Proceedings

21st IEEE Conference on Local Computer Networks

October 13 – 16, 1996

Minneapolis, Minnesota

Sponsored by

IEEE Computer Society Technical Committee on Computer Communications

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

RPX Exhibit 1232

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society Press, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Press Order Number PR07617
IEEE Order Plan Number 96TB100073
Library of Congress Number 83-641144
ISBN 0-8186-7617-5 paper
ISBN 0-8186-7619-1 microfiche
ISSN 0742-1303

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553

Editorial production by Regina Spencer Sipple
Cover design by Joseph Daigle/Studio Productions
Printed in the United States of America by KNI, Inc.

The Institute of Electrical and Electronics Engineers, Inc.

Destination Buffering for Low-Bandwidth Audio Transmissions using Redundancy-Based Error Control

Bert J. Dempsey *Yangkun Zhang*
School of Information and Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3360
{bert,zhany}@ils.unc.edu

Abstract

*Digital audio is becoming increasingly prevalent with the advent of software for Internet telephony and real-time audio playback for World Wide Web browsers. Since audio quality depends on timely delivery of the packet stream, protocol mechanisms must address the control of delays and packet losses in an integrated fashion. Two critical mechanisms for high-quality audio delivery are the receiver buffering strategy and error control. Buffering at the audio receiver is required for continuous playback in the presence of network delay variations (*jitter*), and a number of algorithms have been proposed. Timely recovery of packet loss protects audio quality against network errors. Recent work has suggested the effectiveness of an approach that sends multiple copies of the audio data in consecutive packets so that small burst losses in the network are overcome [3, 8]. In this paper packet-level traces of Internet audioconferencing software were collected over a network path including LANs and a* 28.8 *kbits/s dial-up connection. Using these traces in simulations, receiver buffering strategies for controlling packet jitter and for supporting redundancy-based error control are examined. The study determines that jitter control algorithms will not generally provide adequate buffering when the requirements of error control are included. This observation leads to a proposed modification for one popular jitter control algorithm, and the performance trade-offs are explored.*

Key Words: Packet Voice, Error Control, Redundancy, Jitter Control.

1. Introduction

Digital audio delivered over packet-switched networks continues to expand into an exciting new application area as audio capability on the desktop becomes commonplace. Audio broadcasting and conferencing applications over the Internet multicast backbone, the *Mbone*, have led to the development of mature public domain packet audio tools such as *vat*[12], *nevot*[19], and *rat*[8]. Commercial Internet audio products now cover telephony and conferencing [16, 21] as well as one-way audio streams delivered to World Wide Web clients [17, 23].

This paper examines the problem of delivering high-quality audio across low-bandwidth (e.g., 28.8 kbits/s) dial-up connections. Such dial-up connections represent a low-cost, widely available solution for making Internet-based applications available to the increasing numbers of users located at off-campus sites such as homes and remote offices. With regard to audio, campus-to-home dial-up connections can expand conferences or broadcasts originating within an enterprise network or increase the audience for wide-area audio feeds through reflector nodes [20]. In all network delivery scenarios, however, the sensitivity of digital audio to network delays and loss must be considered, and transmission over low-bandwidth links poses special challenges.

- Low-bandwidth links imply the use of low bit-rate encodings or compression. While compression at the link layer or physical layer may improve transfer rates, low-rate encodings are essential to control bandwidth usage. Software codecs for low-bandwidth Internet audio are available. Two of the most widespread are the Group Speciale Mobile (GSM) standard and the Linear Predictive Coder (LPC) standard. The current *vat* Internet audio tool, for example, incorporates a 17 kbits/s GSM encoder and a 4.8 kbits/s LPC encoder.

0-8186-7617-5/96 $05.00 © 1996 IEEE

- Loss sensitivity is high for low bit-rate audio streams due to the high information density of the encoded stream and the common practice of transmitting relatively long audio intervals in each packet. Putting long intervals of speech into each packet is motivated by the desire to avoid consuming a significant fraction of the network bandwidth with packet headers. For example, a 17 kbits/s data rate generates only 170 bytes in an 80 ms period. If this data is carried as a Real-Time Protocol (RTP) [20] packet over UDP, each audio packet includes 40 bytes of protocol header before link layer encapsulation (12 bytes for the RTP header, 8 for UDP, and 20 for IP). If the packet size were 40 ms, one-third of the total bandwidth used to transmit the audio stream would consist of protocol overhead. As a result Internet-based audio protocols typically use 80 ms or larger packets for low-bandwidth links, in contrast with 20 or 40 ms packets for higher bandwidth links.

  Loss sensitivity is difficult to quantify and heavily dependent on the encoding technique. With large packets, however, even single-packet losses are often significant for quality and/or intelligibility at the receiver [7]. Since interpolation techniques to mask losses at the receiver do not work well for large packets [9], packet-level error control is attractive for low-bandwidth audio.

- Natural interactive conversation is difficult for users to maintain when the roundtrip delay experienced exceeds a few hundred milliseconds. While the threshold involves subjective judgments of quality, studies have reported minimal detrimental effect for one-way delays in the 200-300 ms range [4, 22, 13] and users reporting annoyance with one-way delays in the 320-440 ms range [22, 10]. However, [10] reports that, while users complained about delays in the latter range, they also compensated effectively for them in the context of videoconferencing.

  To meet the requirement for bounded roundtrip delays, audio protocols focus on the one-way end-to-end delay of the audio samples. The components of end-to-end delay include: the time audio data spends at the sender during packetization, network delays, buffering and hardware delays at the audio receiver. The component delays most readily controlled by the audio protocol are the packet size used for network transmission and the receiver buffering algorithm.

  In the campus-to-home network scenario, end-to-end delays tend to be long due to the transmission latency over the dial-up link and the long packetization time at the transmitter. Long end-to-end delays place pressure on the receiver buffering algorithm to minimize the buffering time, if interactive or near-interactive delays are to be achieved.

In this study two popular Internet audio tools are used to collect packet-level traces of audio streams transmitted over a campus-to-home connection, i.e., across a campus network followed by a modem-based dial-up connection. With these traces, a simulation study is constructed to investigate trade-offs in delay and loss during audio playback. In addition to jitter control, the simulations are used to extrapolate the effects of using a version of the redundancy-based error control scheme proposed in [3] for protecting the audio stream from network loss. Support of error control was not considered in the design of current jitter buffer algorithms, and the simulation results demonstrate that modification is required to support this form of error control effectively. Modification of one well-known jitter algorithm is proposed and the performance trade-offs evaluated.

The paper is organized as follows. Section 2 discusses key points of terminology, background, and related work on Internet audio transmissions. Section 3 describes how the packet-level traces of audio transmissions for this study were obtained. Section 4 analyzes the conventional receiver buffering algorithm and the proposed modification using trace-driven simulations of the audio playback. Section 5 presents conclusions.

2. Background and Related Work

In order to discuss related work, it is important to establish a consistent terminology, which can be derived from the timeline for audio playback in Figure 1. Packet speech consists of bursts of activity, *talkspurts*, separated by pauses, *silence periods*. As shown in Figure 1, we denote the transmission time of the ith packet in a talkspurt as time t_i and its arrival time at the audio receiver as a_i. The packet size of the ith packet is defined as $\Delta_i = t_i - t_{i-1}$, and the network delay

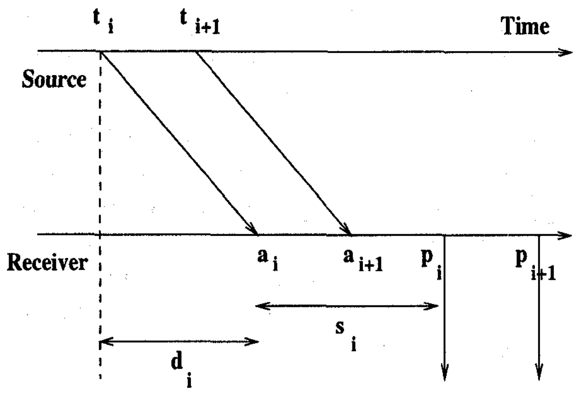

Figure 1. Playback Timeline for Audio.

is $d_i = a_i - t_i$. At the audio receiver, each packet in the talkspurt has a scheduled *playback time*, p_i. The end-to-end delay of the ith packet is the time between placement of the first audio sample into the ith packet at the transmitter and the beginning of the playback of that packet at the audio receiver,

$$e_i = p_i - t_i + \Delta_i \qquad (1)$$

The time spent by the ith packet in the receiving buffer before being submitted to the audio playout device is the *slack* of the ith packet

$$s_i = p_i - a_i. \qquad (2)$$

The purpose of the buffering algorithm at the receiver is to calculate a *hold time* before the first packet in a talkspurt is played out. The hold time, s_1, is crucial since it determines the playback time for all subsequent packets. For jitter control, the *hold time* calculation has the competing goals of minimizing the extra delay introduced by buffering while ensuring that each packet has a positive slack.

The playback time for each packet is usually determined by a timestamp, either implicit or explicit, carried in the packet and an estimate of the network delay. In the context of jitter control, several methods for network delay estimation have been proposed [15, 18, 2, 1, 14]. Given an estimator for network delay, most audio protocols determine the playback time of each packet as follows. For the first packet in a talkspurt

$$p_1 = t_1 + \hat{d}_i + 4\hat{v}_i \qquad (3)$$

where $\hat{d}_i$ and $\hat{v}_i$ are estimates of the mean and variation of the network delay. The use of $4\hat{v}_i$ is motivated by the work on TCP roundtrip time estimation done by Van Jacobson [11]. For the other packets in a talkspurt,

$$p_i = p_1 + t_i - t_1 \qquad (4)$$

Jitter buffering strategies differ only in the way they estimate $\hat{d}_i$ in Equation 3.

Given stochastic network delays, buffering algorithms attempt to adapt the hold time to changing network delay and delay variations. The accuracy of a measuring a packet's network delay is itself dependent on factors such as timestamp granularity and local clock drift. Adaptation to changes in the network delay requires some form of filtering past samples, e.g., using a low-pass filter modeled after the TCP roundtrip time estimator. For wide-area Internet transmissions, the effects of sudden large changes in the delay, *delay spikes*, can skew network delay estimates badly. The study in [18] develops a network delay estimator (Algorithm 4) that explicitly considers the phenomenon of delay spikes. Using wide-area Internet audio traces, this estimator is shown to perform somewhat better than three other buffering algorithms for jitter control. Using a similar motivation, [2] also gives an algorithm designed to recover quickly from sudden delay spikes and presents evidence of good performance.

Work in timely recovery of network losses has advanced rapidly in recent years. Retransmission-based schemes have been proposed and shown effective under some conditions [5, 6], but, in the campus-to-home audio scenario, retransmission is not viable for interactive traffic due to high roundtrip delays. Receiver-only techniques refer to those that attempt to mask lost packets using interpolation at the receiver, such as waveform or silence substitution. The large packet sizes associated with low-bandwidth audio, however, render these techniques much less effective than with shorter packets [9]. With redundant transmissions, a source sends multiple copies of the audio data so that the impact of packet losses is mitigated by the arrival of at least one copy of the data at the receiver. Redundancy avoids the latency penalty of retransmission, at the expense of some additional network bandwidth, computation, and delay.

This study focuses on redundancy-based error control since this technique is applicable to our network scenario and has demonstrated promise for maintaining good audio quality over wide-area lossy paths [3, 8]. For low-bandwidth links, this technique could be used

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.