The Third AFSS(1998). 246－251

Design and Evaluation of a Distributed Multimedia Synchronization Algorithm based on the Fuzzy Logic

Sun-Jin Oh[a], Ihn-Han Bae[b]

a Department of Computer Science, Pohang Sunlin College
146-1, Chogok-dong, Heunghae-eup, Pohang, Kyungbuk, 791-940, Korea
Tel:+82-562-61-1992, Fax:+82-562-61-9455, E-mail:sjoh@www.sunlin.ac.kr
b Department of Computer Engineering, Catholic University of Taegu-Hyosung
330, Kumrak, Hayang, Kyungsan, Kyungbuk, 712-702, Korea
Tel:+82-53-850-2742, Fax:+82-53-850-2704, E-mail:ihbae@cuth.cataegu.ac.kr

Abstract

The basic requirements of a distributed multimedia system are intramedia synchronization which asks the strict delay and jitter for the check period of media buffer and the scaling duration with periodic continuous media such as audio and video media, and intermedia synchronization that needs the constraint for relative time relations among them when several media are presented in parallel.

In this paper, a distributed multimedia synchronization algorithm based on the fuzzy logic is presented and the performance is evaluated through simulation. Intramedia synchronization algorithm uses the media scaling techniques and intermedia synchronization algorithm uses variable service rates on the basis of fuzzy logic to solve the multimedia synchronization problem.

Keywords: Distributed Multimedia Synchronization, Media scaling, Variable service rate, Fuzzy logic

1. Introduction

With the development of multimedia processing techniques and high-speed communication network using computer technologies, studies on various distributed multimedia applications via computer communication network, such as video conference system and multimedia information search system, are now making a great progress in this research fields. There are still several problems to implement such multimedia applications. One of them is the multimedia synchronization technique to maintain time relations among multimedia data. Such multimedia synchronization is achieved by controlling data transmission artificially to support the quality of service(QoS) of multimedia data properly. Since the requirements of QoS depends on both applications and characteristics of media, the appropriate control of the artificial data transmission for synchronization should be allowed.

To solve the intramedia synchronization problem, receiver transmits the feedback unit of scaling messages to sender by appling fuzzy logic based on the buffer state and arrival rate of a media stream. After then, media scaling is performed by a scaling mechanism at sender and the scaled media stream is transmitted to receiver. That is called as intramedia synchronization algorithm. In order to solve the intermedia synchronization problem, receiver checks the differences of each media stream's relative time stamp(RTS) and applies the variable service rates to the slave processor by using fuzzy logic based on the slave media's relative buffer state and synchronous time compared with master media's. It is called as intermedia synchronization algorithm. In this paper, both intramedia and intermedia synchronization algorithms are proposed and their performance is evaluated through simulation.

The remainder of this paper is organized as follows: Section 2 summaries related works for multimedia synchronization. Section 3 surveys scaling methods. Section 4 presents the proposed intramedia synchronization algorithm. The proposed intermedia synchronization algorithm is presented in section 5 and these algorithms are evaluated in section 6. Finally, section 7 concludes the paper.

2. Multimedia synchronization

There are two kinds of multimedia synchronization techniques such as intramedia synchronization and intermedia synchronization. Intramedia synchronization is related to the transmission of each media object within the processing time. For the intramedia synchronization, transmission schedule is generated by computing the completion time of transmission from completion time of process with consideration of network delay. An appropriate scheme that can correct this problem is required at destinations. On the other hand, intermedia synchronization is demanded when there exists some relative time relationships among two media objects in a multimedia object.

Related researches for multimedia synchronization are as follows: Little et al. [5] use the scheme that controls the buffer level at receiver site. Synchronization problem is solved by dropping or duplicating frames according to the transmission rates in the buffer of receiver site. Anderson et al. [4] describe the local synchronization algorithm for recovering from asynchronism among interrupt-driven media I/O devices connected to a continuous media (CM) I/O server. Rangan et al. [1] present feedback techniques for synchronization in distributed multimedia systems.

3. Media Scaling

Scaling is to subsample a data stream and presents some fraction of its original contents only. In general, scaling can be done at either the source or the sink of a stream. Frame rate reduction, for example, is usually performed at the source, whereas hierarchical decoding is a typical scaling method applied to the sink. Scaling methods used in a multimedia transport system can be classified as follows [2, 3]:

- Transparent scaling methods can be applied independently from the upper protocol and application layers, that is, the transport system scales the media on its own. Transparent scaling is usually achieved by dropping some portions of the data stream. These portions - single frames or substreams - need to be identifiable by the transport system.
- Non-transparent scaling methods requires an interaction of the transport system with the upper layers. In particular, this kind of scaling implies a modification of the media stream before it is presented to the transport layer. For the distribution of media parameters of the coding algorithm, stored media can be scaled by recording a stream that was previously encoded in a different format.

In a multimedia system, scaling can be applied to a couple of different media types, for example, video and audio. For audio, scaling is usually difficult because presenting only a fraction of the original data is easily noticed by the listener.. For video stream, users are typically much less sensitive to the quality reductions. For this reason and due to their high bandwidth requirements, video streams are predestined for scaling. The applicability of a specific scaling method depends strongly on the underlying compression technique. There are several domains of a video signal to which scalings can be applied such as temporal, spatial, frequency, amplitudinal and color space scaling[2, 3].

4. Intramedia synchronization algorithm

In this paper, a distributed intramedia synchronization algorithm, operated in the environment of distributed multimedia system as shown in Fig. 1, is designed, where, each media data is transmitted from media server to clients.

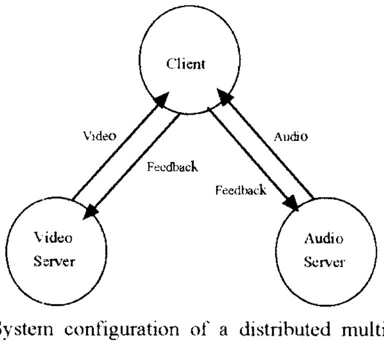

Fig. 1. System configuration of a distributed multimedia

The clients monitor the buffer state of each media periodically. If a media buffer state is overrun or starvation, the client sends a scaling message as a feedback unit to the media server. One practical application of this system is video on demand (VOD) applications.

The intramedia synchronization algorithm monitors each media buffers at clients, and checks buffer occupancy count (BOC) periodically, where BOC is the information which affects variations of data rates directly. If the arrival rate of incoming media streams from a communication channel is higher than the service rate of outgoing media streams for a processor, the media buffer is overrun. Otherwise, the media buffer starves (see Fig. 2). The primary objective of intramedia synchronization is to manage media buffers so that starvation or overrun doesn't occur.

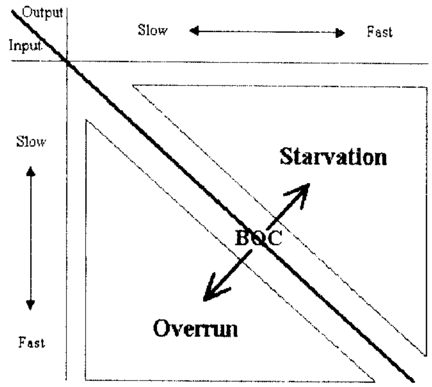

Fig. 2. The relationship of I/O data rates with BOC

We use the media buffer model shown in Fig. 3 for intramedia synchronization. Fig. 3 represents the receiving buffer of a media stream, where s is a start_count that specifies how much streams should be received on the channel before the stream begins presentation to prevent initial starvation, an appropriate value of start_count can be calculated from bounded jitter, and l_1, l_2, l_3 and l_4 are thresholds of buffer length that represent starvation, starvation warning, overrun warning, and overrun states respectively.

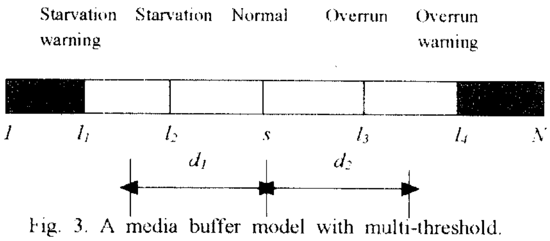

Fig. 3. A media buffer model with multi-threshold.

In the distributed multimedia system, if the current state of receiving buffer is starvation warning or overrun warning, the client send a scaling message to the media server. The media server receiving the scaling message sends r-scaled media streams to the client, where $0<r<1$ is the relaxation parameters which represent relative scaling degree (scaled stream size/original stream size). A relaxation parameter is chosen according to the current state of media buffer. Audio and video streams are scaled by upper and network scaling mechanisms respectively. We compute BOC check period according to the current state of each media buffer using the buffer model above and the media scaling mechanisms. The overhead of intramedia synchronization is reduced by checking BOC according to a compared period. If a media buffer is in starvation warning state or overrun warning state, the media buffer is returned to normal state by the media scalings.

If a current state of media buffer is normal, BOC check period is as in Eq. (1):

$$C_n = Min(\frac{l_3 - l_1}{2(\mu - \lambda_{min})}, \frac{l_4 - l_2}{2(\lambda_{max} - \mu)})\ sec, \quad (1)$$

where λ_{max} and λ_{min} are the maximum and the minimum arrival rates of a media stream according to traffics of a communication channel, and μ is service rate of a media stream by a media processor.

If the current state of media buffer is starvation warning, the client sends a scaling message to the media server, and the media server sends scaled media streams to the client. Then, BOC check periods in starvation warning state are as in Eq. (2):

$$C_s = \frac{l_3 - l_1}{2(\lambda' - \mu)}\ sec, \quad (2)$$

where $\lambda' = \lambda / r$. When the client sends a scaling message to the media server in the state, REL value of the message is r and DUR value of the message is C_s.

If the current state of media buffer is overrun warning, the client sends a scaling message to the media server, and the media server sends scaled media streams to the client. Then, BOC check periods in overrun warning state are as in Eq. (3):

$$C_o = \frac{l_4 - l_2}{2(\mu - \lambda'')}\ sec, \quad (3)$$

where $\lambda'' = \lambda \times r$. When the client sends a scaling message to the media server in the state, REL value of the message is r and DUR value of the message is C_o.

In this paper, the fuzzy logic is used to solve the intramedia synchronization, using multi-level media scaling according to the client's media buffer state and arrival rates of media stream. Rules for scaling relaxation control to maintain the intramedia synchronization is shown in table 1.

Table 1. Rules for scaling relaxation control.

MBS / MAR	SW	NO	OW
L	SR	WR	WR
M	MR	NR	MR
H	WR	WR	SR

the number of rules = 9

(Input variable)
MBS - media buffer state
MAR - media arrival rate
(Output variable)
REL - relaxation
(Linguistic variable)
L - low M - medium H - high NO - normal
SW - starvation warning OW - overrun warning
NR - no relaxation WR - weak relaxation
MR - medium relaxation SR - strong relaxation

The membership functions for media buffer state, arrival rates of media stream and scaling relaxation in this paper are shown in Fig. 4, Fig. 5 and Fig. 6 respectively. The fuzzy mapping function for media buffer state is as in Eq. (4):

$$\mu_{S_1}(x) = S_1(x, l_1, l_4) = \begin{cases} 0 & x < l_1 \\ \frac{x - l_1}{l_4 - l_1} & l_1 \le x \le l_4, \\ 1 & x > l_4 \end{cases} \quad (4)$$

where, S_1 is the fuzzy mapping function that represents the media buffer state and x is the current length of media buffer.

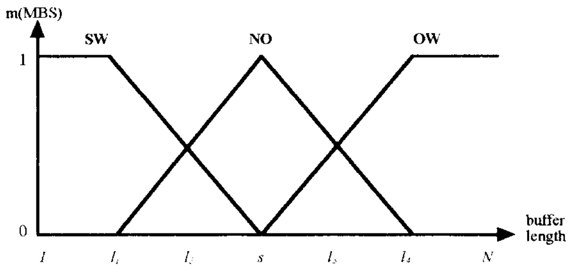

Fig. 4. Membership function for control of media buffer state.

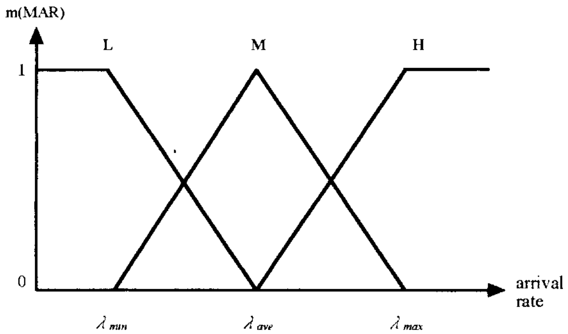

Fig. 5. Membership function for the control of media stream arrival rate.

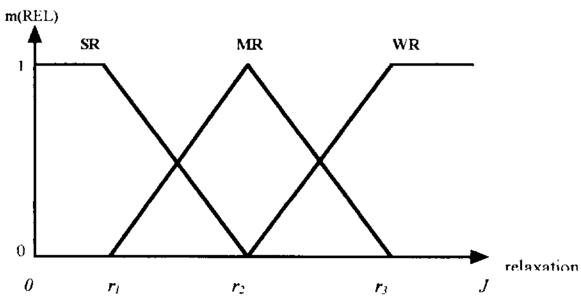

Fig. 6. Membership function for the scaling relaxation control.

The fuzzy mapping function for the arrival rates of media stream is as in Eq. (5):

$$\mu_{S_2}(y) = S_2(y, \lambda_{min}, \lambda_{max}) = \begin{cases} 0 & y < \lambda_{min} \\ \frac{y - \lambda_{min}}{\lambda_{max} - \lambda_{min}} & \lambda_{min} \le y \le \lambda_{max}, \\ 1 & y > \lambda_{max} \end{cases} \quad (5)$$

where, S_2 is a fuzzy mapping function that represents the arrival rates of media stream and y represents the current arrival rates of media stream.

The normalization of fuzzy set for media buffer state and arrival rates of media stream are shown in table 2 and table 3 respectively.

Table 2. The normalization of fuzzy set for media buffer state.

membership function	normalized full set	normalized regions	basic fuzzy set
$\mu_{S_1}(x)$	[0.00 ~ 1.00]	[0.00 ~ 0.24]	SW
		[0.25 ~ 0.75]	NO
		[0.76 ~ 1.00]	OW

Table 3. The normalization of fuzzy set for the arrival rates of media stream.

membership function	normalized full set	normalized regions	basic fuzzy set
$\mu_{S_2}(y)$	[0.00 ~ 1.00]	[0.00 ~ 0.24]	L
		[0.25 ~ 0.75]	M
		[0.76 ~ 1.00]	H

Defuzzification of fuzzy set for scaling relaxation is as follows:

$$\begin{cases} defuzzifier(WR) = 0.9 \\ defuzzifier(MR) = 0.8 \\ defuzzifier(SR) = 0.7 \end{cases}$$

5. Intermedia synchronization algorithm

In this section, intermedia synchronization algorithm, which is executed under the distributed multimedia environment as shown in Fig. 1, is designed, where, each media data is transmitted from media server to client. Since human recognition of audio error is more sensitive than that of video error, audio packets have higher priority than video packets. Therefore, let audio be the master media and video be the slave media respectively.

In this paper, by using fuzzy logic based on the slave media's buffer state compared with master media's and intermedia synchronous time, intermedia synchronization algorithm that plays back the slave media with variable service rates is designed. The intermedia synchronous time is calculated from Rangan[1]'s approach. If τ_m and τ_s are the arrival times of media unit n_m and n_s at client from media server s_m and s_s respectively, the earliest and the latest generation times of n_m and n_s represented as $g^e{}_m(n_m)$, $g^l{}_m(n_m)$, $g^e{}_s(n_s)$, and $g^l{}_s(n_s)$ are evaluated as in Eq. (6):

$$g^e_m(n_m) = \tau_m - \Delta_{\max},\quad g^l_m(n_m) = \tau_m - \Delta_{\min}$$
$$g^e_s(n_s) = \tau_s - \Delta_{\max},\quad g^l_s(n_s) = \tau_s - \Delta_{\min} \qquad (6)$$

where, g is the generation time of media unit, n_m is the master media unit, and n_s is the slave media unit. The generation time of media unit n_m and n_s, $g_m(n_m)$ and $g_s(n_s)$, are included in generation region $[g^e{}_m(n_m), g^l{}_m(n_m)]$, $[g^e{}_s(n_s), g^l{}_s(n_s)]$ respectively. Since the exact generation time of media unit is not known to client, the synchronous time can be calculated based on the generation interval of media units transmitted from master and slave servers as in Eq. (7).

$$\delta g_{\max}(n_m - n_s) = max(g^l_m(n_m) - g^e_s(n_s),\ g^l_s(n_s) - g^e_m(n_m)) \qquad (7)$$

The membership function for the degree of synchronization among media is shown in Fig. 7, and the fuzzy mapping function for the degree of synchronization among media is as in Eq. (8):

$$\mu_S(\delta g_{\max}(n_m - n_s)) = S(\delta g_{\max}(n_m - n_s), \varepsilon)$$
$$= \begin{cases} 0 & \delta g_{\max}(n_m - n_s) \le -\varepsilon \\ \dfrac{\delta g_{\max}(n_m - n_s) + \varepsilon}{\varepsilon} & -\varepsilon < \delta g_{\max}(n_m - n_s) < 0 \\ \dfrac{\varepsilon - \delta g_{\max}(n_m - n_s)}{\varepsilon} & 0 \le \delta g_{\max}(n_m - n_s) < \varepsilon \\ 0 & \delta g_{\max}(n_m - n_s) \ge \varepsilon \end{cases} \qquad (8)$$

where, S is the fuzzy mapping function that represents the degree of synchronization and ε is the maximum tolerable asynchronous time.

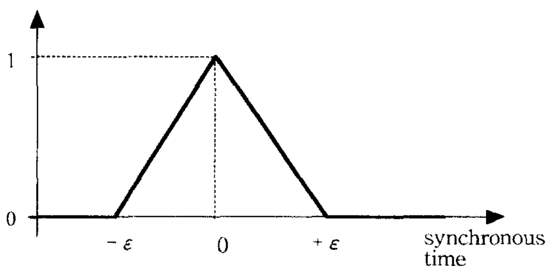

Fig. 7. The membership function for the degree of synchronization among media

The normalization of fuzzy set for the degree of synchronization among media is shown in table 4. Rules for variable service rate control, which is based on the intermedia synchronous time and the state of slave media for intermedia synchronization, is shown in table 5, where, if $\delta g_{max}(n_m - n_s) > 0$ then the state of slave media is SLOW, and FAST otherwise.

Table 4. The normalization of fuzzy set for the synchronization among media

membership function	normalized full set	normalized regions	basic fuzzy set
$\mu_S(\delta g_{\max}(n_m - n_s))$	[0.00 ~ 1.00]	[0.00 ~ 0.24]	WSYN
		[0.25 ~ 0.49]	MSYN
		[0.50 ~ 1.00]	SSYN

Table 5. Rules for variable service rate control

SMS \ SYN	SSYN	MSYN	WSYN
SLOW	NSR	LHSR	VHSR
FAST	NSR	LLSR	VLSR

the number of rules = 6

(Input Variable)
SYN - Synchronization
SMS - Slave Media State

(Output Variable)
VSR - Variable Service Rate
(Linguistic Variable)
SLOW - A slave media slower than the master media
FAST - A slave media faster than the master media
SSYN - Strong Synchronization
MSYN - Medium Synchronization
WSYN - Weak Synchronization
NSR - Normal Service Rate
LLSR - Little Low Service Rate
VLSR - Very Low Service Rate
LHSR - Little High Service Rate
VHSR - Very High Service Rate

6. Simulation and Evaluation

In this section, the performance of proposed distributed multimedia synchronization algorithm is evaluated through simulation. For intramedia synchronization algorithm, the buffer length of each media is examined and the buffer smoothness, synchronization accuracy, the number of scaling messages transmitted from clients to media server, and the frequency of occurrences of scaling relaxation are evaluated.

The results of the simulation for intramedia synchronization algorithm are as follows: The performances of the intramedia synchronization algorithm are depicted in Fig. 8 and Fig. 9 respectively. Fig. 8 shows the variation of buffer length on video media with time. Neither starvation nor overrun is not occur during simulation time in Fuzzy_Scaling which is proposed in this paper. However, starvation occurs between 400 and 700 seconds in No_Scaling method which does not use scaling technique. Thus, the Fuzzy_Scaling method seems to stabilize the state of media buffers. Fig. 9 shows the variation of the buffer length on audio media with time and starvation occurs between 350 and 550 seconds in No_Scaling method, but the buffer state is stabilized in Fuzzy_Scaling method.

Fig. 10 shows the variation of the number of scaling messages which a client transmits to media servers in simulation time.

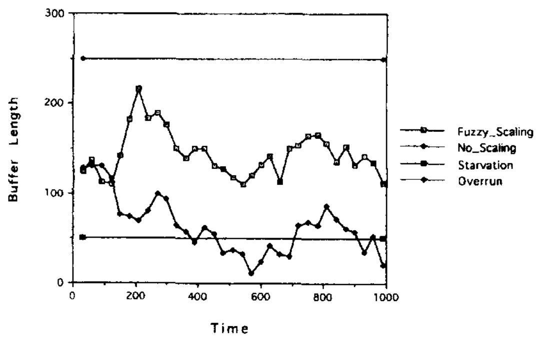

Fig 8. Variation of the buffer length on video media.

During simulation time of 1000 seconds, total 71 scaling messages were transmitted (that is, 7 scaling messages on video media and 64 scaling messages on audio media). The overhead of scaling message transmission is trivial, and scaling messages were transmitted near the starvation or overrun threshold of the buffer length in Fuzzy_

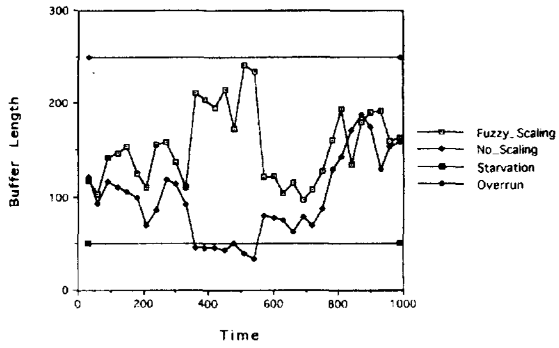

Fig. 9. Variation of the buffer length on audio media.

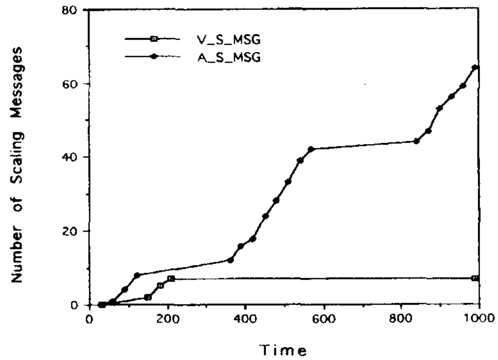

Fig. 10. The number of scaling messages transmitted by the client.

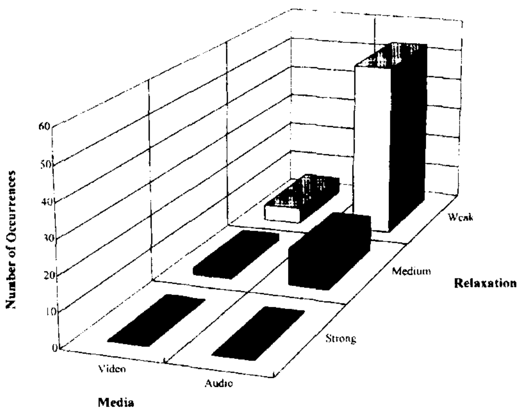

Fig. 11. The frequency of occurrences of scaling relaxation for each media.

Scaling method as shown in Fig. 8 and Fig. 9 respectively

Fig. 11 shows the frequency of occurrences of scaling relaxation for each media, where, for video media, strong relaxation (0.7) is not occurred, instead 2 times of medium relaxation (0.8) and 5 times of weak relaxation (0.9) are occurred. For audio media, strong relaxation is not occurred, instead 11 times of medium relaxation and 53 times of weak relaxation are occurred. Accordingly, we know that QoS degradation due to media scaling is minimal.

The performance of proposed intermedia synchronization algorithm is evaluated through simulation with parameters listed in table 6.

Table 6. Simulation parameters

parameter	value
max. asynchronous time(ε)	150 ms
synchronous jitter	5 ms
NSR	30 frames/sec
LLSR	28 frames/sec
VLSR	26 frames/sec
LHSR	32 frames/sec
VHSR	34 frames/sec

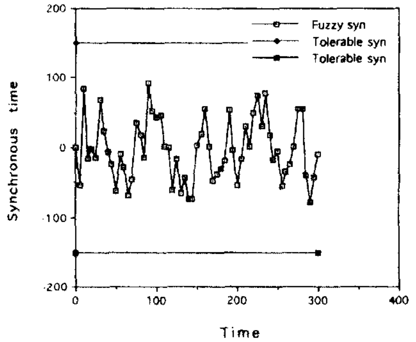

Fig. 12 The synchronous times among media

Fig. 12. shows the synchronous time between master and slave media with time. Since the intermedia synchronous time is less than the maximum tolerable asynchronous time between audio and video, intermedia synchronization algorithm works quite well among media.

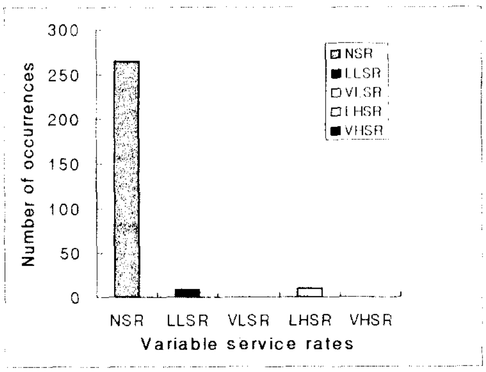

Fig. 13 The frequency of occurrences of variable service rates

Fig. 13. shows the frequency of occurrences of variable service rates. As shown in fig. 13, 88% of the total processed video frames are processed with normal service rate(NSR), 2.66% are with little low service rate(LLSR), and 3.33% are with little high service rate(LHSR). So, no video frames are processed with either very low service rate(VLSR) or very high service rate(VHSR). Therefore, QoS degradation due to the variable service rate is almost none since the variable service rate is within the range of 55% of normal service rate in video media.

6. Conclusion

In this paper, a distributed multimedia synchronization algorithm based on the fuzzy logic that supports both intramedia synchronization and intermedia synchronization is designed and the performance is evaluated through simulation.

The intramedia synchronization is achieved by using media scaling techniques on the basis of fuzzy logic for the clients' media buffer state and arrival rates of media stream. The intermedia synchronization is achieved by using variable service rates based on the fuzzy logic for the buffer state of slave media at receiver site and intermedia synchronous time.

The proposed algorithms are evaluated with respect to buffer smoothness, synchronization accuracy, the number of transmitted scaling messages, the frequency of occurrences of scaling relaxation, and the frequency of occurrences of variable service rates through simulation. Simulation results shows that the proposed algorithms are perform quite well with regards to both intramedia and intermedia synchronization. The overhead due to scaling messages scarcely exists and the QoS degradation due to both scaling messages and variable service rates is trivial.

A realization of the distributed multimedia synchronization using media scalings and variable service rates based on fuzzy logic and effective interaction between intramedia and intermedia synchronization are our future works

References

1. P. V. Rangan, S. Ramanathan, H. M. Vin, and T. Kaepper, "Techniques for Multimedia Synchronization in Network File Systems", Computer Comm., Vol. 16, No. 3, pp. 168~176, March 1993.
2. J. Sandvoss, J. Winckler and H. Wittig, "Network Layer Scaling Congestion Control in Multimedia Communication with Heterogeneous Networks", IBM European Networking Center Technical Report 43.9401, 1994.
3. L. Delgrossi, C. Halstrick, D. Hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss and C. Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", Proceedings ACM Multimedia 93, pp. 99~104, Aug. 1993.
4. D. P. Anderson and G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism", IEEE Computer, Vol. 32, No. 10, pp. 51~57, Oct. 1991.
5. T. D. C. Little, F. Kao, "An Intermedia Skew Control System for Multimedia Data Presentation", Proc. of the 3rd Int. Workshop on Network and Operating System Support for Digital Audio and Video, 1992.