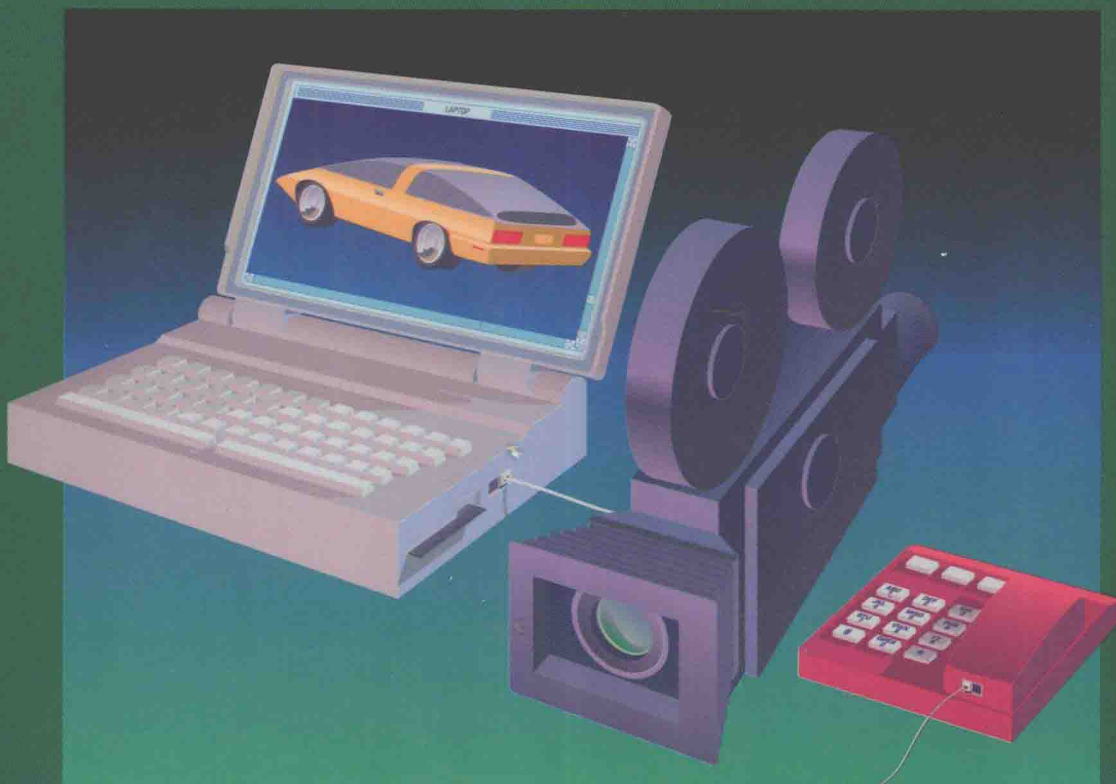
PROCEEDINGS OF THE

INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS



MAY 14-19, 1994 • BOSTON, MASSACHUSETTS

Sponsored by the IEEE Computer Society Task Force on Multimedia Computing

IEEE COMPUTER SOCIETY PRESS

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Proceedings of the

International Conference on Multimedia Computing and Systems

May 14 – 19, 1994 Boston, Massachusetts

Sponsored by

IEEE Computer Society Task Force on Multimedia Computing

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

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

IEEE Computer Society Press Order Number 5530-02
Library of Congress Number 93-80789
IEEE Catalog Number 94TH0631-2
ISBN 0-8186-5530-5 (paper)
ISBN 0-8186-5531-3 (microfiche)

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 Mary E. Kavanaugh
Cover design by Alexander Torres
Printed in the United States of America by McNaughton & Gunn

The Institute of Electrical and Electronics Engineers, Inc.

General Chair's Message

The convergence of several technologies for multimedia applications is now commonplace. One of the consequences of this convergence is the need to be interdisciplinary. Mastering a specific discipline must be coupled with being conversant with other fields. In fact, today an ensemble is required to explore and construct new products and systems which meet human needs.

This intercultural cooperation among professionals is becoming unavoidable since problems and applications do not align themselves well along disciplinary lines — they need the collective minds of many. I am confident that this first conference on Multimedia Computing and Systems will deepen as well as broaden the know-how of participants: the users, specialists and system builders in the multimedia arena. The conference committee also hopes that the event itself and the following material will generate creative discussions among people with a variety of backgrounds.

Les Belady
Conference Chair
Mitsubishi Electric Research Laboratories
Cambridge, Massachusetts
E-mail: belady@merl.com

Program Chairs' Message

During recent years the term "multimedia" has become a catchword which comprises everything from the simplest to the most sophisticated in hardware, software, research, development and products. As a first endeavor to understand what multimedia is we can say:

> "A multimedia system is characterized by the computer-controlled integrated generation, manipulation, presentation, storage and communication of independent digital information. This information is most often coded in continuous-time dependent-media (e.g., audio, video) as well as discrete-time independent-media (e.g., text, graphics)." *

This notion is for most of us in the multimedia community a basis for many systems research and development activities. The feedback in terms of submissions to this conference also showed that all systems aspects are better understood than applications-related issues. However, we need to better understand these applications-related topics and their respective social, cultural, and legal implications.

Unfortunately the results of work on multimedia issues seem to be widely spread. As researchers in this area, we have had to be present at very different events of the various "traditional" computing societies. We've encountered excellent work in the networking, graphics, and user-interface domains through many non-related events and publications. Through the taskforce initiated by Tadao Ichikawa and supported actively by many of us in these fields, the IEEE Computer Society has started to focus these activities. Thus we now we have the IEEE Multimedia Magazine and are presenting this first IEEE International Conference on Multimedia Computing and Systems, in Boston.

The high quality of the selected papers and the wide attraction of this event shows that we are working towards the correct goal. We hope you will find this event both exciting and stimulating and welcome you to the conference!

Scott Stevens
Program Co-Chair
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
E-mail: sms@sei.cmu.edu

Ralf Steinmetz
Program Co-Chair
IBM European Networking Center
Distributed Multimedia Solutions
Heidelberg, Germany
E-mail:
kiwi.heidelbg.ibm.com!rst@ibmpa.awdpa.ibm.com
or steinmet@dhdibmip.bitnet

* Steinmetz, *Multimedia Technology: Fundamentals and Introduction,* Springer-Verlag, 1993.

Organizing and Program Committees

Conference Chair

Laszlo A. Belady
Mitsubishi Electric Research
USA

Program Co-Chairs

Scott M. Stevens
Carnegie Mellon University
USA

Ralf Steinmetz
IBM European Network Center
Germany

Tutorial Chair

Erich Neuhold
T.H. Darmstadt
Germany

Publicity Chair

John Buford
U. Massachusetts - Lowell
USA

Exhibits Chair

William Lambert
Horizon Research
USA

Publication Chair

Tibor Vais
Compuserve
USA

Registration & Finance Chair

Joseph Boykin
GTE Laboratories
USA

Local Arrangements Chair

Michael Bove
MIT Media Lab
USA

[*continued next page*]

Program Committee

Table of Contents

Session 12 — Database and Information Systems

Session 13 — Capture and Creation of Content II

Session 14 — Hardware Architecture

Session 15 — Programming

Session 16 — Video Compression and Processing

Panel III — Immersive Environments: A Physical Approach to the Multimedia Experience

Moderator: *Allison Druin, Research Scientist, N.Y.U. Media Research Laboratory*

Poster Session

Keynote Address

***Presenter*:** Nicholas Negroponte, Director, MIT Media Laboratory

Session 1

Scheduling and Synchronization

Synchronization Architecture and Protocols for a Multimedia News Service Application *

L. Lamont and N. D. Georganas

Multimedia Communications Research Laboratory
Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

Abstract

In this paper, we present a complete software control architecture for synchronizing multiple data streams generated from distributed media-storing database servers without the use of a global clock. Traffic prediction and scheduling is performed by the independent sources for the remote presentation. Certain compensation mechanisms at the receiver are also necessary because of the random network delays. A stream synchronization protocol (SSP) allows for synchronization recovery to preserve a high quality multimedia display at the receiver. SSP uses Synchronization Quality of Service parameters to guarantee the simultaneous delivery of the different types of data streams. The various components of the architecture interact to provide real-time multimedia functions such as those found on VCRs.

1 Introduction

The evolution of multimedia technologies and broadband networks has significantly contributed to the emergence of new multimedia applications such as remote document creation/editing, multimedia news services, etc. The presentation of such applications is characterized by the spatio-temporal integration of the multimedia objects [1]. The temporal integration produces the required presentation sequence of the objects. Multimedia integration can be described by scenarios, which represent the temporal relationships among the data objects involved in the document. In a multimedia newspaper application, the scenario is designed and stored along with the articles in a remote database, while in a remote document creation application [2] [3], the scenario is created by users using the appropriate tools at their workstation.

In this paper we focus on a target presentational application, namely a multimedia news service. The application allows users to browse through items of current-affairs such as multimedia newspaper articles and to select a specific article for viewing. The articles are stored in a remote multimedia database. The news articles are composed of media such as text, images, graphics, animations, video and audio. Each multimedia article has an associated temporal presentation scenario that is also stored in the multimedia database. Each medium resides on its own medium-server. The media-servers are of different types, such as image server, voice server, text and graphic server and video server. Through multiple servers, activities such as multimedia data retrieval and document delivery are performed continuously in real-time. Data in different media are transferred over different parallel connections or streams according to their traffic characteristics and transmission requirements. For instance, voice is transferred on a 64 kbits/sec connection while high quality uncompressed digitized video requires a connection of more than 30Mbits/sec. In order to support this real-time application, a real-time Stream Synchronization Protocol (SSP) is required to preserve the synchronization among the data carried over different connections.

Several synchronization control methods have been introduced for solving the problem of stream synchronization. In [4], Nicolaou presents a scheme for implementing synchronization among the related streams by inserting synchronization points in each individual stream. In [5], Steinmentz introduced the concept of Logic Data Unit (LDU). A continuous media data stream is viewed as a sequence of LDUs. The synchronization control is performed on a LDU. In [6], Ferrari proposes a delay jitter control scheme that insures media synchronization as long as a bound on delay can be guaranteed and that the source and destination clocks are kept in synchrony. Every node on the transmission path in the network is involved in the synchronization control by regulating the random delay which has been introduced to an arriving packet by the previous hop. It might be difficult to implement this scheme on an ATM network because it introduces too much workload on the intermediate nodes that carry thousands of connections. In [7], Escobar et al. address the flow synchronization control protocol by implementing an end-to-end synchronization delay. The random delay experienced by each packet is regulated at the desti-

*This work was supported by a grant from the Canadian Institute for Telecommunication Research(CITR), under the Networks of Centers of Excellence Program of the Canadian Government.

0-8186-5530-5/94 $3.00 © 1994 IEEE

nation node. The actual delay experienced by every data packet cannot exceed the preset end-to-end delay for the network connection and a global accurate clock is required. In [8], Little et al. propose computing schedules before transmitting the data packets. The multimedia traffic runs according to the schedule to reach the destination in time for the presentation. However, because of the random network disturbance introduced by the networks, scheduling and predicting the traffic is not sufficient to maintain a simultaneous data delivery. Therefore, a compensation mechanism at the receiver is necessary when synchronization errors occur. In [9], Rangan et al. present an intermedia synchronization techniques for multimedia on-demand services. The centralized multimedia server uses feedback units transmitted by the receivers to detect the asynchronies among them. Correction of the asynchrony is made at the source (the multimedia server) by speeding up or slowing down the traffic on the various media streams. A real-time synchronization method is needed to display the document in real-time and to preserve the synchronization among the different streams. When using this scheme, it might be difficult to detect and correct the asynchronies before the user can notice them when the network is heavily loaded.

In this paper, we present a complete software control system for synchronizing multiple data streams generated from distributed media-storing database servers without using a global clock. Besides the traffic prediction and scheduling of the object delivery, synchronization recovery is performed at the receiver before the playback of the multiple data streams. A stream synchronization protocol (SSP) prevents the synchronization errors from being rampant in the display, even during network fluctuation. The SSP uses the skew tolerance parameters [10] to guarantee the control for the different types of application data streams. We also present a set of distributed algorithms for some of the real-time multimedia user interactions.

This paper is organized as follows. In the next section we introduce the architecture for our multimedia news service application and we discuss the scheduling strategies for the real-time delivery of the multimedia objects. In section 3, the Stream Synchronization Protocol is presented. In section 4, we present some of the user interaction algorithms and in section 5 the conclusions.

2 Synchronization control of independent sources in a real-time multimedia newspaper application

In the multimedia newspaper application, when the user requests a news document, the request is transmitted to the newspaper database (see Fig1.). The database sends the presentational scenario to all the media-servers that are involved in this document and to the user who made the request. The media-servers which are supplying the data objects participate in the temporal integration, by scheduling their object delivery. Synchronization among the independent media-

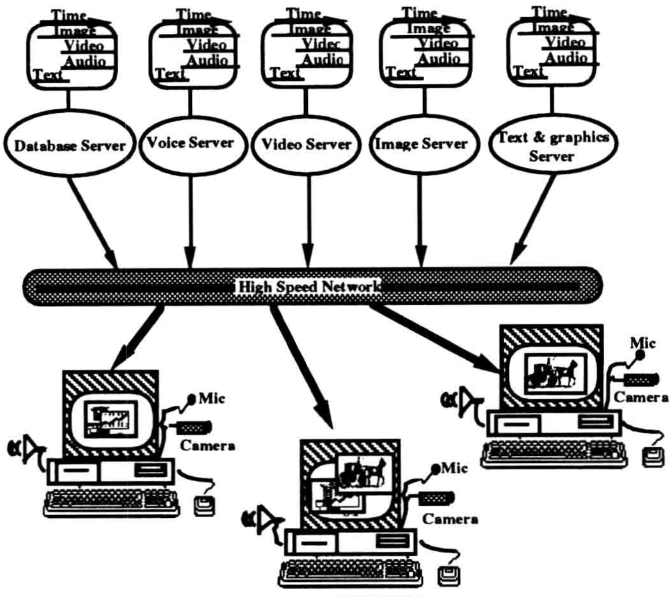


Figure 1: A Distributed Multimedia Newspaper Application

servers is required. According to a given scenario, the distributed media-servers will co-operatively work together to negotiate the 'remote' temporal presentation of the multimedia objects. The Temporal Scheduler Controller (TSC) process is responsible for scheduling the earliest time for the presentation of the object at the remote workstation. A TSC process exists on each media-server that is involved in the document (see Fig2.). The TSC that is involved in the presentation of the first object will determine when it can start transmitting the object. If n multimedia objects are to be started at the same time, all the media-servers involved will determine their starting time. The TSC will also need to know the network delay between the media-server and the user's site in order to complete the scheduling. The Media Synchronization Controller (MSC) process is responsible for opening up a connection to the user's site with the proper transmission characteristics. A MSC process exists on each media-server that is involved in the document. There exists a target MSC process for each stream that is setup at the user's site. As soon as a target MSC process is created, it knows about the original scenario. The target MSC is responsible for returning the network delay between the media-server and the user's site. The TSC broadcasts the network delay information, the duration of its objects and the starting time to all the other media-servers involved. The starting time is determined by the server who can start the latest. The data transmitted by each media-server does not need markers or headers to indicate the simultaneous temporal relationship [4][11] since the target MSCs all have the presentation scenario and they can therefore deduce which data streams should be received simultaneously and how many packets should be received in an interval of time (speed of the connection and

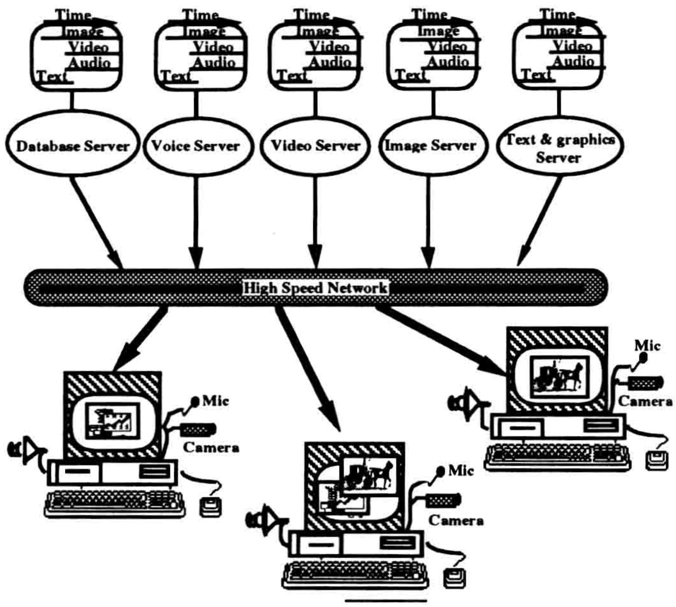


Figure 1: A Distributed Multimedia Newspaper Application

Media		Mode, Application	QOS
video	animation	correlated	+/- 120 ms
	audio	lip synchronization	+/- 80 ms
	image	overlay	+/- 240 ms
		non overlay	+/- 500 ms
	text	overlay	+/- 240 ms
		non overlay	+/- 500 ms
audio	animation	event correlation (e.g. dancing)	+/- 80 ms
	audio	tightly coupled (stereo)	+/- 11 us
		loosely coupled (dialog mode with various participants)	+/- 120 ms
		loosely coupled (e.g. backgroung music)	+/- 500 ms
	image	tightly coupled (e.g. music with notes)	+/- 5 ms
		loosely coupled (e.g. slide show)	+/- 500 ms
	text	text annotation	+/- 240 ms
	pointer	audio relates to showed item	- 500 ms + 750 ms

Table 1: Quality of Service for synchronization purposes

presentation duration of that object).

3 The stream synchronization protocol

Scheduling and predicting the traffic are not sufficient to maintain a simultaneous data delivery since the networks introduce random network disturbances. Random network delay destroys the continuity of the data stream by introducing gaps and jitters during the data transmission.Therefore certain compensations at the receiver are necessary when synchronization errors occur [12][13]. A Stream Synchronization Protocol (SSP) is discussed in this section. Communication between the target MSCs is required to indicate the first packet that has been received on a stream and when synchronization error recovery is needed because of late packet arrival (packets not being received in the allowable time frame). The synchronization errors that can be tolerated by human perception vary in different application scenarios. For instance, from the experiments conducted recently by Steinmetz in IBM Heidelberg [10], 120msec mismatching in lyp-synchronization between an audio and video stream can be perceived as disturbing by the application users. On the other hand, if text annotation is played out 200msec earlier than voice, the presentation is still well accepted. We call these application defined parameters "skew tolerance parameters" or QoS values. In [10] a set of synchronization QoS values are provided. They define the acceptable synchronization boundaries for various application scenarios. Table1 shows the QoS values or skew tolerance parameters that are provided by Steinmetz's experiments.

In our architecture, skew tolerance parameters determine the synchronization goal that should be achieved by the target MSC processes. Different classes of "QoS" parameters are defined and stored in the presentational scenario. The target MSCs use the QoS parameters to determine the mismatching tolerance time (skew tolerance parameter) admitted by the application before applying synchronization error recovery. For instance in Fig.3a, the MSCs involved in the first activity of the scenario, get the video duration time length T_v, the starting time difference for the text and voice activity T_s and the text and audio duration time T_a & T_t from the presentational scenario. Tolerance times between the audio and the text streams T_{d1}, and between the video and the audio/text streams T_{d2} are also retrieved. The receiving MSCs accept late arrivals as long as the mismatching does not exceed the tolerance T_{d1} and T_{d2}. Late arrivals in at least one of the streams cause gaps in the display when one of these streams is used in conjunction with other streams in the following activity (see Fig.3b). In order to filter the gaps and thus decrease the mismatching at the end of an activity, an "intentional delay" T_i [12][13] is either placed on the streams that have no traffic flow in the previous activity or on the streams that are delayed by an amount $T_d < T_{dmax}$, where T_{dmax} corresponds to the stream that endured the longest delay. In the latter case the intentional delay T_i for the streams that are delayed by T_d will correspond to $T_{dmax} - T_d$. While the intentional delay T_i is in effect, each MSC involved should help decrease the interruption caused by the filling gaps. The MSC associated with the video, fills in the gaps by repeating the last frame received in the previous activity while the other MSCs simply pause their display to the screen.

The idea of adding an intentional delay at the end of an activity allows for the synchronization to be recoved without having to discard or skip late packets. This method, which can also be refered as a "time

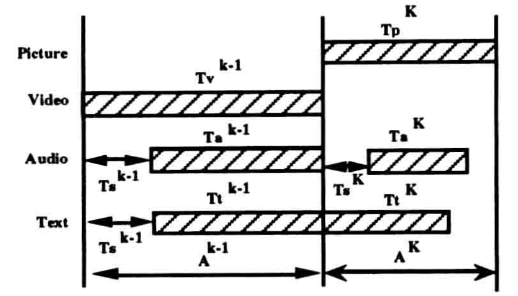


Figure 3.a: Parameters in Activity Requirement

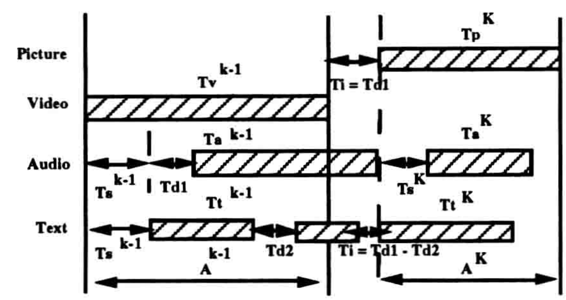


Figure 3.b: Activity Mismatch and Recovery

expanding policy", is very well suited for applications such as multimedia news on-demand services, since it ensures a high quality multimedia presentation at the receiver.

3.1 Communication Between Target MSC Processes

Fig. 4 shows how the SSP is executed. Communication between the target MSCs is required to indicate the first packet that has been received on a stream and to notify each MSC of the delays, if any, encountered on each stream. The MSC that is involved in the presentation of an object notifies the MSCs involved in the scenario that a packet has arrived, as soon as the first packet is received on its stream. If n multimedia objects are to be started at the same time, then only the MSC that receives the first packet is required to advise the MSCs involved in the scenario. The MSCs that receive the notification ensure that the packets that arrive on their stream do not exceed the skew tolerance parameter defined between related streams. If packets exceed the skew tolerance parameters, the application is aborted after the user is notified that the skew tolerance has been exceeded. If packets are delayed on some streams and they do not exceed the predefined skew tolerance parameters, the involved MSCs notify all the receiving MSCs, involved in the scenario, of the delay encountered along with the current activity number (taken from the presentation scenario). The receiving MSC use the delays to calculate the intentional delay for the next presentation and/or use them to calculate the time to fill in the gap until the

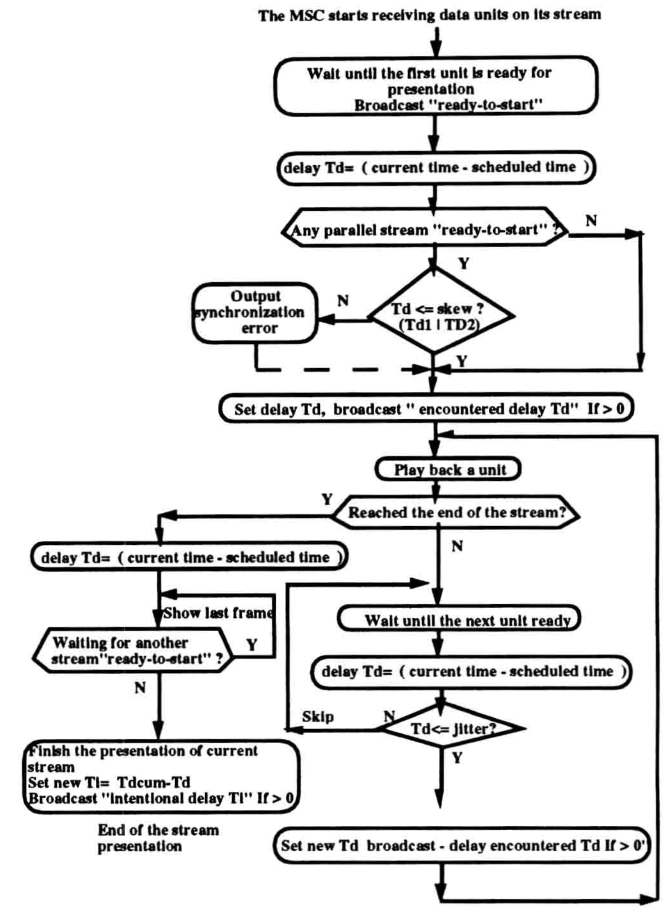


Figure 4: SSP Flow Chart

next activity starts. In some instances an MSC associated with a continuous media (video and audio) might be required to skip a few frames to compensate for large network jitters. The jitter parameter for each stream is set when a connection is established between a media-server and the user's site. The intentional delay or the gap filling for each stream is calculated in the following way:

1. The MSC that does not encounter any delay, uses the longest cumulative delay Td_{cum} as the intentional delay T_i or the gap filling delay T_{dg}.

2. The MSC that encounters some delays, determines if there are any cumulative delays longer that the one it has encountered. If there are, it takes the longest cumulative delay Td_{cum} and subtracts its own cumulative delay T_d to obtain the final delay $T_{df} = T_{dcum} - T_d$.

3. The MSC that encountered the longest cumulative delay T_{dcum}, does not add any intentional delay T_i or gap filling delay T_{dg} to the next activity.

When packets exceed the skew tolerance parameters, a less rigid policy may be applied. The application is aborted only when the skew between related