# 1996 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS

JUNE 23-27
TEXAS
IM '96, ICA Expo and IEC

## CONVERGING TECHNOLOGIES
## FOR TOMORROW'S APPLICATIONS

ICC '96
JUNE 23 - 27, 1996
DALLAS CONVENTION CENTER
DALLAS, TEXAS USA

## CONFERENCE RECORD
## VOLUME 3 OF 3

SPONSORED BY THE IEEE COMMUNICATIONS SOCIETY AND THE DALLAS SECTION OF IEEE

IEEE ICC GLOBECOM COMSOC

RPX Exhibit 1228

# 1996 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS

ICC '96 JUNE 23-27 DALLAS TEXAS
With SUPERCOMM '96, ICA Expo and IEC

| | |
|---|---|
| IEEE Catalog Number: | 96CH35916 |
| ISBN Numbers: | |
| Softbound Edition: | 0-7803-3250-4 |
| Casebound Edition: | 0-7803-3251-2 |
| Microfiche Edition: | 0-7803-3252-0 |
| CD-ROM | 0-7803-3382-9 |
| Library of Congress Number: | 81-649547 |

## COPYRIGHTS AND REPRINT PERMISSIONS

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the lim
U.S. copyright laws for private use of patrons those articles in this volume that carry a code at the bott
first page, provided the per-copy fee indicated in the code is paid through the:

Copyright Clearance Center
222 Rosewood Drive
Danvers, MA 01923

For copying, reprint, or republication permission, write:

Manager of Copyrights
IEEE
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331

Additional copies of this publication are available from:

IEEE Operations Center
P. O. Box 1331
445 Hoes Lane
Piscataway, NJ 08855-1331 USA

1-800-678-IEEE
1-908-981-1393
1-908-981-9667 (Fax)
833-233 (Telex)
email: customer.service@ieee.org

# Dynamic Video Playout Smoothing Method for Multimedia Applications

Maria C. Yuang, Shih T. Liang, Yu G. Chen, and Chi L. Shen

Department of Computer Science and Information Engineering
National Chiao Tung University, Taiwan

**Abstract:** Multimedia applications including video data require the smoothing of video playout to prevent potential playout discontinuity. In this paper, we propose a dynamic video playout smoothing method, called the *Video Smoother*, which dynamically adopts various playout rates in an attempt to compensate for high delay variance of networks. Specifically, if the number of frames in the buffer exceeds a given threshold (*TH*), the Smoother employs a maximum playout rate. Otherwise, the Smoother uses proportionally reduced rates in an effort to eliminate playout pauses resulting from the emptiness of the playout buffer. To determine *TH*s under various loads, we present an analytic model assuming the Poisson Process arrival correspondent with a network with the traffic shaper. Based on the analytic results, we establish a paradigm of determining *TH*s and playout rates for achieving different playout qualities under various loads of networks. Finally, to demonstrate the viability of the Video Smoother, we have implemented a prototyping system including a multimedia teleconferencing application and the Video Smoother performing as a part of the transport layer. The prototyping results show that the Video Smoother achieves smooth playout incurring only unnoticeable delays.

## 1. Introduction

Recent evolution in high–speed communication technology enables the development of distributed multimedia applications combining a variety of media data, such as text, audio, graphics, images, and full–motion video. For supporting distributed multimedia applications, researchers have encountered various design problems. In particular, the smoothing of video playout has been considered crucial to prevent potential playout discontinuity. Several playout smoothing methods with various degrees of performance have been proposed. These methods fall into two main categories: *buffer–oriented* and *bufferless*. Buffer–oriented methods preserve playout continuity by buffering packets at the receiver [17,18] or delaying the playout time of the first packet received [2,3,4,9,15,17,23,26]. These methods have been shown to be feasible but at the expense of a decrease in playout throughput. On the other hand, bufferless methods [12,22] smooth playout through adjusting the source generation rate by means of feedback techniques. These methods are effective, however, unviable for live–source applications.

In this paper, we propose a dynamic video playout smoothing method, called the *Video Smoother*. Generally, unlike existing methods described above, the Video Smoother dynamically adopts various playout rates according to the number of frames in the playout buffer in an attempt to compensate for high delay variance of networks. Specifically, if the number of frames in the buffer exceeds a given threshold (*TH*), the Smoother employs a maximum playout rate. Otherwise, the Smoother uses proportionally reduced rates in an effort to eliminate playout pauses resulting from the emptiness of the playout buffer. To determine *TH*s under various loads, we present an analytic model network with the traffic shaper. Based on the analytic results, we establish a paradigm of determining *TH*s and playout rates under various loads of networks. Finally, to demonstrate the viability of the Video Smoother, we have implemented a prototyping system including a multimedia teleconferencing application and the Video Smoother performing as a part of the transport layer. The prototyping results show that the Video Smoother achieves smooth video playout incurring only unnoticeable delays.

The remainder of this paper is organized as follows. Section 2 presents our playout smoothing method, including the analytic model and results. The prototyping system and experimental results are then demonstrated in Section 3. Finally, conclusion remarks are given in Section 4.

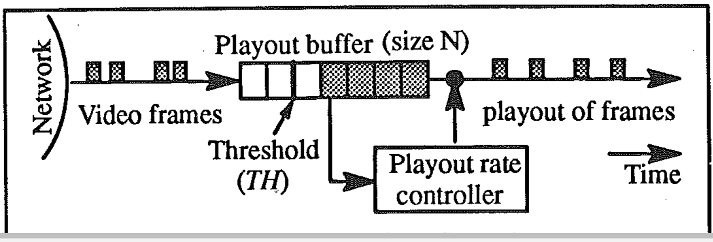
## 2. Video Smoother

Generally, the Video Smoother dynamically adopts various playout rates according to the number of frames in the playout buffer. In this section, we first present a queueing model and analysis. The analytic data in turn establish the paradigm on which the playout rates under various network arrivals and loads are based.

### 2.1 Model and Analysis

The model for the Video Smoother, as shown in Figure 1, is composed of an arrival stream of video frames, a finite playout buffer with size $N$, an output stream of video frames to be playbacked, and a playout rate controller responsible for adjusting the playout rate. The playout rate is dependent on the current number of video frames in the buffer and the *TH*. When the number of frames in the buffer exceeds the *TH*, the Video Smoother employs a maximum playout rate denoted as $\mu$; otherwise, the smoother uses proportionally reduced rates to eliminate playout pauses resulting from the emptiness of the playout buffer.

It is worth noting that the determination of the *TH* can profoundly affect the system performance. If the *TH* is overestimated, the playout rate declines resulting in serious degradation of the playout performance. On the other hand, if the *TH* is underestimated, the probability of having an empty buffer increases resulting in playout discontinuity. To determine an optimal *TH*, we propose a queueing analysis in which the service rate is state–dependent. In general, we first derive the steady–state queue occupancy distribution as a function of the *TH*. This, in turn, allows us to compute the probability of having an empty



buffer and the mean playout rate. The optimal $TH$ can then be selected by trading off the rise in the probability of having an empty buffer against the increase in the playout rate. In what follows, the analytic model is given in detail.

Let $p_{i,j}$ denote the transition probability of the queue occupancy altered from $i$ to $j$ frames, as seen by departure frames. Thus, the queue occupancies at frame departing epoches form an embedded Markov chain, as shown in Figure 2, with the state transition probability matrix ($P$) given as

$$P = [\, p_{i,j} \,] = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,N} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ & \cdots & & & \\ p_{N,0} & p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{bmatrix}. \tag{1}$$

To derive the steady–state buffer size distribution, one has to first determine $p_{i,j}$, which is dependent on both the current state ($i$) and the frame arrival process. The arrival is modelled by the Poisson process. Let $\lambda$ be the mean frame arrival rate, and $B_i(t)$ be the service–time Cummulative Distribution Function (CDF) when $i$ customers are in the buffer. Then,

$$p_{i,j} = \begin{cases} \displaystyle\int_0^\infty \frac{e^{-\lambda t}(\lambda t)^{j}}{j!} dB_i(t), & i = 0,\ i–1 \le j < N; \\ \displaystyle\int_0^\infty \sum_{k=N}^{\infty} \frac{e^{-\lambda t}(\lambda t)^{k}}{k!} dB_i(t), & i = 0,\ i–1 \le j = N; \\ \displaystyle\int_0^\infty \frac{e^{-\lambda t}(\lambda t)^{j-i+1}}{(j-i+1)!} dB_i(t), & 0 < i \le N,\ i–1 \le j < N; \\ \displaystyle\int_0^\infty \sum_{k=N}^{\infty} \frac{e^{-\lambda t}(\lambda t)^{k-i+1}}{(k-i+1)!} dB_i(t), & 0 < i \le N,\ i–1 \le j = N; \\ 0, & \text{elsewhere.} \end{cases} \tag{2}$$

Let $i$ be the current number of frames in the buffer. Consider a smoothing strategy in which the mean service rate is set to a maximum value ($\mu$) when $i$ exceeds the $TH$, and is set to $\dfrac{\max\{i,1\}}{TH}\mu$ when $i$ decreases below the $TH$. Suppose the service time is exponentially distributed, i.e.,

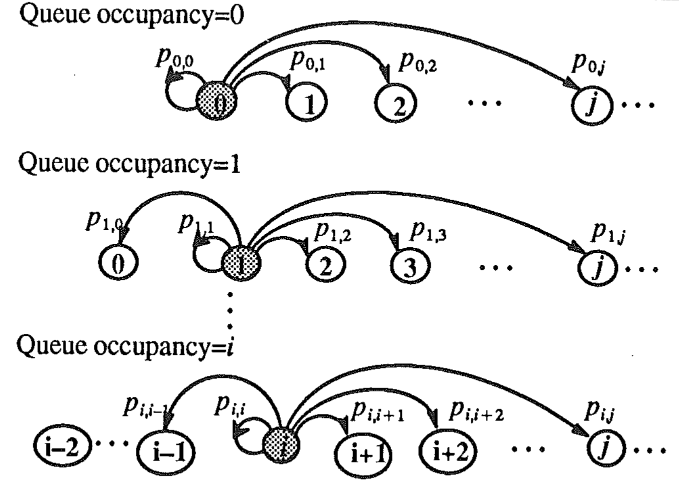


Figure 2. State–transition diagram for queue occupancies

$$B_i(t) = \begin{cases} 1 - e^{-\frac{\max i\{,1\}}{TH}\mu t}, & i < TH; \\ 1 - e^{-\mu t}, & i \ge TH. \end{cases} \tag{3}$$

Thus, $p_{i,j}$ can be obtained by applying Equation (3) to Equation (2) as

$$p_{i,j} = \begin{cases} \dfrac{\lambda^{j} \cdot \frac{\mu}{TH}}{(\lambda + \frac{\mu}{TH})^{j+1}}, & i = 0,\ i–1 \le j < N; \\ \left(\dfrac{\lambda}{\lambda + \frac{\mu}{TH}}\right)^{j}, & i = 0,\ 0 \le j = N; \\ \dfrac{\lambda^{j-i+1} \cdot \frac{\min\{TH,i\}}{TH}\mu}{(\lambda + \frac{\min\{TH,i\}}{TH}\mu)^{j-i+2}}, & 0 < i \le N,\ 0 \le j < N; \\ \left(\dfrac{\lambda}{\lambda + \frac{\min\{TH,i\}}{TH}\mu}\right)^{j-i+1}, & 0 < i \le N,\ i–1 \le j = N; \\ 0, & \text{elsewhere.} \end{cases} \tag{4}$$

With $p_{i,j}'s$ given in Equation (4), the stochastic equilibrium distribution for the queue occupancies, $\Pi \equiv [\pi_0, \pi_1, \cdots \pi_N]$, can be directly computed by solving the stationary equation, $\Pi = \Pi P$. In addition, the frame loss probability ($p_L$), given the buffer size of $N$, can be obtained as

$$p_L = \pi_0 \cdot \left(\frac{\lambda}{\lambda + \frac{\mu}{TH}}\right)^{N+1} + \sum_{i=1}^{N} \pi_i \cdot \left(\frac{\lambda}{\lambda + \frac{\min\{TH,i\} \cdot \mu}{TH}}\right)^{N-i+2} \tag{5}$$

Furthermore, the mean playout rate ($\overline{B}$) can be formulated as

$$\overline{B} = \sum_{i=0}^{N} \pi_i \cdot \frac{\min\{TH, \max\{i,1\}\}}{TH}\mu. \tag{6}$$

## 2.2 Analytic Results

We have so far obtained the steady–state buffer size distribution as a function of the $TH$. To optimize the $TH$, we consider three variables: the probability of having an empty buffer ($\pi_0$), the frame loss probability ($p_L$), and the mean playout rate ($\overline{B}$). To analyze how the $TH$, $\pi_0$, $p_L$, and $\overline{B}$ are related, we performed simulations on a system of buffer size = 100 frames, frame size = 15K bytes, network access rate = 100 Mbps (implying a slot time of 1.2 ms), and the maximal playout rate ($\mu$) = 1/25 frames/slot–time (corresponding to a rate of 30 frames per second).

In Figure 3, $\pi_0$ decreases with the $TH$, as was expected. This is because, the greater the $TH$ is, the faster the playout rate reduces, thus the smaller the probability of having an empty buffer. We also observe that, for any given $TH$, $\pi_0$ increases as the mean arrival rate declines. Figure 4 unsurprisingly illustrates that $p_L$ increases with the $TH$ and Figure 5 demonstrates that the mean playout rate decreases with the $TH$. This is because the larger the $TH$, the faster the playout rate reduces. In addition, the figures also show that both the frame loss probability and the mean playout rate increase with the mean arrival rate.

On the whole, a larger $TH$ which results in a smaller $\pi_0$, implies pausedless playout; whereas a smaller $TH$ implies better playout quality. In what follows, we present a paradigm of the determination of approprite $TH$s under various arrivals.

### 2.3 Formal Description of Algorithm

We now formally present the playout smoothing algorithm employed in the Video Smoother.

[Video Playout Smoothing Algorithm]

(1) Determine a suitable *TH*. Based on the analytic results obtained from the previous subsection, a paradigm of the *TH* determination can be constructed for each case of arrivals. Tables 1 shows the paradigm for the Video Smoothers achieving four different playout qualities under a given traffic characteristic ($\lambda=0.875\mu$).

(2) Determine the playout rate.

```
while (a frame to be playbacked) do
    if (the number of frames in the buffer ≥ TH)
        playout_rate = maximum_playout_rate;
    else
        playout_rate = Dynamic_reduced_rate(TH, i);
```

where the dynamic reduced rate can be selected according to Equations (3).

[End of Algorithm]

## 3. Prototyping System and Results

The prototyping system (see Figure 6) was developed in Intel 80486 personal computers under the MS–Windows environment. The prototyping system is composed of a teleconferencing multimedia application using Intel Smart Video Recorder, WinKing [11] and an Ethernet network. The WinKing package is an implementation of TCP/IP including the API, called Winsock, developed by Institute for Information Industry (III), Taiwan. In particular, the Video Smoother was implemented as a part of WinKing. The simulated frame arrivals are assumed to follow the Poisson process with $\lambda=0.875\mu$. According to the constructed paradigm shown above, we employed an optimal *TH* of 7 for the Video Smoothers to achieve the playout quality of $\pi_0<10^{-3}$, $p_L<10^{-6}$, and $\bar{B}>0.93\mu$.

A motion picture (see Figure 7) was adopted to demonstrate the viability of the Video Smoother. In the experiment, the film was captured every 96 ms with and without the Video Smoother. Figures 7(b) and (a) show a series of scenes taken from the film with and without the Video Smoother, respectively. Without the Video Smoother, part (a) of the figure reveals the playout discontinuity problem. In particular, playout pauses occur in a02–a03 and a10–a11. By contrast, with the Video Smoother, as shown in the part (b), the movements of the bus in Figure 7(b) are much smoother than the ones in Figure 7(a), without delays.

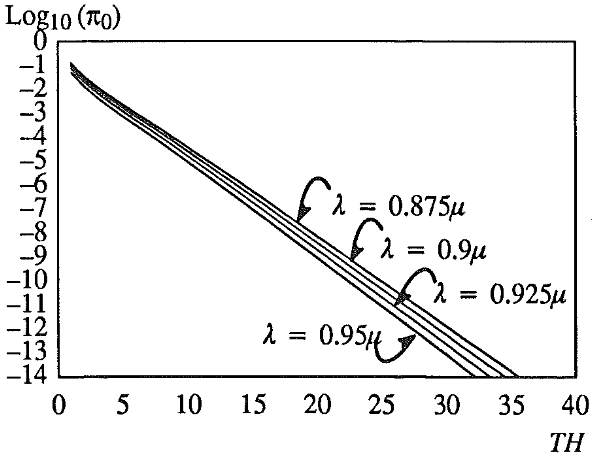


Figure 3. Effect of *TH* on $\pi_0$ under various arrival rates.

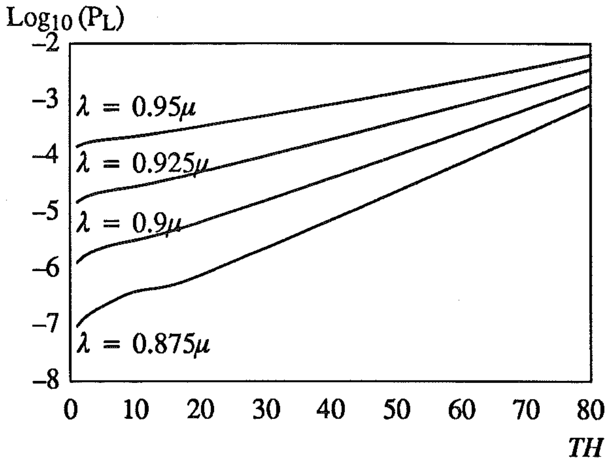


Figure 4. Effect of *TH* on $P_L$ under various arrival rates.

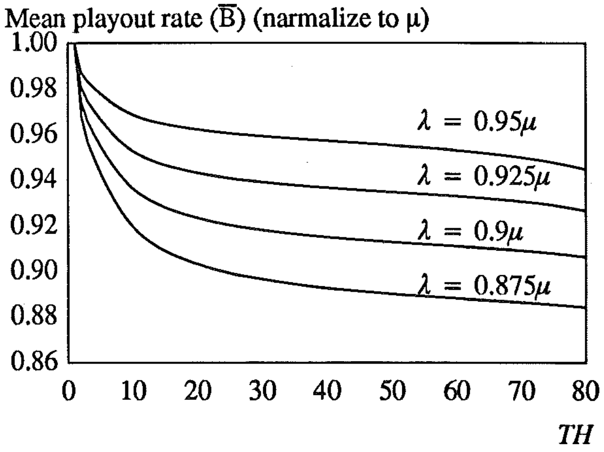


Figure 5. Effect of *TH* on $\bar{B}$ under various arrival rates.

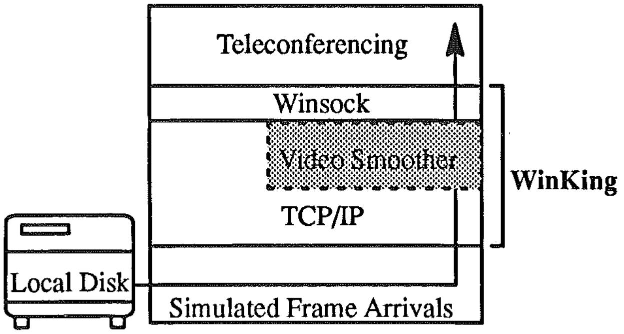


Figure 6. Prototyping system architecture.

Table 1. Recommended *TH*'s assuming Poisson arrivals

| Maximum $\pi_0$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ |
|---|---|---|---|---|
| Maximum $P_L$ | $10^{-6}$ | $10^{-6}$ | $10^{-6}$ | $10^{-6}$ |
| Minnmum $\bar{B}$ | 0.95μ | 0.93μ | 0.92μ | 0.91μ |
| Recommended TH | 4 | 7 | 9 | 12 |

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

### LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

### FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

### E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

**WHAT WILL YOU BUILD?** | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.