Design and Implementation of a Lecture on Demand System
隨選教材系統之設計與實作

郭經華, 莊淇銘, 宋立群, 周子全, 王泰山
Chin-Hwa Kuo, Chi-Ming Chung, Li-Chun Sung, Tzu-Chuan Chou and Tay-Shen Wang

Multimedia Information NEtworking (MINE) Laboratory
Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan R.O.C
e-mail: chkuo@mail.tku.edu.tw

摘要

這篇論文表現了設計及實作一個應用在教學上的多媒體隨選系統。藉著這個系統，老師可以透過網路去對教材做存取、註解的動作；同時，這個系統也提供互動的方式讓學生可以同樣的透過網路去取回檔案並且播放，學生可以藉著這樣子的動作去加速學習的過程並且可以得到較好的學習效果。在設計這個系統時，採取主從架構的模式，而這個系統是由教材錄製子系統、教材儲存伺服器和網路子系統所構成。在實作這個系統時，將教材隨選伺服器架設在Windows NT的平台上而使用者可以藉由Windows NT/95的個人電腦連線上來。這個系統的原型在淡江大學多媒體網路實驗室中發展。

關鍵字： 主從架構，多媒體隨選系統，資源管理，電腦輔助教學，多媒體加註

Abstract

We design and implement a networked multimedia on demand system to perform the lecture tutoring and review process. By using this system, an instructor can access, annotate, and store lecture files over network. In the mean while, students can retrieve the corresponding files and playback in an interactive manner. Thus, students review the whole lecture in efficient way to improve their learning process. In the designing of such a system, a client-server model is chosen. The whole system consists of lecture recording subsystem, lecture storage server, and network subsystem. We implemented the server end on Windows NT platform and the client end on Windows NT/95 platform in the Multimedia Information NEtworking (MINE) Lab. at Tamkang University.

Keyword: Client-Server system, Multimedia on Demand, Resource Management, Computer Aided Education, and Multimedia Annotation

1. Introduction

The advances of computer and networking technologies make on-line access to multimedia information both possible and cost-effective. In the present paper, we design and implement a networked multimedia on demand system, we term it Lecture on Demand (LoD), to perform the lecture tutoring and review process. By using this system, an instructor can access, annotate, and store lecture files. In the mean while, students can retrieve the corresponding files and playback in an interactive manner. Thus, students review the whole lecture in efficient way to improve their learning process. The proposed system is of benefit to computer-based training, computer-based instruction, and many other cases.

Due to (1) continuous media such as voice and video that requires real-time delivering over network, (2) many users may simultaneously connect to the system, and (3) a large storage space is required, the design of a multimedia on demand system involve many challenging and complex issues. In particular, the media retrieval and storage techniques of a computer system and multimedia synchronization issue are of concerns. Many researchers have devoted to the above issues. In the study of media retrieval techniques, there are two approaches, i.e., Constant Time Length (CTL) and Constant Data Length (CDL) [1, 2, 3, 7]. Both methods retrieve data in a periodic manner. In the case of CTL, its data retrieved rate is not a constant. However, the data retrieved rate is a constant in the case of CDL. In the study of disk scheduling, the Early Deadline First (EDF), scan, round robin, and Grouped Sweeping Scheduling(GSS) are shown in the literature [4, 5, 6, 7]. In the meanwhile, different admission control and buffer management methodologies are also illustrated. In the study of interface to the client, depending on data access types, there are stream-based and file-based approaches [4, 8, 9]. In recent years, commercial multimedia on demand systems, such as Starling's StarWare and Tiger video server by Microsoft, are also available in PC environments [12].

In the present work, we design and implement a lecture on demand system. It consists of lecture storage server, lecture recording subsystem, network subsystem to perform lecture annotation, storage over network. We choose the TCP/IP protocol suite as system communication protocol. The synchronization of voice and lecture slide is achieved by using time marker method[13]. We implemented the server end on MS Windows NT platform and the client end on MS Windows NT/95 platform. The present paper is outlined as follows: In section 2, an overview of the proposed system is shown. We describe the special features of the proposed system. The design of lecture recording mechanism is presented in section 3. The storage system is presented in section 4. The design of network subsystem is given in section 5. In section 6, implementation and graphical user interface that facilitates editing and interactive processes is illustrated. Then, we conclude the present paper in section 7.

2. LoD system overview

As shown in Fig. 1, the whole lecturing and review process can be separated into three phases: (1) editing phase, (2) Lecture storage phase, and (3) replay phase. In the editing phase, an instructor retrieves lecture files from the lecture server and then annotates the corresponding files. This process happens when an instructor wants to enhance or to comment the original presentation file. As we known, information generated during this phase deserves to retain. In the Lecture storage phase, the annotated files are stored through the underlying network back to lecture storage system. In the replay phase, students interactively retrieve and replay the annotated file. On the other hand, student can also download the annotated file from server and replay it.

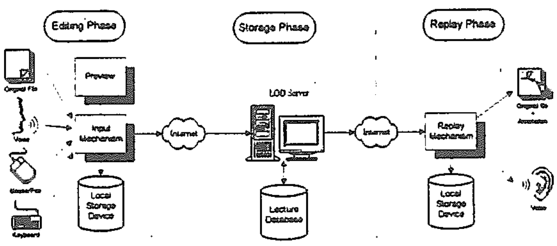


Figure 1. LoD system operation porcess

In the present paper, the proposed system considers the case of slide presentation. In such a common case, the presentation slides and the commentary part are the main data types that construct an annotated lecture. The presentation slides are the original file prepared by an instructor. This original file may be made by using MS PowerPoint or any other tools. So the original lecture file consists of text, images, or any other media. The commentary part consists of two media: the instructor commentary voice and the telepointer movement information. An instructor uses this part of data to enhance his/her presentation. As an example, the typical data consumption rate at the client end can be depicted as Fig. 2. It has the following properties.

(1) Time varying: At any time interval, not all three media (slide, telepointer, and voice) will appear.
(2) Random burst: When a slide is presented, it requires more bandwidth. Furthermore, slide sequence is not in a periodic manner.
(3) Synchronization: These media requires synchronous playback.

Thus, the above characteristics are of concern in the designing of such a system.

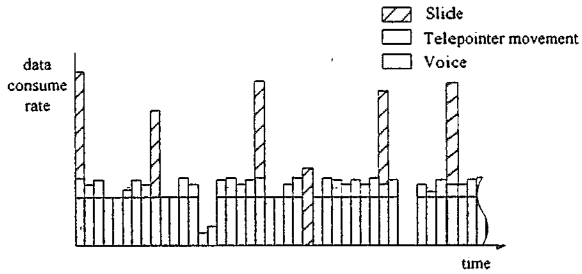


Figure 2. Lecture components playback relationship

As common approach, we utilize a client-server model to implement the Lecture on Demand (LoD) system. The network structure of a LoD system is shown as Fig. 3. It consists of the server end and the client ends. An instructor and users can access the lecture files from any client sites of the network. In this manner, the designing of a LoD system consists of the following critical subsystems:

- *Lecture recording subsystem* that manufactures and edits lecture media,
- *Lecture storage server* that supports continuous media storage and retrieval,
- *Network subsystem* that synchronously delivers media information, on time, to the client sites, and
- *Graphical user interface* that facilitates media interactive playback and annotation process is required.

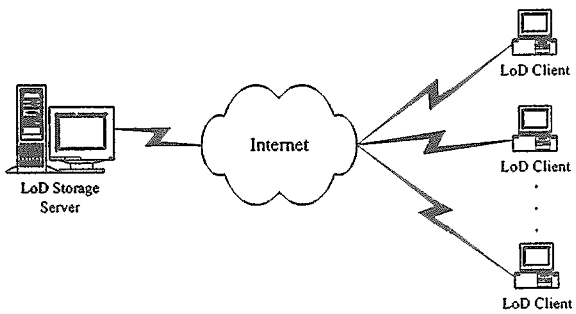


Figure 3. LoD system network structure

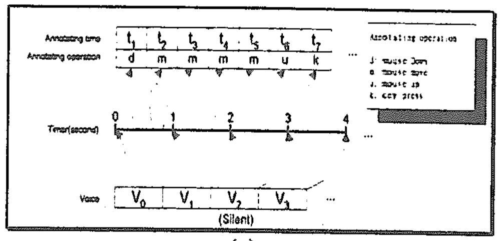
3. Lecture recording subsystem

The primary function of the lecture recording subsystem is to prepare a convenient environment for lecture annotation, synchronization, and interactive playback. In our work, a software application is designed to annotate the original lecture file. This application takes voice and telepointer movement information as inputs. Once the annotating process completed, user can select *store* function to store the annotated file back to the lecture storage server. The detail portion of this work can be found in [13].

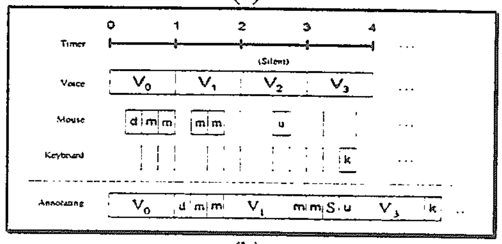
A time-based synchronization mechanism is used in our work. As shown in Fig. 4, the commentary part include voice, mouse movement, and keyboard event. These media and the original file require synchronous playback. To achieve this goal, the commentary part at each time interval, that is one second in our design, is recorded as the following format. As shown in fig. 4(a), in the first second, the system has the V_0 voice data, the mouse down event at time t_1, the mouse movement event at time t_2, and another mouse movement event at time t_3. In the 2nd second, the system has the V_1 voice data, the mouse movement event at time t_4, and another mouse movement event at time t_5. In the third second, the system has a silence interval and the mouse up event at time t_6. In the fourth second, the system has the V_3 voice data and the key press event that may indicate another slide is selected.

Then the mechanism constructs the data format in a sequence as shown in fig. 4(b). In this example, the mouse downs at the time t_1 and ups at time t_6. During this time interval, several mouse movement events may occur. Notice that the system-input mechanism detects the voice input signal at every time interval and justifies that particular time interval belongs to talk spurt or silence gap. If it is a talk spurt, no matter how long of that talk spurt, the whole time interval voice data is stored. Notice also that the storage data of the mouse event includes the time instant such that the replay time can be duplicated in the replay phase. Because the whole data in a particular second is recorded in the near location, we can replay the annotation process synchronously in the easy way during replay phase.

In order to accomplish the possibility of user playback interaction, we also designed the special indices. To facilitate the replay process, as shown in fig. 5, the annotated file contains two special indexes, i.e., the time index and the slide index. The time index is to indicate the time sequence of an annotated file. The slide index is to indicate the slide sequence. Users select this index number to change the starting slide or replay sequence. In a similar manner, users skip the undesired slides during the replay phase. However the users want to jump any portion of the courses, we can use these two indices to indicate the position of the lecture files.



(a).



(b)

Figure 4. A sample of time-based temporal synchronization

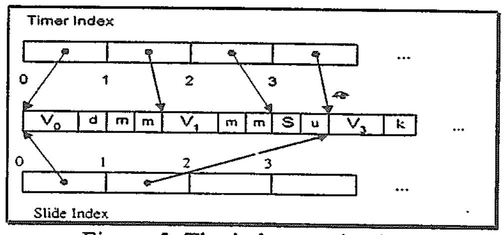


Figure 5. The index mechanism

4. Lecture storage server

The lecture server is responsible for (1)storing lecture files, (2)providing lecture contents and related information, (3)providing lecture uploads, downloads, and auto indexing services, (4)providing lecture real time as well as non-real time delivery. To perform the above tasks, in our work, the whole system is separated into three units. It contains (1)Connection management unit, (2)Lecture management unit, and (3)Resource management unit. We present the detail of each unit as follows:

(1) Connection management unit:

The connection management unit is the interface between network subsystem and resource management unit. Due to editing feature in our system, the connection management unit is designed to handle different types of service request. These service types may be real time, e.g. lecture playback, or non-real time services, e.g. lecture downloads and uploads. As shown in Fig. 6, when a request is received, the service type is further conformed. Then, the connection management unit informs its corresponding service. In each service process, it contains two types of message, i.e., control message and data message. The control message is used for request and interactive playback purposes.

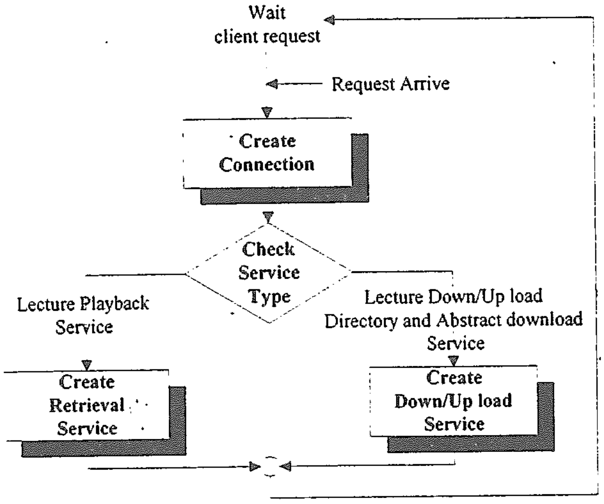


Figure 6. Connection management flowchart

The request from client end needs to be passed to resource management unit for further conformation. In this manner, we can ensure that quality of real time retrieval service. If the system does not have enough resources, the request needs to reject. On the other end, the required bandwidth is allocated. The whole process is depicted in Fig. 7.

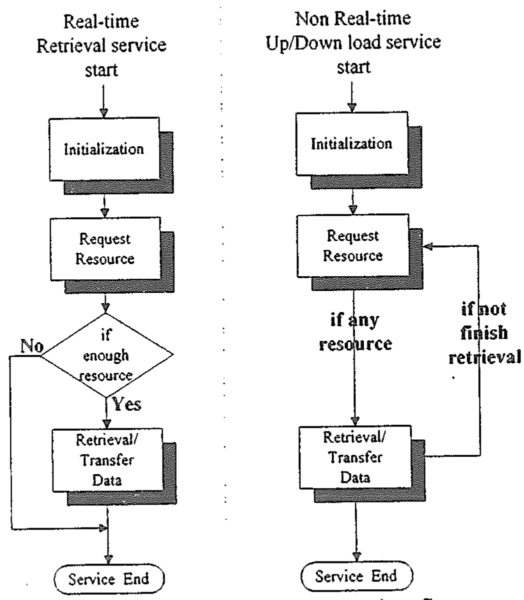


Figure 7. Service Process operation flow

(2) Lecture management unit:

This unit is in charge of lecture files and commentary files access, placement, and management. To improve lecture search efficiency, a database is used to record the corresponding lecture related information such as authors, require playback time, abstract etc. Furthermore, the SCAN strategy is used in the disk scheduling. As shown in Fig. 8, the reading requests for each service process are sorted in In-buffer, then present to Out-buffer for arranging file retrieval sequence.

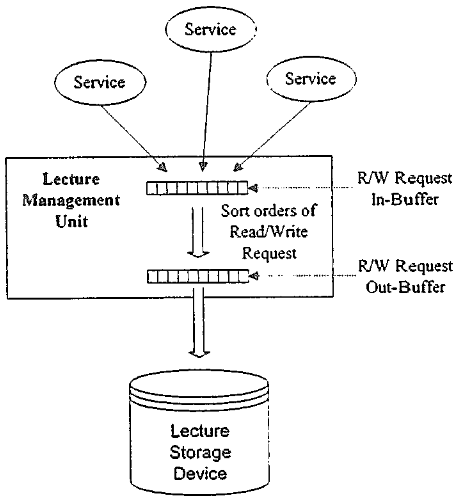


Figure 8. Lecture Management Unit structure

(3) Resource management unit:

This unit is responsible for resource management and admission control. In resource utilization, the requirements of resources are different between real-time and non-real-time service processes. Because the required resources of real-time service processes must be allocated fully before beginning service, we only enforce the admission control strategy to these service processes for ensuring the QoS.

Similar to many other on-demand systems, the unit admits a real-time service process if the system can reserve sufficient disk bandwidth and buffer space for continuous playback. In general, the simplest reservation strategy is based on the worst playback rate, but this strategy wastes many resources for VBR media. As described in section 2, the traffic characteristic of the media used in our system is a kind of VBR media. In order to simplify the calculation and improve the resource utilization, we adopt a polyline-like smoothing policy to increase the stability of the transmission rate and the peak-rate reservation strategy on smoothed traffic. Due to the limitation of space, we do not describe the smoothing algorithm in detail.

In resource managing, we adopt SCAN algorithm for disk scheduling and double buffering policy for memory allocation. Therefore, we define two rules for admission control according the characteristics of the managing strategies. These two rules are described as following:

rule 1 : for disk bandwidth (using SCAN algorithm)

$$(\sum_{i=1}^{t} r_i + r_{new}) * T_{Trans} + T_{Seek} + (t+1) * (T_{Latency} + T_{Overhead}) < T_{round}$$

rule 2 : for buffer allocation (using double buffering strategy)

$$2*(\sum_{i=1}^{t} r_i + r_{new}) < B_{total}$$

In these equations, r_i denotes the smoothed peak-rate of the ith admitted process that is executing, t denotes the total admitted processes that are executing, r_{new} denotes the smoothed peak-rate of the new service process, T_{Trans} denotes the unit transmission time, T_{seek} denotes the maximum seek time, $T_{latency}$ denotes the maximum rotational latency, $T_{Overhead}$ denotes the maximum overhead for switching processes, T_{round} denotes the period of one round (at present, we define T_{round} = 1 second), and B_{total} denotes the total memory spaces in the server. The two equations must be held after admitting any new service process.

5. Network subsystem

As we mentioned before, there are three phases in the whole system process. According to the services that provided in our system, the styles of transmissions include (1) uploading lecture data from instructor site, (2) download lecture data to student site, and (3) interactive data-stream between the server and student site. In the first two types, the LOD system simply transmits the whole file of lecture data from instructor site or to student site. But in the last type, it's a real-time transmission.

In order to realize these functions, we create two network connections between the service end and the client end: command connection for message exchanging, and data connection for lecture transmission. Furthermore, as shown in fig. 9, we use the reliable protocol, TCP, in the both connections because whole data that needed for lecture playback could not lost during transmission.

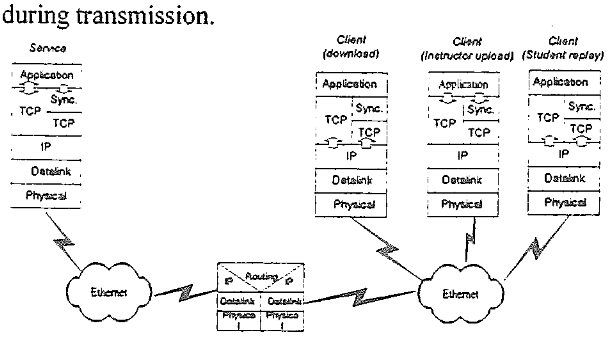


Figure 9. Network data transfer diagram

6. Graphic user interface and implementation

We implemented the proposed Lecture on Demand System that the server is on the Microsoft Windows NT platform, and instructors and student clients are on the Microsoft Windows NT and 95 platform. Voice input signal is treated by a Sound Blaster card which uses the PCM encoding scheme. To facilitate the lecture storage process, we design a graphical user interface as shown in fig. 10. By this interface, an instructor can fill the form up to indicate the attributes of the lecture and select the position of existing database to insert the lecture. The students can select to retrieve the lecture of the desired course by the user interface shown as fig. 11. After a course is selected, the students just click the start bottom to replay the annotated lecture.

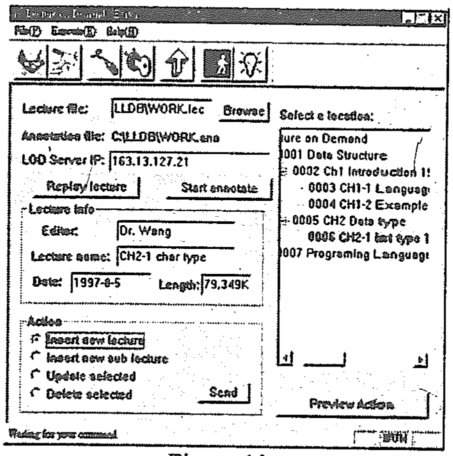


Figure 10.

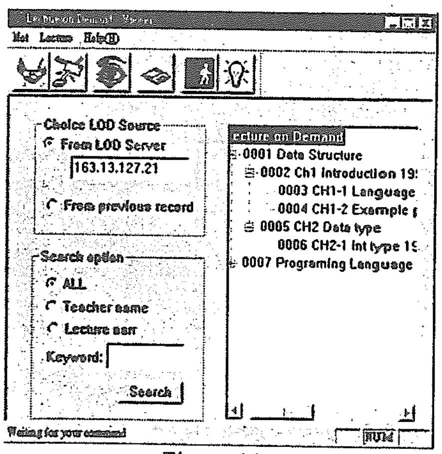


Figure 11.

Under the above circumstances, the file is replayed in the same speed as the annotating process. For example, Fig. 12 is an example of annotation. Due to the restriction of this paper presentation, the voice portion does not appear. However, from the designed data structure of the file storage one can see the voice and the pointer movement are presented simultaneously and synchronously.

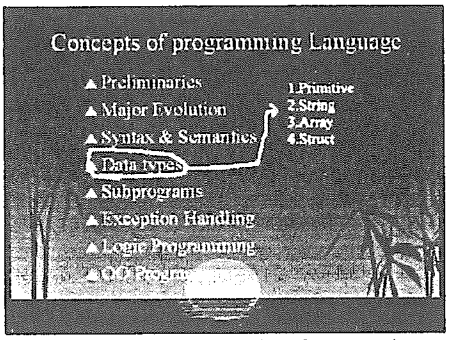


Figure 12. An example of annotation

If users want to interact with the system during replay phase, they just make a click on the screen then the designed user interface, i.e., fig. 13, will appear. By pressing the corresponding bottom, users can pause, forward, backward, and skip the slides. Furthermore, users can select a particular slide to play by using the index dialog box.

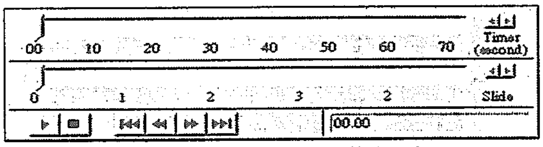


Figure 13. The index dialog box

7. Conclusions and future works

We design and implement a multimedia on demand system to facilitate lecture review process. The design and implementation of such system includes many challenging tasks. Our contributions are as follows: (1) A lecture storage system is implemented on Windows NT to store multimedia files, (2) A synchronization mechanism is implemented to ensure inter-media synchronization, and (3) A graphical user interface is implemented to facilitate the lecture annotation and interactive playback functions. Thus, an instructor's lecturing process can be recorded and then students can replay the annotated file in an interactive manner. System of this kind can be significantly benefit to our learning and training process.

To facilitate global access, a Java-based graphical interface is preferred. Furthermore, a robust resource allocation or admission control scheme is required to support multiple users access environments. We would like to pursue these issues in our future work.

References

[1] E. Biersack, F. Thiesse, and C. Bernhardt, " Constant Data Length Retrieval for Video Servers with Variable Bit Rate Streams," in proc. of IEEE Multimedia 96, pp. 151-155, 1996.

[2] S. Sahu, Zhi-Li Zhang, J. Kurose, and Don Towsley, "On the Efficient Retrieval of VBR Video in a Multimedia Server," in conf. of IEEE Multimedia Computing and System 97, pp. 46-53, June 1997.

[3] J. Gemmell and S. Christodoulakis, "Principles of delay-sensitive multimedia data storage and retrieval," ACM Trans. On Information Systems, vol. 10, no.1, pp.51-90, 1992.

[4] J. Gemmell, H. M. Vin, D. D. Kandler, P. V. Rangen, and L. A. Rowe "Multimedia storage servers: A tutorial," IEEE Computer, pp. 40-49,May 1995.

[5] P. S. Yu, M.-S. Chen, and D. Kandlur, "Grouped sweeping scheduling for DASD-based multimedia storage management," Multimedia Systems, pp. 99-109, 1993.

[6] A. Dan and D. Sitaram and P. Shahabuddin, " Scheduling Policies for an on-demand video server with batching," in proc. of ACM Multimedia 94, pp. 15-23, Oct. 1994.

[7] R. Steinmetz and K. Nahrstedt, "Multimedia: COMPUTING, COMMUNICATIONS, AND APPLICATIONS" Prentice-Hall inc., 1995.

[8] D. P. Anderson, Y. Osawa, and R. Govindan, "A file system for continuous media," ACM Trans. on Computer Systems, vol. 10, no.4, pp.311-337, Nov. 1992.

[9]T. N. Niranjan, Tzi-cker Chiueh, and G. A. Schloss, "Implementation and Evaluation of a Multimedia File System," in conf. of IEEE Multimedia Computing and System 97, pp. 269-276, June 1997.

[10]J. Gemmell, "Multimedia network file servers: Multi-channel delay sensitive data retrieval," Multimedia Systems, vol. 1, no. 6, pp.240-252 ,Apr. 1994.

[11]Y. S. Ryu ,and K. Koh, "A Dynamic Buffer Management Technique for Minimizing the Necessary Buffer Space in a Continuous Media Server," in proc. of IEEE Multimedia 96, pp. 181-185, 1996.

[12]Palmer W. Agnew ,and Anne S. Kellerman, " Distributed Multimedia : Technologies, and Opportunities in the Digital Information Industry," Addison-Wesley publishing company, 1996.

[13]Chin-Hwa Kuo, Timothy K. Shih, and Tzu-Chuan Chou, "A Synchronization Scheme for Multimedia Annotation," in proc. of the IEEE International Conference on System, Man, and Cybernetics Information, Florida, U.S.A., Oct.1997.