Multimedia Conferencing System with Intramedia and Intermedia Synchronization Support

Sang-Shin Yoo and Duck-Jin Kim

Abstract

In this paper, we describe the design, implementation and evaluation for a multimedia conferencing system with intramedia and intermedia synchronization support between audio and video. The synchronization mechanism proposed here is capable of dynamically adapting to various network conditions thus providing an optimized QoS. In realizing the system based on this mechanism, NeVoT on Mbone is used for audio and VIC for video. Furthermore a synchronization controller is designed and realized with a unique process in supporting intermedia synchronization. Each media agents handling its media stream are modified with intramedia synchronization function. And a communicative function between media agents and synchronization controller is added as well for intermedia synchronization function. Each media agents function reports its buffering status to the synchronization control process which in turn send out optimized buffering delay value thus supporting intermedia synchronization. The realized system is configured and tested on Ethernet and ATM network where performance measurements were performed and its effective synchronization support has been assured.

I. Introduction

Recent advances in high-speed networks and multimedia technologies have provided environments for the rapid development in wide spectrum of distributed multimedia applications which generate, integrate, process, store, and distribute multimedia data[1].

Unlike the existing text-based applications, multimedia applications have quite different approach in relating variety of media stream presentation to the real time-flow. That is, each data units in a multimedia object has tightly or loosely coupled timing-relationships between them. For example, video stream is made up with lots of frames where each frame occurs with specific timing restrictions as well as a fixed presentation interval. Therefore, the original presentation interval of each frame must be preserved in a similar fashion during playback sequences at the receiver. These timing relationships exist not only in a single medium but in a group of related media. In multimedia conferencing applications, timing relationships appear especially between audio and video data. As a typical example, the motion of a persons lip should be in synchronized manner with the voice which we call it lip synchronization[2, 3]. In case where various media types are integrated into a single structure and it can be transmitted as a single unit, then the timing relationships between media may not be an influential factor at all. However in most distributed multimedia applications, a parallel transmission through multiple channels is desired for decreasing transmission delay as well as natural and effective use of its resources if the related media is to be reconstructed in the original form at the receivers side[4]. In such case, the multimedia data stream from the sender may suffer from problems such as system clock mismatch between sender and receiver, random queueing delay in high-speed networks, packet error, packet loss, and routing delay due to different routing paths in each medium which open a way for lost timing relationships at the receiver to budge in and that is why a synchronization function must be emphasized for high- quality smooth presentation at the receiving side[1-10].

To effectively support a QoS-considered multimedia applications, a synchronization mechanism capable of handling timing relationships between various media streams transmitted through the network with a specific or limited timing constraints, is therefore required. Such method can restore distorted timing relationships of media streams even before the presentation at the receiver by some processing mechanism reducing jitters and skews due to queueing delay and clock mismatch, etc.

The purpose of this paper is to propose a mechanism which supports intramedia synchronization which assures continuous playout and end-to-end timing restrictions within single medium and intermedia synchronization, responsible for maintaining timing relationships between several media. In particular, proposed mechanism is to be applied to various applications including stored and

Manuscript received February 12, 1997; accepted May 23, 1997.
S. S. Yoo and D. J. Kim are with Dept. of Electronics Eng., Korea University.

live media streams. In accordance with this purpose, we adopted a new approach that enables dynamic configuration of existing tools for the desired applications[11]. Therefore as a specific example, the proposed mechanism is embedded into a multimedia conferencing system with high-quality synchronization requirements which is applied to Internet or ATM networks thus acquiring its performance measurements and analysis.

The paper is organized as follows. Section 2 presents a mechanism in providing intramedia and intermedia synchronization. Section 3 technically describes the approach methods for realizing multimedia conferencing system and then describes the overall system architecture and the multimedia synchronization controller and of its architecture. Section 4 gives measurements and analysis of the realized mechanism and finally Section 5 sums up with the conclusions.

Ⅱ. Intramedia and Intermedia Synchronization Mechanism

Fig. 1. represents a model of multimedia data. Each media stream is divided into constant length of synchronization intervals, and this interval is divided even further into Synchronization Block Units(SBU). We may think of it as a frame of a video media stream. All SBUs included in one of the synchronization interval are called MSBU(Macro SBU). An MSBU may be considered as one or two GOPs(Group of Picture) in MPEG video coding. Thus, each media stream can be represented as the size of synchronization interval, the number of SBUs generated in a synchronization interval, end-to-end delay bound, and etc. Here we define synchronization interval as ψ, and maximum end-to-end delay as d_e^{max}.

When we consider a synchronization mechanism for interactive multimedia services, it should satisfy two restrictions, continuity and real-time(or end-to-end delay) characteristics. When data stream is delivered, the playout time(t_0') at the receiver should not be later than d_e^{max} from the start time(t_0) of the data trans mission. That is, d_e^{max} is the maximum allowable end-to-end delay permitted by the corresponding applications. This can be represented by equation (1).

$$P_t^i - S_t^i = t_0' - t_0 \leq D_{app}^{max} \quad (1)$$

where, S_t^i and P_t^i represent the ith departure time of MSBU and the playout time of receiver side, respectively.

The synchronization control in this paper is made up with MSBU as a unit thus enables us to reduce the induced overhead due to frequent synchronization control and it can be applied even in a distributed environment.

Next we propose a combined structure of both feedback and feedforward control as depicted in Fig. 2. so that it can dynamically be applied to various applications which are realized through dynamic configuration of existing media agents[3, 11, 15]. The purpose of such a structure is to dynamically adapt to various environments where the application requirements as well as network configuration, connection types, and the related systems performance are considered[16-18].

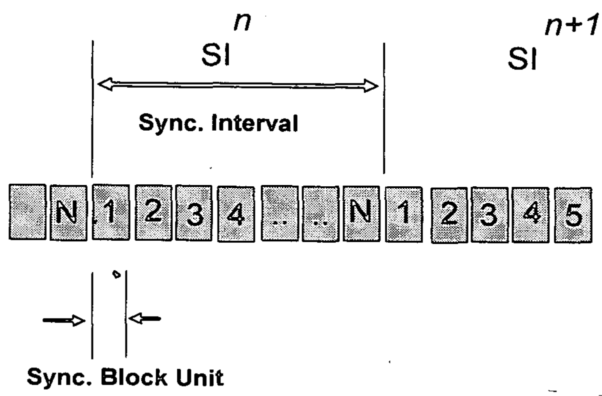


Fig. 1. Time-representation model of a media stream.

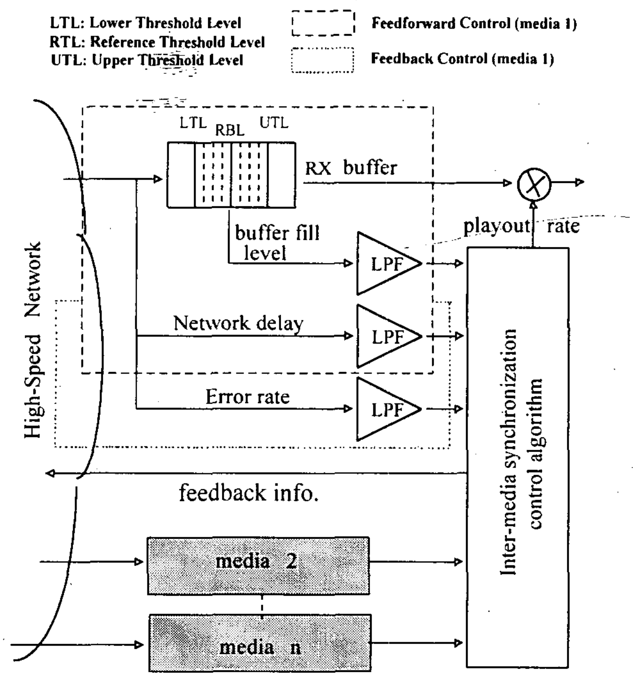


Fig. 2. Overall architecture of synchronization controller.

The feedback control has already been realized in VIC therefore an additional realization is not necessary here[18, 19]. Therefore the focus of this research emphasizes feedforward part, that is synchronization control mechanism at the receiving side.

The feedback mechanism in VIC is as follows. When an application has multicast type connections, then every receiving sides categorize the network status as unloaded, loaded, or congested. And receivers feedback this information to the sender where the congested status observed by the receiver gets counted by the sender. Furthermore the transmission rate is adjusted according to the congestion rate of current receiver status[18].

The feedforward mechanism for intramedia synchronization support is described next. We believe that reduced variability in the frame rate produces higher quality video playout compared to minimizing number of dropped frames[18]. Users are more sensitive to random frame drops than to the regular drops. Consequently, the adaptive algorithm attempts to reduce the variation of fps(frames per second) played. That is, adaptive synchronization using feedforward control provides the optimal synchronization service by dynamic control of buffering level and playout rate that was determined through a multimedia call establishment. However in high-speed packet networks, the end-to-end delay may be fluctuated, and may incur severe packet loss. So initial buffering (d_e^{init}) is intentionally added to make a fixed delay offset to satisfy equation (1). Theoretically, equation (1) is satisfied if delay offset greater than the maximum of the actual delay time(d_e^{real}) between sender and receiver is set, and this delay offset is less than the boundary of an end-to-end delay allowance. Thus, ideal buffering requirement satisfies the following.

$$D_e^{real} \le D_e^{init} = t_0' - t_0 \le D_{app}^{\max} \qquad (2)$$

Setting initial delay offset(d_e^{init}) of multimedia data stream can be achieved through a relative method which represents the difference between arrival time of the first MSBU and playout time of the first MSBU.

$$d_e^{init} = d_T + d_B^{init}$$

Where, d_B^{init} represent initial buffer delay of the first MSBU and d_T represents the network transmission delay.

If an initial buffering delay offset(d_B^{init}) is set, then the playout time of the following MSBUs is automatically determined. And the continuity(intramedia synchronization) and the real-time characteristics for a synchronization can be provided by adjusting buffer delay of the received MSBUs. That is, through maintaining reference buffer level, intramedia synchronization is achieved. Thus it is necessary to adjust the buffering delay properly according to the buffer status. Here this adjustable delay is represented by Δ_{adj} and current network status and clock mismatch are determined by monitoring the buffering delay of the latest N MSBUs.

There are several methods to get the adjustable delay Δ_{adj}. In this paper, simple method called LPF(Low Pass Filter) method is used. As shown in Fig. 2, LPF is used to eliminate instantaneous noise, which can be implemented in various ways[20].

Adjustable delay Δ_{adj} is absorbed by proper adjustment policy of the receiver's buffering condition. And most multimedia data stream permits jitters in some extent to the size of a processing interval. Therefore, the method which adjusts SBU playout rate into the synchronization interval is used. Playout rate is the rate in which the data is played out per unit time.

In this paper, in the case of normal network state, it is assumed that the network delay is guaranteed to be less than several tens of percents. Therefore both delay fluctuation and clock mismatch can be absorbed by adjusting the playout rate through SBU interval modification. In the case of congestion state, SBUs drop or duplication is inevitable. For a criterion of selecting two adjustment policies, the proposed mechanism divides total buffering space into 3 parts: lower, normal, upper areas. In case of normal area, buffering level is adjusted by playout rate using the interval modification. In contrast, buffering level is controlled by SBU drop and duplication in upper and lower area. The process of drop/duplication is same as that of transmission frame rate adjustment at the sender, except for using buffer-fill level as the reference. Therefore only playout rate control through SBU interval adjustment is described in this section.

Adjustment of the playout rate in the normal buffering area can be achieved within the adjustable boundary of playout rate (γ_{adj}^B). When n is the number of SBUs in sth synchronization interval and the reference playout rate is μ_{ref}, the size of maximally adjustable delay which can be absorbed by a playout rate adjustment of sth SI is represented as equation (3).

$$n \times \frac{1}{\mu_{ref}} \times \gamma_{adj}^B = \Delta_{adj}^{\max}, \quad (0 < \gamma_{adj}^B < 1) \qquad (3)$$

If the value of an adjustable delay Δ_{adj} is greater than $\Delta_{adj}^{\max}$, the adjustable delay is limited by maximum adjustment of a playout rate within the range specified by the applications. If the value of adjustable delay Δ_{adj} is less than $\Delta_{adj}^{\max}$, the playout rate to absorb the adjustable delay gets changed in stages. That is, $\Delta_{adj}^{\max}$ is divided into W stages and makes the required adjustable delay be absorbed gracefully in stages according to the status of buffering. Through this mechanism, the graceful degradation of QoS is provided, and thus obtaining the smoothness.

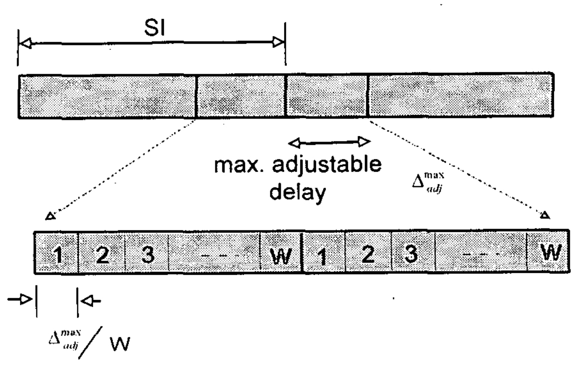


Fig. 3. The step-wise adjustment of playout rate.

The actual adjustment extent of playout rate can be found by equation (4), and lets the respective SBUs be processed with the playout rate of equation (5).

$$\left\lceil \frac{\Delta_{adj}}{\left(\frac{\Delta_{adj}^{\max}}{W}\right)} \right\rceil = k, \quad \Delta'_{adj} = \frac{k}{W} \cdot \Delta_{adj}^{\max} \tag{4}$$

$$\mu_c^{\exp} = \mu_c + \mu_{ref} \times \gamma_{adj}^{B} \times \frac{k}{W}, \quad (0 < \gamma_{adj}^{B} < 1), \tag{5}$$

where, $\mu_c^{\exp}$ is expected playout rate for the next MSBU, μ_c is current SBU playout rate, and the expression $\lceil \rceil$ represents the maximum integer less than that value enclosed.

If we represent minimum and maximum playout rate as $\mu_{\min}$ and $\mu_{\max}$, $\mu_c^{\exp}$ should satisfy the restrained condition described in equation (6) according to the requirements of an application.

$$\mu_{\min} = \mu_{ref}(1-\gamma_{adj}^{B}) \le \mu_c \le \mu_{\max} = \mu_{ref}(1+\gamma_{adj}^{B}) \tag{6}$$

Therefore, the respective playout rate μ_c^{next} for the next MSBU is determined by equation (7) and equation (8). In case where $\mu_c^{\exp} \ge \mu_{ref}$,

$$\mu_c = MIN\ (\mu_{\max},\ \mu_c^{\exp}) \tag{7}$$

and, in case where $\mu_c^{\exp} < \mu_{ref}$,

$$\mu_c = MAX\ (\mu_{\min},\ \mu_c^{\exp}) \tag{8}$$

Then a mechanism for intermedia synchronization can be described as follows. The coincidence of an adjustable delay as the dynamic condition of this synchronization mechanism is achievable by designating properly shared measurement model. The shared measurement model can be specified according to the applications. As a typical example, there are master/slave model and conservative model[17].

In a master/slave model, one of the related media stream is selected as the master stream(or reference stream), while others are considered as the slave stream. In this model, the intra-media synchronization of the slave stream can be sacrificed because the buffer status of a slave stream is not reflected to total synchronization. Therefore, this model should not be used in a specific application that doesn't allow a sacrifice of intra-media synchronization, such as the distribution applications in stereo music. In a conservative shared measurement model, the measurement of a buffer status is performed on all media streams required for an intermedia synchronization, and the respective adjustable delay is computed, where the most conservative value of these is selected as the shared adjustable delay.

We have used a modified master/slave model which gave us the priority to the intramedia synchronization control. At first, assume that the two related media i, j exists where media i is the master stream. P_k^i and P_k^j represent playout time of kth MSBU of media i, j, respectively. P_{k+1}^i and P_{k+1}^j represent playout time of $k+1$th MSBU of media i, j, respectively. Maximum acceptable skew value between two media streams are represented by $S_{k+1}^{(i,j)}$. The procedures for intermedia synchronization is described in Fig. 4. First, intramedia synchronization algorithm is independently performed on each medium. As a result of that process, a desired playout time(d_{k+1}^i) of the next MSBU is obtained. Each estimated playout time is fed into intermedia synchronization algorithm. If the difference between requested playout time is less than the maximum allowable skew, their original playout time is returned only. But if the difference is larger than the allowable skew, playout time of slave stream should be modified to be synchronized within the skew. The modification into slave stream is achieved differently according to the buffering level. At first, skew is maintained by adjusting playout time from desired time to requested time in normal buffering area case. However end-to-end delay may be partially sacrificed. In case of upper or lower buffering area, restoration of end-to-end delay is more important than the requested skew restoration. Therefore skew may be ignored.

$S_{k+1}^{(i,j)} = d_{k+1}^i - d_{k+1}^j$;

if ($|S_{k+1}^{(i,j)}| \le S_{ref}^{(i,j)}$)

  return $P_{k+1}^j = d_{k+1}^j$;

else

  if ($d_{k+1}^i > d_{k+1}^j$)

    return $P_{k+1}^j = d_{k+1}^j + (S_{k+1}^{(i,j)} - S_{ref}^{(i,j)})$;

  else

    return $P_{k+1}^j = d_{k+1}^j - (S_{k+1}^{(i,j)} - S_{ref}^{(i,j)})$;

Fig. 4. The procedure for intermedia synchronization algorithm.

III. Realization of the Conferencing Application with Synchronization Support

1. Dynamic Configuration Approach of the Conferencing Applications

Various tools for multimedia conferencing system on Internet were built by many combined efforts of researchers in recent several years. The powerful driving force behind the developed conferencing tools lies in the IP Multicast [21]. IP Multicast technology extends the traditional IP routing model by providing an efficient multiparty packet delivery. The incremental deployment of IP Multicast has been realized through the multicast backbone(Mbone), a virtual multicast network built on top of the current internet[21, 22].

Mbone has advantages of having various application tools available which support audio and video such as VIC [19], NeVoT

[23], NV [22], VAT [24], and IVS [25]. Even more flexible multimedia application architecture has recently sprang out, that have emphasized independently developed controller between the audio and video tools[11]. This approach has a general concept which can be applied not to a specific application, but to various applications made just by simply combining the given tools required by target applications. Most conventional monolithic video conferencing system suffers from a problem of having to add a separate synchronization function into each applications where our system carries an advantage of having synchronization supported controller which is applicable to all the applications therefore having higher efficiency. However this approach requires a common synchronization controller which can satisfy the synchronization requirements for various applications. Therefore we aim to propose a general synchronization mechanism for various applications including live or stored media.

This approach begins its base from the architecture proposed by Ousterhout [26] which can be summed as letting large systems be easily composed from small tools that can be integrated with simple communication primitives. In other words, several independently produced media agents can be flexibly integrated through the dynamic configuration approach thus building variety of applications executing desired tasks. The notion of realizing various applications by integrating conventional tools induces variations and reuse in media agents and its applications due to controllers, and enable us in progressively improving each composed parts.

Each media agents handle the corresponding media streams directly and send out low-speed reporting message as needed. Therefore required controller mechanism for executing a specific conferencing scenario can easily be programmed through a simple interpreter language such as Tcl/Tk [26] and Perl [27]. It has an advantage of providing simple solution in integrating audio and video tools and simple modification in media agents. Furthermore the conferencing controller can be independently designed from each media agents.

Currently, first trial for video conferencing controller which links media agents in Mbone is ISC(Integrated Session Controller) by GMD group [28]. The video conferencing in ISC is realized by connecting conventional Mbone applications such as Vic or NeVoT with the session controller. And the communication between components is done through local multicast or a separate process called PMM(Pattern Matching Multicast) for distributing local messages[11].

However, the conventional Mbone audio tools such as Vat, NeVoT, or video tools such as Vic, ivs, and nv do not have synchronization supports [2]. Furthermore the video tools such as vic, ns, ivs immediately display incoming frames as soon as they are received and decoded. Therefore a different multimedia conferencing system based on the conferencing applications in ISC is realized and evaluated in this paper where intermedia and intramedia synchronization function is applied into it.

2. The Architecture of Multimedia Synchronization Controller

Fig. 5. depicts multimedia conferencing applications which has a synchronization support. The architecture has great flexibility due to dynamic configuration of the existing media agents used in designing the system [11]. The overall conferencing applications contain media agents capable of processing a specific medium where NeVoT was used for audio support tool and VIC for video support tool.

The media agents, which process the related media streams, basically contain an intramedia synchronization function where proposed synchronization mechanism gets executed. Furthermore a communicative function exchanging messages with the synchronization controller is added as well, which enables intermedia synchronization.

Each media agent is an independent separate application and handles the corresponding medium differently from other media agents. In this paper, a multimedia synchronization controller which manages and integrates media agents according to an overall conferencing scenario is designed and realized which highly relates itself to each media agents thus executing the synchronization policy according to the proposed intermedia synchronization mechanism.

Fig. 6. shows the synchronization controlling method between the synchronization controller and the media agents [2]. The synchronization controller decides an optimized buffering delay value for the media agents according to the current buffering delay and the playout time from audio and video agents.

For example, a synchronization supporting video tool(Sync-Vic) informs the synchronization controller(MSC) of its desired buffering delay and executes the synchronization function according to the optimized buffering delay value from the synchronization controller. In other words, all of the media agents adjusts itself to the reference delay value sent from the controller thus executing its intramedia synchronization function which in turn enables an efficient adjustment of synchronization between the independent media agents.

The designed synchronization controller has three synchronization modes for controlling intermedia synchronization which is

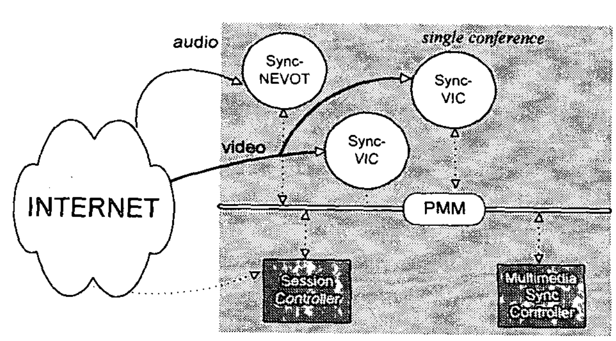


Fig. 5. The components of synchronized multimedia conferencing system.

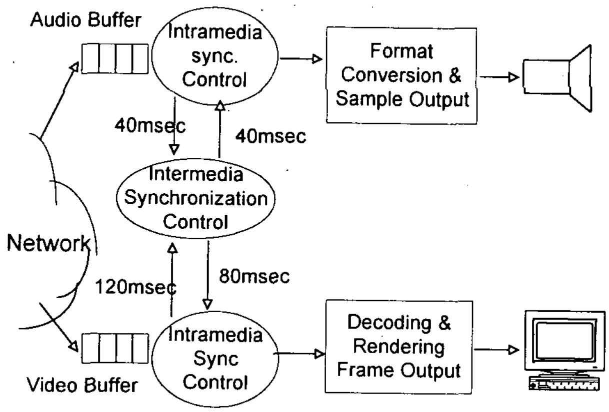


Fig. 6. Intermedia synchronization control method between audio and video.

described as follows. Mode 1 is a non synchronization mode where Mode 2 is intramedia synchronization control only mode within the media agents and non intermedia synchronization control. Mode 3 is the mode which executes both the intramedia and intermedia synchronization control. Mode 3 can have wide range of characteristics according to the given QoS parameters. Generally audio data is very sensitive to loss and delays where loss and delays in video data has somewhat little affect in the quality of service. In other words, the delays in audio is quickly sensed as bad quality audio to the listener where limited video data loss or delay does a little or no harm to the video quality. In such real-time applications, the delay or loss sensitive audio becomes the reference media in Mode 3 for intermedia synchronization in respect of QoS characteristics. But, in case where intermedia synchronization seriously harms intramedia synchronization, in other words in case where intramedia QoS is deteriorated over the specific limit, intramedia synchronization is primarily applied by the media agents. Fig. 7. describes a message exchange scenario between media agents and controller for intermedia synchronization.

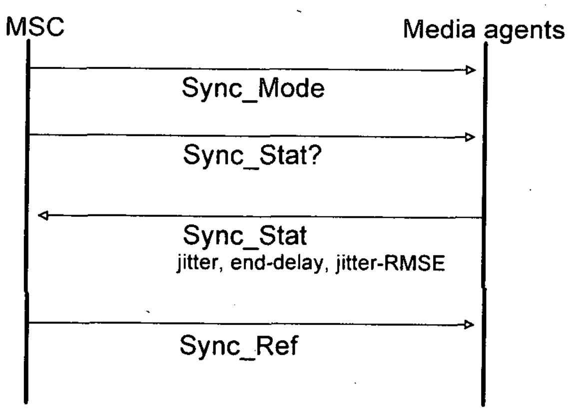


Fig. 7. Message flow between media agents and synchronization controller.

First, multimedia synchronization controller begins by sending out Sync_Mode message which contains information on synchronization mode to every media agent(NeVoT, Vic). Then the controller requests synchronization-related statistics information in every synchronization adjustment period through Sync_Stat? message from the media agents. The media agents collect the synchronization-related information at every Sync_Stat? messages then send jitter, buffering delay, and end-to-end delay information through Sync_Stat message. The intermedia synchronization controller MSC which has received synchronization-related information from each media agents must calculate the reference buffering delay of the next synchronization interval unit which is send out through Sync_Ref message.

3. Realization of the Synchronization Controller

The Mbone is a shared packet network. Routers operate on a first-in first-out basis and statistically multiplex traffic from different sources. The result of such operation on a real-time traffic induces jitter to the inter-packet timing relationships. Jitter must be removed from audio packet streams, since it renders the speech unintelligible. To relieve audio data from such jitters, an audio synchronization buffer is inserted to intentionally induce delays in audio stream. Audio stream delay is easily achieved by simply adjusting the delay time into the voice reconstruction buffer in existing NeVoT. However, video tools do not have such buffers. Therefore an additional buffering function must be added to the video tool for synchronization. Fig. 8. depicts the multimedia synchronization controller designed and realized inside the video tool. The controller is structured as a ring buffer to reduce network jitters as well as offer synchronization function.

Data streams from the sender are temporarily stored in the ring buffer and is sequentially displayed in accordance with the selected playout speed within allowed range for intramedia synchronization by the scheduler. The delay variations in networks can be significantly smoothed out through this mechanism. In other words, through the synchronization control, the playout rate

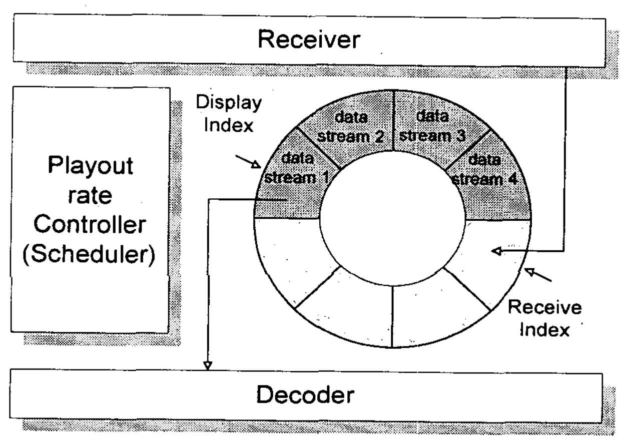


Fig. 8. The buffering structure of synchronization controller.

the playout rate controller must absorb end-to-end delay variations and clock mismatch as well as differences between various receivers due to multicasting.

Since the total delay used for a reference in the synchronization control can be represented as the sum of buffering delay and network delay, the reference value of the buffering delay at each receiver must be selected based on the estimated delay value of the current network environment. Furthermore intermedia synchronization can be achieved by combined control of the playout point in each media stream.

IV. The Performance Evaluation in Synchronized Multimedia Conferencing

Researches on synchronization is only at its birth and therefore researches for its performance evaluation metrics are made only in some areas. However, there are consistencies when it comes to jitter, delay, and skew as a significant measure in multimedia QoS[1, 2, 29-33]. The delays in transmitting video, audio, or image can be defined as the delay between signal production and signal presentaion. Generally, the end-to-end delay in networks varies randomly as time flows. The random delay characteristics in networks must be controlled so that buffering at the receiver and synchronization controller are constantly maintained within a limited range.

For delay sensitiveness and continuous flow characteristics, delays for the continuous media must especially be controlled by adjusting the playout rate at the receiver within a limited range so that QoS is well maintained. Therefore three parameters of the buffering delay, the jitter, and the skew variations are used as performance metrics for evaluating the synchronization performances in the realized multimedia conferencing application. Following equation (9), (10), (11) are the definition of these three metrics.

$$P_v=\sqrt{\frac{\sum(P_{ref}-P_c)^2}{n}} \quad (9)$$

$$B_s=\sqrt{\frac{\sum(B_{ref}-B_c)^2}{n}} \quad (10)$$

$$S_v=\sqrt{\frac{\sum(T_{master}-T_{slave})^2}{n}} \quad (11)$$

where P_{ref} and P_c represent the reference playout rate and the current playout rate, respectively. B_{ref} and B_c represent the reference buffering delay and the current buffering delay, respectively. T_{master} and T_{slave} represent the presentation times of next MSBU in master and slave media, respectively

As our testing environment for performance evaluations, an Ethernet LAN connected with FDDI as a backbone is configured as shown in Fig. 9.

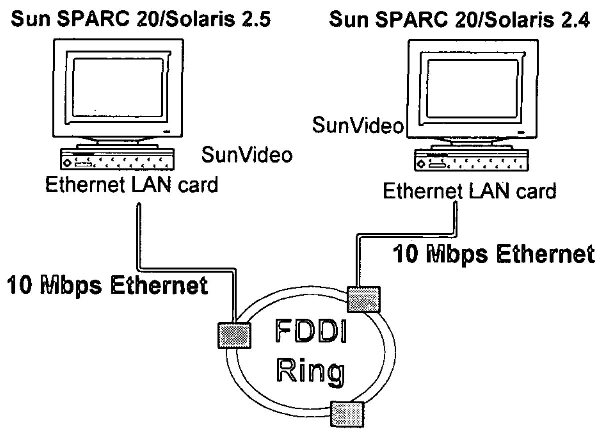


Fig. 9. The test environments for performance evaluation.

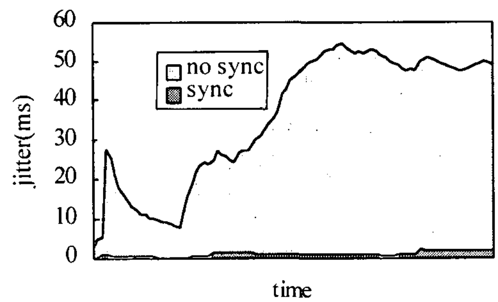


Fig. 10. Average video jitter with and without sync control.

The testing on LAN environment is configured on SunSparc 20 operated by solaris 2.4 and by Solaris 2.5 as shown in Fig. 9. For video card, SunVideo was used.

Fig. 10. shows an average jitter variations in video media with and without synchronization control. It shows jitter variations at the receiver when the frames were sent with periods of 100mS.

Without synchronization control, jitter increased from 10mS to 50mS maximum but with synchronization control, jitters were stabilized within 5mS. Intramedia synchronization works by controlling the difference between reference buffering delay and current buffering delay, disregarding other media. In addition the maximum adjustable delay allowed by the application is divided into W steps thus letting network jitters be step-wisely absorbed in this research. Fig. 11(a), (b), and (c) show video jitter variations with 1, 2, and 4 number of steps respectively. Buffering delay variation in intramedia during synchronization control is shown in Fig. 12. And Fig. 13. shows the skew variations in case of with and without intermedia synchronization control. Table 1. shows the results of the playout rate jitter, the buffering delay, and the skew variations used for evaluating the proposed synchronization mechanism.

With initial adjusting step set to 1 in intramedia synchronization mode, playout rate, buffering delay, and skew variations were in average of 28mS, 237mS, and 264mS respectively

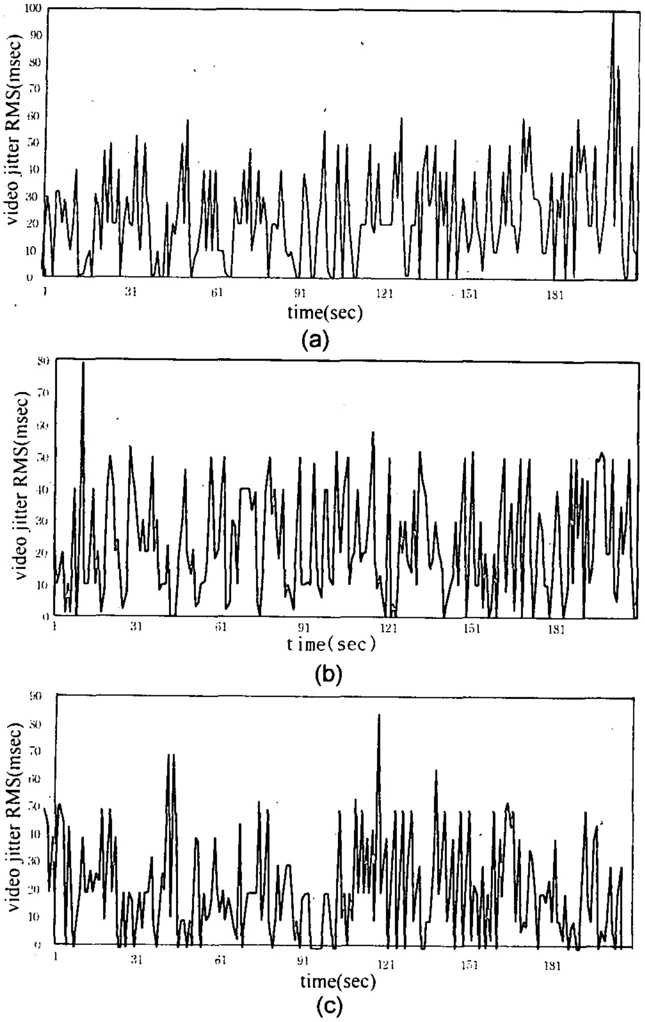


Fig. 11. Video jitter variations in the intramedia sync. Mode. (a) No. of steps W=1 case (b) W=2 case (c) W=4 case

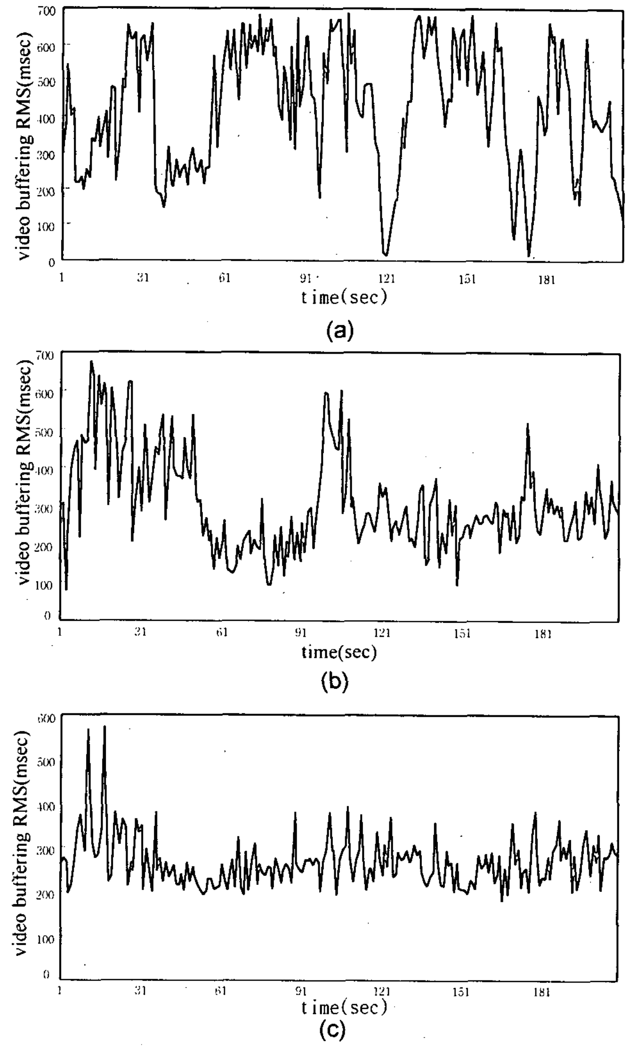


Fig. 12. Video buffering delay variations in the intramedia sync. Mode. (a) No. of steps W=1 case (b) W=2 case (c) W=4 case.

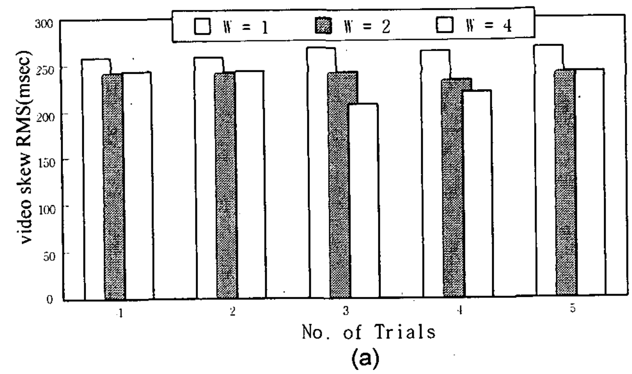
however, with adjusting step number set to 4 as proposed in ourstep-wise adjustment scheme, 25mS, 178mS, and 221 msec were achieved respectively which proves that 10-20% performance increase can be achieved.

Unlike conventional system controlling only when network buffering level reaches the upper and lower threshold value, our system presents several control points thus allowing smooth absorption of the network environment variations furthermore allowing smooth QoS degradation in worst network condition and therefore bring about improvements in system performance.

However, a problem of incurring jitter or delay variations larger than expected exist since available bandwidth is about 70-80 kbps in the networks. Furthermore we cannot even guarantee this bandwidth. Therefore it can be said that the transmission rate of video changes randomly through feedback control. In case of Vic, video with noticeable motions on a point-to-point ATM connection requires approximately 300kbps of bandwidth. Therefore the problems seem to arise from limited bandwidth, unavoidable transmission rate variation by feedback control during heavy bandwidth variation, and absorption of such variations through buffering at the receiver. So the best solution would be to limit transmission rate variations and to adjust variations through the quantization factor.

Table 1. Performance rusults of the proposed synchronization mechanism.

Metrics \ Mode	Intramedia Sync. Mode		Intermedia Sync. Mode	
No. of steps	1	4	1	4
P_v	28	25	30	32
B_s	237	178	187	178



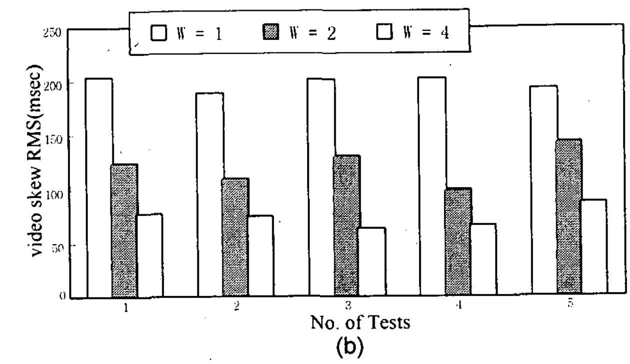


Fig. 13. Skew variations in several tests.
(a) In case of the intramedia sync. mode,
(b) In case of the intermedia sync mode

V. Conclusions

As a typical example for supplying distributed multimedia service, a multimedia conferencing application with a synchronization function maintaining original timing relationships between sender and receiver is designed and realized for this paper. The systems expandability and flexibility are assured by using dynamic configuration approach which integrates independently formed media agents on Mbone, unlike conventional applications where a single independent application controls all media. In addition, our proposal includes a mechanism for stream synchronization which is able to support a required synchronization requirements at the communication protocol level.

The synchronization performances of the realized video conferencing system was configured on Ethernet LAN environment where jitter, buffering delay, and skew variations were used as performance metrics. According to the results, jitter variations were 50 times less achieved with the synchronization function than without. Furthermore, jitters, buffering delay, and skew variations were limited to 30mS, 250mS, and 74mS, respectively which gave us comparatively better QoS delivery even with significant delays and jitters.

It is estimated that the results can be effectively applied to future application services requiring flexible QoS delivery in addition to the current multimedia services.

Acknowledgements

This research has been supported by Information Technology Assesment of Korea.

References

[1] R. Steinmetz and K. Nahrstedt, *Multimedia, Computing, Communications and Applications*, Prentice Hall, 1995.

[2] I. Kouvelas, V. Hardman, and A. Watson, "Lip Synchronization for use over the Internet: Analysis and Implementation," *Globecom96*, Dec. 1996.

[3] J. Y. Hui and E. Karasan, "Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrivals," *IEEE JSAC*, Vol. 14, No. 1, pp. 226-237, Jan. 1996.

[4] D. Shepherd and M. Salmony, "Extending OSI to Support Synchronization Required by Multi-media Applications," *Computer Communications*, Vol. 13, No. 7, pp. 399-406, Sep. 1990.

[5] W. A. Montgomery, "Techniques for Packet Voice Synchronization," *IEEE Journal on selected Areas in Communication*, Vol. SAC-1, No. 6, pp. 1022-1027, Dec. 1983.

[6] R. Steinmetz, "Multimedia Synchronization Techniques : Experiences Based on Different System Structures," *Proc. of IEEE Multimedia '92*, pp. 306-314, 1992.

[7] T. D. C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," *IEEE Journal on Selected Areas in Communication*, Vol. 9. No. 9, pp. 1368-1382, Dec. 1991.

[8] Ramanathan and P. V. Rangan, "Feedback Techniques for Intra-Media continuity and Inter-Media Synchronization in Distributed Multimedia Systems", *The Computer Journal*, Vol. 36, No. 1, pp. 19-31, Mar. 1993.

[9] R. Steinmetz, "Synchronization properties in multimedia systems," *IEEE Journal on Select Areas in Communication*, Vol. 8, No. 3, pp. 401-412, Apr. 1990.

[10] C. C. Yang and J. H. Huang, "A real-time synchronization model and transport protocol for multimedia applications," *INFOCOM'94, 1994.*

[11] H. Schulzrinne, "Dynamic Configuration of Conferencing Applications Using Pattern-Matching Multicast," *NOSSDAV '95*, Apr. 1995.

[12] L. Li, A. Karmouch, and N. D. Georganas, "Synchronization in Real Time Multimedia Data Delivery," *Proc. of ICC'92*, pp. 321.1, 1992.

[13] L. Li, A. Karmouch, and N. D. Georganas, "Multimedia Segment Delivery Scheme and Its Performance for Real-Time Synchronization Control," *Proc. of ICC'94*, pp. 1734-1739, 1994.

[14] S. Ramanathan and P. V. Rangan, "Feedback Techniques for

Intra-Media continuity and Inter-Media Synchronization in Distributed Multi-media Systems," *The Computer Journal*, Vol. 36, No. 1, pp. 19-31, Mar. 1993.

[15] J. Escobar, D. Deutsch, and C. Patridge, "Flow Synchronization Protocol," *Proc. of Globecom'92*, pp. 1381-1387, 1992.

[16] ITU-T REC. I.374, "Framework recommendation on network capbilities to support multimedia services," 1995.

[17] K. Rothermel and T. Helbig, "An Adaptive Stream Synchronization Protocol," *Technical Report 14/1994*, Stuttgart Univ. Dec. 1994.

[18] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of Multimedia Applications based on RTP," May 1995.

[19] S. McCanne and V. Jacobson, "vic: A flexible framework for packet video," *Proceedings of ACM Multimedia '95*, Nov. 1995.

[20] H. Massalin and C. Pu, "Fine-grain adaptive scheduling using feedback," *Computing System*, pp. 139-173, 1990.

[21] S. E. Deering, "Multicast Routing in a Datagram Internetwork," PhD thesis, Stanford University, Dec. 1991.

[22] H. Schulzrinne, "Guide to NeVoT," GMD Fokus, Berlin, Germany, 3.32 edition, Sep. 1995. The software is available from ftp://gaia.cs.umass.edu/pub/hgschulz/nevot

[23] R. Fredrick, "Experiences with real-time software video compression," *Proc. 6th Int. Workshop Packet Video*, pp. F1.1-1.4, Sep. 1994.

[24] V. Jacobson and S. McCanne, *The LBL audio tool vat*, Available from ftp://ftp.ee.lbl.gov /conferencing/vat/, Jul. 1992.

[25] T. Turletti, "The INRIA videoconferencing system(ivs)," *ConneXions-The Interoperability Report*, Vol. 8, No. 10, pp. 20-24, Oct. 1994.

[26] J. K. Ousterhout, *Tcl and the Tk Toolkit. Reading*, Massachusetts: Addison-Wesley, 1994.

[27] L. Wall and R. L. Schwartz, *Programming perl.* Sebastopol, California: OReilly, 1991.

[28] H. Schulzrinne, "Integrated Session Controller 1.1," May 1996.

[29] F. Fluckiger, *Understanding Networked Multimedia*, Pretice Hall, 1995.

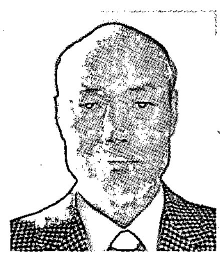
[30] L. Lamont, L. Li, and N. D. Georganas, "Synchronization of multimedia data for a multimedia news-on-demand application," *IEEE JSAC*, Jan, 1996.

[31] P. V. Rangan, S. Ramanathan, and T. Kappner, "Performance of Intermedia synchronization in Distributed and Heterogeneous Multimedia Systems," *Computer Networks and ISDN Systems*, 1993.

[32] S. S. Yoo, and D. J. Kim, "An Adaptive Synchronization Mechanism of Multimedia service in ATM Networks," KITE Journal of Electronics Engineering, Vol. 7, No. 2, pp. 7-14, Jun. 1995.

[33] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE/ACM Trans. on Networking,* Vol. 4, No. 3, pp. 340-351, Jun. 1996.

Sang-Shin Yoo was born in Buan, Korea in 1963. He received his B.S. and M.S., and Ph. D. degrees in Electronics Engineering from Korea University, Seoul, Korea, in 1986, 1988, 1997, respectively. From 1988 to 1994 he was a researcher at Samsung Electronics Communication R&D center. He is currently working as a senior researcher at Research Institute for Information and Communication. He is interested in the distributed multimedia applications, B-ISDN signalling, and ATM networks.



Duck-Jin Kim was born in Korea on December 12, 1933. He received his B.S. degree in Electronics Engineering from Seoul National University, Seoul, Korea, in 1957, his M.S. degree in Electrical Engineering from Illinois Institute of Technology(I.I.T) in 1962, and Ph.D degree in electronics Engineering from Korea University in 1972. Since 1971, he is a professor in Electronics Engineering in Korea University. He is in charge of the B-ISDN Laboratory and Pattern recognition Laboratoty. His research interests include B-ISDN signalling, wireless ATM networks, and multimedia applications.