ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Real-Time Imaging ▮ (▮▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/rti

Real-time scheduling based on imprecise computation for scalable streaming media system over the Internet

Kui Gao[a,b,*], Wen Gao[a,b], Simin He[a], Yuan Zhang[b,c]

[a]*Digital Media Lab, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*
[b]*Graduate School, Chinese Academy of Sciences, Beijing 100039, China*
[c]*Beijing Broadcasting Institute, Beijing 100024, China*

Abstract

This paper proposes a performance metrics and a real-time scheduling algorithm based on imprecise computation workload model for delivery of scalable streaming media, which can be adapted to network status and QoS requirement over the best-effort Internet. The scheduling task of a scalable streaming media is partitioned into two subtasks: the mandatory subtask for the base layer and the optional subtask for the enhancement layers. The imprecise computation workload model and real-time scheduling algorithm provide scheduling flexibility by trading off video quality reconstructed in client to meet the playback deadline. Thus, the better usage of available bandwidth and smoother playback are achieved.
© 2004 Elsevier Ltd. All rights reserved.

1. Introduction

With the development of multimedia computing and communication technologies, it is feasible to provide streaming media services over the Internet. Heterogeneous access rates, packet loss and fluctuating network bandwidth over IP networks motivate the use of scalable/layered encoding. For instance, the MPEG-4 FGS coding scheme, which has been accepted as a part of MPEG-4 standard [1,2], further provides fine granular scalability for adapting to the network variation. It consists of one non-scalable coded base layer and one or multiple bit-plane-encoded scalable enhancement layers. The base layer provides the basic visual quality, and the other enhancement layers improve the base-layer quality. Fine granularity is implemented by decoding the enhancement stream at any point. It differs from all the previous layered video coding schemes where only limited layers are available. Another advantage of FGS is that the bit rate can be adjusted at transmission time with very fine granularity and very little complexity.

Scalable streaming media have timing constraints because of the sensitivity to delay and jitter. In a real-time streaming system, every video picture/frame must meet its timing constraint typically specified as its deadline, after a streaming starts to playback. If the media data do not arrive at the client in time, the playback will be paused, which is annoying to human ears and eyes. Due to the limitation of available bandwidth and the transmission errors over the Internet, it is difficult to make all the packets to meet their deadline at all times. The delivery of streaming media data should be treated differently from other traditional non-real-time data. The wide range of variation in effective bandwidth and other network performance characteristics over the Internet makes it necessary how to schedule the packets to be transmitted. The problem with the optimized scheduling of layered streaming media delivery was first proposed by Podolsky et al. [3], who adopted the Markov chain to analyze and find the optimal packet transmission and retransmission policies.

*Corresponding author. Digital Media Lab, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China. Tel.: +86 10 82649316.

E-mail addresses: kgao@ict.ac.cn (K. Gao), wgao@jdl.ac.cn (W. Gao), smhe@jdl.ac.cn (S. He), yzhang@jdl.ac.cn (Y. Zhang).

1077-2014/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rti.2004.08.002

Chou et al. [4] and Miao et al. [5] also addressed the same problem with a rate-distortion analysis. Saparilla et al. [8] adopted a fluid model for optimal streaming of layered video.

FGS encodes the video into a base and one or several enhancement layers [1,2]. Fine granularity is achieved from the base layer bit rate upwards as each enhancement layer can be encoded independently with an arbitrary number of bits as allowed. Imprecise computation model [9] provides scheduling flexibility by trading off result quality to meet computation deadlines. In this paper, we adopt an imprecise computation workload model and propose a real-time scheduling algorithm for scalable streaming media. The scheduling task of a stream is partitioned into two subtasks: the mandatory subtask for the base layer and the optional subtask for the enhancement layers. The workload model and scheduling algorithm improve the utility of the bandwidth and smooth the playback quality reconstructed in client by determining how to select and transmit the packets subject to a given time.

This paper is organized as follows. Section 2 briefly introduces the framework of FGS video coding and the architecture of the scalable streaming system. Section 3 presents the imprecise computation workload model, performance metrics and scheduling algorithm for scalable media stream delivery. Section 4 gives some experimental results and comparisons among different algorithms. Section 5 concludes this paper.

2. The architecture for scalable video streaming system

2.1. The framework for scalable video coding

In response to the increasing demand for streaming video applications over the Internet, the coding objective of scalable streaming video is to optimize video quality for a wide range of bit rates, rather than for a fixed bit rate as in the traditional scheme. Fine granularity scalable (FGS) video coding [1,2] has been accepted in just such a technique 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 raw video sequence into a base layer and one or multiple enhancement layers. An FGS encoder using the motion-compensated DCT transform coding which can be compatible to other standards, such as MPEG-2, MPEG-4, H.263 and H.264, etc., generates a base-layer video to reach the lower bound of the bit-rate range. And then the encoder uses bit-plane coding to represent the enhancement layers. The enhancement layer is to code the difference between the original picture and the reconstructed picture using bit-plane coding of the DCT coefficients. The FGS enhancement layers may be truncated into any

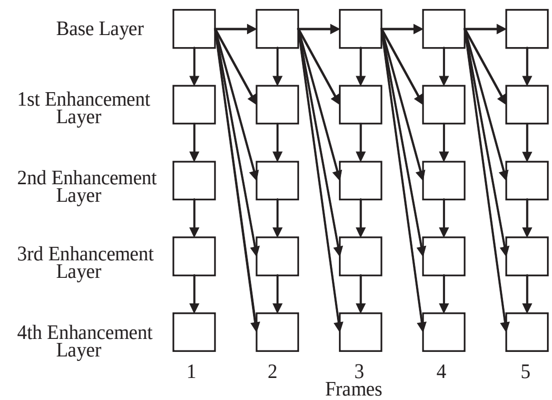

Fig. 1. The FGS framework.

number of bits per picture/frame after encoding is completed. The decoder should be able to reconstruct an enhancement video from the base layer and the truncated enhancement layers. The enhancement-layer video quality is proportional to the number of bits decoded by the decoder for each picture/frame. Fig. 1 conceptually illustrates such an exemplary framework of FGS coding. FGS coding scheme is such that the base layer and all enhancement layers in the predicted frame are always predicted from the reconstructed version of the base layer in the reference frame. The fine scalable characteristic of FGS is very important, since the same content can be accessed over heterogeneous network by various receivers with different computing power, memory, display resolutions, etc.

2.2. The architecture of scalable streaming system

A typical streaming system consists of clients and servers over the Internet. Fig. 2 shows the architecture of a scalable streaming media system. Each client may make real-time requests for scalable streaming. The client requests are sent to the streaming media server via network connections, which also serve for transmission of media data. To satisfy the performance requirements of each client, a scalable streaming server must employ an admission control algorithm 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 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. 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. For a video streaming session, it is desirable to adjust its sending rate

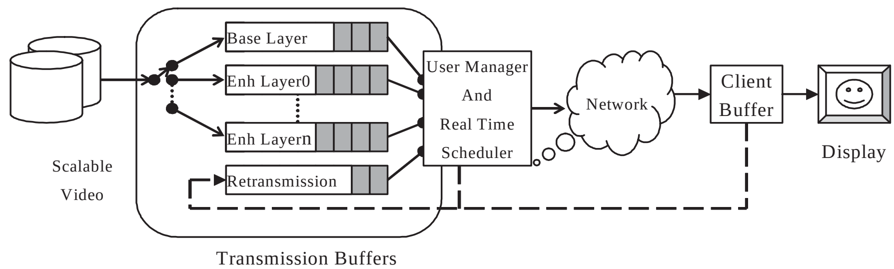

Fig. 2. The architecture of FGS video streaming system.

according to the perceived congestion level in the network and the resource available in the server. Through this adjustment, a suitable loss level can be maintained and resources of network and server can be shared fairly among connections. The receiver monitors the network condition and gathers related information, while the sender changes its sending rate according to the available network bandwidth estimated from the packet loss rate, RTT (round-trip-time), and RTO (retransmission timeout) values. A retransmitted packet typically has an extra delay of one or more RTTs, 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. It may be possible to quickly transmit/retransmit the base layer so that it is insufficient to support all the layers, the server does not transmit higher layers, which results in lower but often acceptable quality for the user. A critical property of layered encoding is that in order to decode a layer, all the lower layers must also be present at the client. We propose a performance metrics and a real-time scheduling algorithm based on the imprecise computation workload model for delivery of scalable streaming media over the Internet. The details are described in the next section.

3. Imprecise computation workload model and scheduling algorithm for scalable streaming system

3.1. Imprecise computation workload model for scalable streaming system

In a real-time streaming system, the server packetizes the coded scalable/layered streams into some packets and then sends them to the client through the Internet. Substreams at different layers have different contributions to the playback quality obtained in client. The base layer provides the basic visual quality, and the other enhancement layers improve the base-layer quality. Therefore, we adopt imprecise computation workload model [9] for scalable media stream. Logically, each stream is decomposed into two parts by the FGS encoder: the base layer and the enhance layers. The scheduling task of a streaming media is partitioned into two subtasks: the mandatory subtask M for the base layer and the optional subtask O for the enhancement layers. The mandatory subtask M is required for an acceptable QoS and must be scheduled before the deadline. The optional subtask O refines the result. It can be left unfinished and terminated at its deadline, if necessary, lessening the playback quality at the client end. Furthermore, the optional subtask O is dependent on the mandatory subtask M; the mandatory subtask M must execute before the optional subtask O.

We are given a set of n frames in a scalable media stream, $F = \{F_1, F_2, \ldots, F_n\}$. Let $p_{i,\,j}$ denote the packet of the jth layer (or bit-plane) in frame F_i. m is the number of layers including base layer and all the enhancement layers. When $j = 0$, that means $p_{i,j}$ is the base layer packet of frame F_i. When $j > 0$, $p_{i,j}$ is the enhancement layer packet. Thus, the set of mandatory subtask $M = \{p_{1,0}, p_{2,0}, \ldots, p_{n,0}\}$, and the set of optional subtask $O = \{p_{1,1}, p_{1,2}, \ldots, p_{i,j}, \ldots, p_{n,m}\}$ $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$. The mandatory subtask of frame F_i is $m_i = \{p_{i,0}\}$, and the optional subtask of frame F_i is $o_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,m}\}$. The release time $r_{i,j}$ is the earliest time at which the packet $p_{i,j}$ becomes ready for scheduling in the transmission buffer. Deadline d_i is the latest time at which all packets of frame F_i should be sent to the client, otherwise it is too late for playback. We assume that different layers in frame F_i have the same deadline d_i. We call the time interval $[r_{i,j}, d_i)$ the feasibility interval of the packet $p_{i,j}$. The sending time $\tau_{i,j}$ $(r_{i,j} \leqslant \tau_{i,j} \leqslant d_i)$ 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 size of the packet $p_{i,j}$ is $b_{i,j}$. The current channel bandwidth is $B(t)$. The sending time of the packet $p_{i,j}$ is $c_{i,j}(t) = b_{ij}/B(t)$. The fulfill-time of a packet $p_{i,j}$ is $f_{i,j}(t) = \tau_{i,j}(t) + c_{i,j}(t)$. The decoding time is $dt_i(t)$.

If the available bandwidth is insufficient, the optional subtask O can be left incomplete. To guarantee that the video playback is continuous and the reconstructed video quality can be accepted, the scheduler needs to guarantee that all mandatory subtasks M are allocated sufficient available bandwidth to transmit the packets by their deadline; it uses the leftover available bandwidth to transmit as many packets of the enhancement layers as possible.

To ensure that imprecise computation works properly, we propose a performance metrics and a real-time scheduling algorithm to make sure that all the mandatory subtasks have bounded resource and processing time requirements and are allocated sufficient available bandwidth to be transmitted by their deadlines. The performance metrics and real-time scheduling algorithm for the imprecise computation workload model are described as follows.

3.2. The performance metrics for the imprecise computation workload model

A traditional way of measuring distortion in multimedia signal is using PSNR (Peak Signal to Noise Ratio). Reconstruction with only base layer results in a higher distortion, i.e., a smaller PSNR, while when more layers are used the distortion is reduced. Chou et al. [4] and Miao et al. [5] proposed a transmission scheme to minimize the end-to-end distortion in a rate-distortion optimized way. An interactive descent algorithm is adopted in order to minimize the average end-to-end distortion. Zhang et al. [6] adopt rate-distortion function in order to provide the user with the best-perceived video quality. Dai et al. [7] analyzed the rate distortion optimal modeling of MPEG-4 FGS. We propose a performance metrics that take into according not only the distortion of the packets but also the real-time restrict (i.e. the playback deadline).

Let $D_{i,j}$ be the distortion of the packet $p_{i,j}$. We define the total distortion of a frame F_i, D_{Fi}, as the distortion when a frame is completely lost; a frame reconstructed with all its layers has a minimum distortion, i.e., zero. Define the total distortion of the media stream as the sum of D_{Fi} of all the individual frames. With the FGS coding scheme, each frame at the base layer and an enhancement layer is predicted from the previous frame at the base layer and not dependent on the bits of the enhancement layer. Precedence constrains specify the dependences between the packets in set F. The mandatory subtask M must be finished before the deadline in order to get an acceptable playback quality in client. The optional subtask O just improves the playback quality. So the scheduling algorithm for the imprecise computation workload model adopts different real-time scheduling schemes for different subtasks. The server and network must provide enough resources (i.e. available bandwidth) to finish the mandatory subtask M before they miss the deadline. As for the optional subtask O of the enhancement layer, the loss impact of lower enhancement layer within a frame on video quality is much greater than that of the higher enhancement layers within it. The constraints are given by a partial-order relation "$\prec$" defined over F[4]. If $p_{i,j} \prec p_{l,k}$, $p_{l,k}$ is a successor of $p_{i,j}$, and $p_{i,j}$ is a predecessor of $p_{l,k}$. That is to say, in order for packet $p_{l,k}$ to be decoded, packet $p_{i,j}$ must also be decoded. Thus, if a set of packets is received at the client, only those packets whose ancestors have all been also received can be decoded. The playback distortions are defined as the total distortion minus the distortion of all layers and frames that are actually decoded at playback. Define $a_{i,j}$ as an indicator such that $a_{i,j} = 1$ if $p_{i,j}$ is used for playback; otherwise $a_{i,j} = 0$. For a media sequence with n frame in the set F, with l layers in each frame, the playback distortion, D_{Fi}, can be obtained as

$$D_{Fi} = \sum_{i=1}^{n}\sum_{j=1}^{l} d_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{l} a_{i,j}d_{i,j}. \quad (1)$$

Now that the on-time arrival of a packet $p_{i,j}$ does not necessary mean that it can be used for playback, as decoding a packet is not possible unless all its parent packets have arrived on time. In our system, we propose that the sender knows the distortion of the packets in the buffer. When streaming stored video, the distortion can be computed off-line from the original uncompressed video segment. It can be stored at the sender, together with the video file. Obviously, an optimal scheduling scheme would be yielded when

$$J = \min\left(\sum_{i=1}^{n} D_{Fi}\right) \quad (2)$$

is minimum.

In this paper, we set the accurate priority of the packet according to the distortion. The base layer bitstream is very sensitive to channel errors. Any random errors or burst errors may cause the decoder to lose synchronization, and the decoded errors will propagate to the start of the next GOP. However, the enhancement layers can tolerate the channel errors. When there are errors in the enhancement layers, a decoder can simply drop the rest of the enhancement bitstream of this frame and search for the next synchronization marker. We adopt FEC scheme for base layer and retransmission scheme for enhancement layer. Therefore, we will set the higher priority to the base layer packets. For the enhancement layer packets, the packet loss will only affect the single frame and will not propagate to the later frames. Therefore, the incurred distortion from the enhancement layer packers can be accurately calculated from only one frame. The packet priority in enhancement is calculated as

$$w_{i,j} = \frac{D_{i,j}}{b_{i,j}}. \quad (3)$$

The optimal algorithms of real-time scheduling imprecise computations to meet deadlines and minimize total distortion use a modified version of the classical earliest deadline first algorithm. This is a preemptive, priority-driven algorithm that assigns priorities to tasks according to their deadlines. The scheduling algorithm in detail is as follows.

3.3. The real-time scheduling algorithm imprecise computations workload model

We now propose a real-time scheduling algorithm based on the imprecise computation workload model to select packets for transmission at any given time. It is assumed that packets have the same release time in a frame. The release time a_i of packet $p_{i,j}$ is the time when it is put into transmission buffer and becomes ready for scheduling. The packet $p_{i,j}$ can be selected and scheduled if the following conditions are satisfied: the current time t_{cur} ($t_{cur} \in [t, t+T)$) is later than its release-time a_i, and its fulfill-time $f_{i,j}$ is earlier than its deadline, i.e., $a_i \leqslant t_{cur}$ and $t_{cur} + c_{i,j} \leqslant d_i$.

For the set of mandatory subtasks $M = \{p_{1,0}, p_{2,0}, \ldots, p_{n,0}\}$ (i.e., the base layer packets) are set at the higher priority, therefore these subtasks will scheduled according to the earliest deadline first. We adopt FEC scheme to recover the loss packet of the base layer. For the set of optional subtask $O = \{p_{1,1}, p_{1,2}, \ldots, p_{i,j}, \ldots, p_{n,m}\}$ (i.e., the enhancement layer packets), we can construct the possible model that a packet in transmission buffer is one of two states: ready or blocked. When the packet is in ready state, it is prepared to schedule when given the opportunity. While the packet is in blocked state, it cannot be scheduled until getting NAKs or its timeout is not larger than current time t_{cur}. The server wishes to select a packet ready to be transmitted at t_{cur} from the transmission buffer. Let S_{buf} be the set of all the ready packets that are the candidates to be transmitted in the buffer, i.e., $S_{buf} = \{p_{i,j} | a_{i,j} \leqslant t_{cur}$ and $t_{cur} + c_{i,j} \leqslant d_i$ and in ready state$\}$. The server selects the optional subtask with priority calculated according to (3). If there are same priorities, the server will select and schedule the optional subtask with earliest deadline. The enhancement layer data can be transmitted/retransmitted earlier. If it is lost, it can have more chances to be retransmitted. The algorithm not only improves the utility of the bandwidth but also smoothes the playback quality. The precise description of real-time scheduling algorithm is given below.

The imprecise computation scheduling algorithm:

Step 1: compare the current time t_{cur} with the deadline d_i of all the mandatory subtasks and the optional subtasks.

If $t_{cur} > d_i$, discard the subtasks;

Step 2: let $T(m)$ = set of ready mandatory subtasks M and $T(o)$ = set of ready optional subtasks O;

Step 3: if $T(m) \neq \emptyset$, schedule and execute the mandatory subtasks with the earliest deadline from $T(m)$.

Step 4: if $T(m) = \emptyset$, schedule and execute the optional subtasks with the highest priority first. The subtasks in the same priority are served according to the earliest deadline and partial-order relation. If a set of packets is sent by the scheduler, only those packets whose ancestors have all been also sent. And set the timeout of the subtask with $t_{cur} + RTT$.

Step 5: if the server gets ACK of an optional subtask from the client, remove it from the buffer;

else if an optional subtask reaches its timeout, the server will move it from transmission buffer into retransmission buffer, and set its release time as the current time t_{cur}.

Go to Step 1.

The imprecise computation scheduling algorithm combines importance of different layer to reconstruct the playback quality and real-time scheduling algorithm. Thus, it is guaranteed that the most important data (i.e. the base layer sub stream) can be transmitted before the deadline. The scheduling algorithm not only improves the utility of the bandwidth but also smoothes the playback quality.

4. Simulation results

A two-state Markov model proposed by Gilbert [10] is used to simulate packet losses in the Internet channel. 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 bursts with relatively long error-free intervals (gaps) between them. The state transitions are shown in Fig. 3 and summarized by the following transition probability matrix:

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

The average packet loss rate is:

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

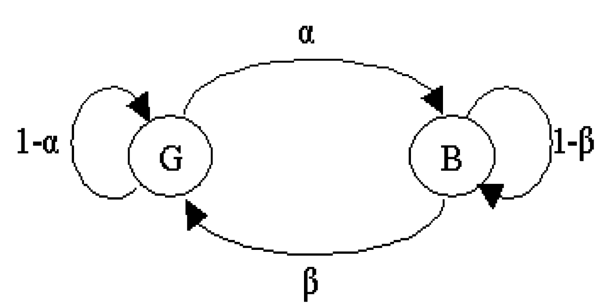

Fig. 3. Two-state Markov model for the network simulation.

Details of the model can be found in [10]. It is assumed that the sending rate can be decided by TCP-friendly Rate Control (TFRC) protocol. The receiver monitors the network condition and gathers related information, while the sender changes its sending rate according to the available network bandwidth estimated from the packet loss rate, round-trip time, and retransmission timeout values. The protocol uses an equation-based way to estimate available bandwidth [11]:

$$R_{net} = \frac{s}{R\sqrt{(2p/3)} + t_{RTO}(3\sqrt{(3p/8)})p(1+32p^2)},$$

where s is the packet size, R is the round trip time, t_{RTO} is the retransmission timeout value, and p is the packet loss ratio. The sending rates in different RTTs and loss rates are simulated by ns-2 [13].

The MPEG-4 FGS-MoMuSys encoder/decoder [12] is used in the simulation. The base layer is encoded with MPEG-4, and the enhancement layer is encoded with FGS coding. Extensive simulations have been performed to test the performance of the proposed algorithms. The sequences Foreman, Coastguard and Akiyo in CIF format are used in the simulation. They are encoded with 30 frames per second and 300 frames are encoded and transmitted. For example, the maximum level of bit-plane is 7 in the sequence Foreman, so there are 7 Enhancement layers. Different bit-plane have different sizes. The enhancement layer 0 (EL0) has the smallest size, yet it is most significant. The enhancement layer 6 (EL6) is the largest in size, yet it is the least significant. In our simulations, the channel packet loss rate varies from 0.5% to 10% and the RTT varies from 20 to 160 ms. The playback quality is measured by PSNR of the video frames reconstructed in client based on all available packets.

The layer-based imprecise computation real-time scheduling algorithm (LB-ICRT) and the frame-based round-robin scheduling algorithm (FB-RR) are adopted to schedule the packets. It can be seen that, overall, LB-ICRT algorithm outperforms FB-RR algorithm. Fig. 4–6 show comparisons of PSNR with different scheduling algorithms. Fig. 7 shows that the bit numbers of enhancement layer can be used to reconstruct video quality in client under different scheduling algorithms. Obviously, the layer-based imprecise computation scheduling algorithm improves the utility of the bandwidth and smoothes the playback quality in various situations with different round-trip times, channel errors, etc.

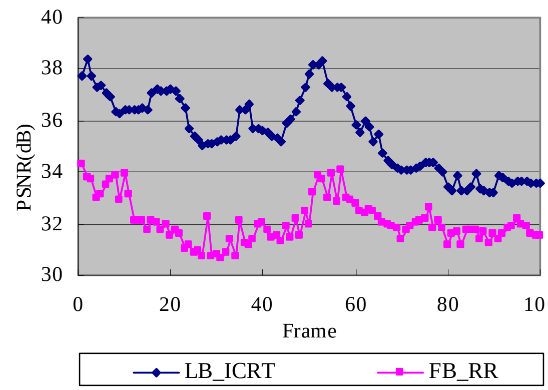

Fig. 5. PSNR comparisons of different scheduling algorithms with loss rate = 5% and RTT = 20 ms.

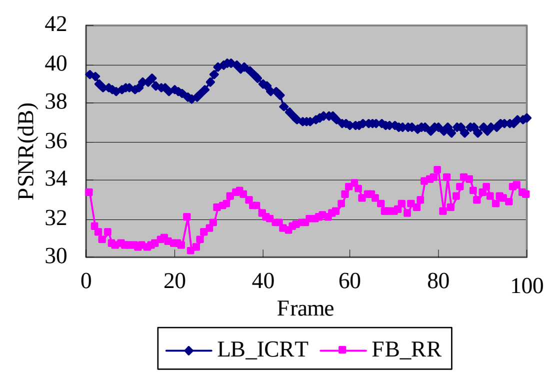

Fig. 6. PSNR comparisons of different scheduling algorithms with loss rate = 2% and RTT = 40 ms.

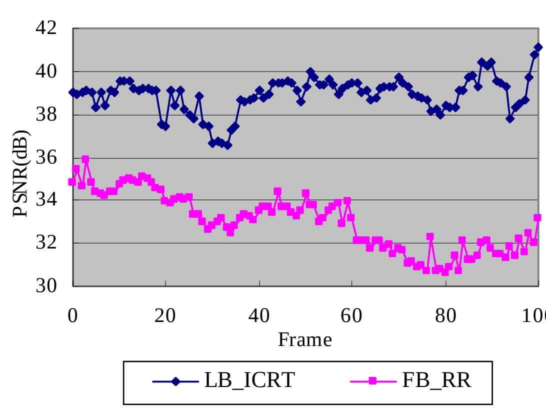

Fig. 4. PSNR comparisons of different scheduling algorithms with loss rate = 2% and RTT = 20 ms.

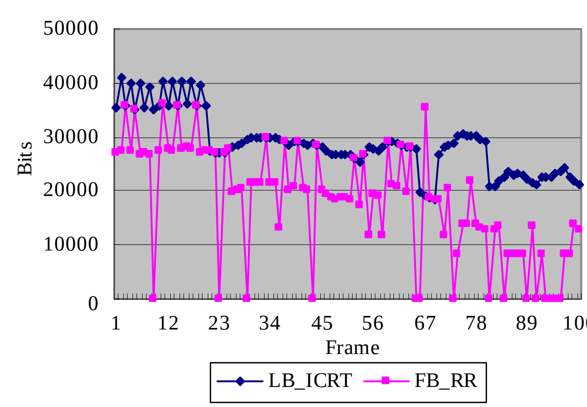

Fig. 7. Enhancement Layer bits number comparisons of different scheduling algorithms with loss rate = 5% and RTT = 20 ms.

5. Conclusion

In this paper, we adopt a real-time imprecise computation workload model and propose an imprecise computation scheduling algorithm on scalable media stream delivery. The goal is to find an optimal transmission policy for the scalable streaming server to achieve the best playback quality at the client end. The scheduling task of each stream is divided into two parts: a mandatory subtask and an optional subtask. The mandatory task is for the base layer substream and the optional task is for the enhancement layer substreams. Different subtask adopts different real-time scheduling scheme. The imprecise computation workload model and scheduling algorithm can efficiently solve the real-time scheduling problem of the packets in the scalable streaming server buffer before transmission. The simulation results show that the imprecise computation workload model and scheduling algorithm outperform the traditional best-effort model and frame-based scheduling algorithm in various situations with different RTTs, channel errors, etc. The imprecise computation scheduling algorithm enables the use of imprecise computation workload model as a means to provide scheduling flexibility in scalable streaming systems and enhance their fault tolerance and improve the playback quality. The low complexity of the proposed algorithm also enables them to be applied in real-time applications.

Acknowledgment

This work has been supported by the National High Technology Development 863 Program of China under grants No. 2001AA112100 and the National Natural Science Foundation of China under grant No. 69983008.

References

[1] Li W. Fine granularity scalability in MPEG-4 for streaming video. IEEE International Symposium on Circuits and Systems (ISCAS2000) 2000;6:29–32.
[2] Li W. Overview of fine granularity scalability in MPEG-4 video standard. IEEE Transactions on Circuit and System for Video Technology 2001;11(3):301–17.
[3] Podolsky M, McCanne S, Vetterli M. Soft ARQ for layered streaming media. Journal of VLSI Signal Processing Systems (Special issue on Multimedia Signal Processing) 2001;27(1–2):81–97.
[4] Chou PA, Miao Z. Rate-distortion optimized streaming of packetized media.IEEE Transactions on Multimedia, February 2001, submitted for publication.
[5] Miao Z, Ortega A. Expected run-time distortion based scheduling for delivery of scalable media. The 12th International Packet Video Workshop (PVW 2002), Pittsburgh, PA, April 2002.
[6] Zhang Q, Zhu W, Zhang Y-Q. Resource allocation for multimedia streaming over the internet. IEEE Transactions on Multimedia 2001;3(3):339–55.
[7] Dai M, Loguinov D, Radha H. Analysis and distortion modeling of MPEG-4 FGS, Proceedings, ICIP 2003, Barcelona, Spain, September 2003.
[8] Saparilla D, Ross KW. Optimal streaming of layered encoded video, Infocom 2000. Israel: Tel Aviv; 2000.
[9] Liu JWS, Lin KJ, Shih WK, Yu A, Chung JY, Zhao W. Algorithms for scheduling imprecise computations. IEEE Computer 1991; 58–68.
[10] Yee JR, Weldon EJ. Evaluation of the performance of the error-correcting codes on a Gilbert channel. IEEE Transactions on Communications 1995;43(8):2316–23.
[11] Floyd S, Handley M, Padhye J, Widmer J. Equation-based congestion control for unicast applications. SIGCOMM 2000, February 2000.
[12] ISO/IEC 14496-5:2001/FDAM1, ISO/IEC JTC1/SC29/WG11/N4711, Jeju Island, March 2002.
[13] ns-2 network simulator. http://www.isi.edu/nsnam/ns, 2002.