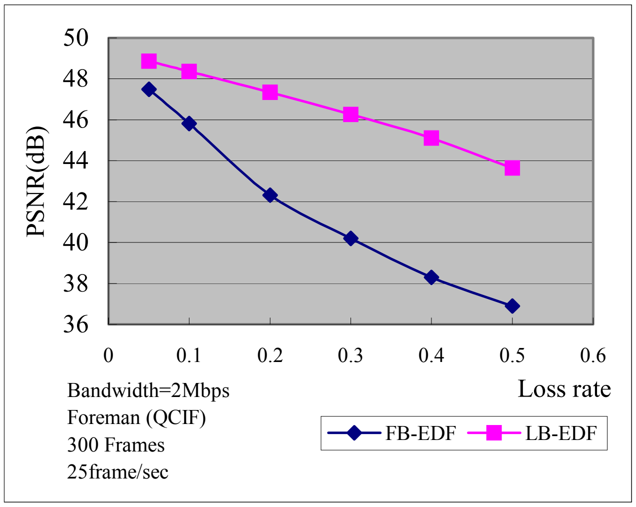

Figure 4 Playback various quality under various channel error.

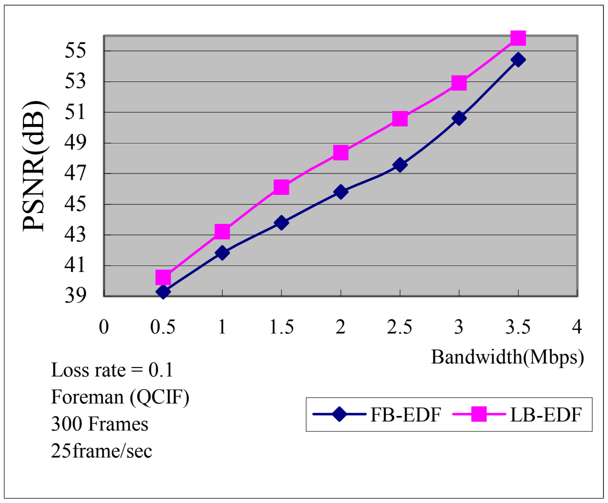

Figure 5 Playback quality under various bandwidths.