INTERNET Research™

ISSN 1066-2243
VOLUME 4 • NUMBER 2
SUMMER 1994

Electronic Networking Applications and Policy

Mecklermedia

A Model to Support Synchronization in Multimedia On-Demand Systems

Hossein Saiedian and Mahmoud Awad

We propose a Multimedia On-Demand (MMOD) system architecture that will support Multimedia On-Demand network services and will have the ability to solve presentation synchronization problems caused by either transmission line delay or any mismatch between the data transmission rates of the different parts of the system. Message exchange between the local host and the central host is done via a transmission request from the local host to the central site and via multimedia data block transmission from the central site to the local site. The packet structure for both kinds of exchanges is introduced. According to our synchronization technique, the local sites are given part of the synchronization duties. However, they can ask for help from the central site if their local buffers are flooded and reach some threshold value. The central location increases or decreases the data transmission rate to recover from asynchrony. The central site maintains a log of "break points" at which asynchrony occurred while recording the multimedia data, and it is the central site's responsibility to recover from such asynchrony.

On-Demand Network Services

During the last decade, there has been a sharp decrease in the cost of hardware processors and memory, combined with similar cost reductions and technological advances in the field of communications. The result has been to make computer networks attractive, viable and cost-effective tools in many environments. Industrial organizations, educational institutions, and even ordinary users have recognized the advantages of interconnected computers in providing interaction, cooperation, and sharing their sources. The availability of computer-based interaction and internetworked environments has opened the door to new and more fascinating applications. One such area is the Multimedia On-Demand System.

Multimedia communication systems have recently become one of the most talked about areas in communications. Many multimedia-related problems are still unsolved, but progress in hardware, algorithms, and standards has facilitated research in this field and made it easier to interactively access digital multimedia information systems. One of the multimedia communication services that was a dream in 1990, and is about to be a reality, is On-Demand TV/Video. Some researchers refer to it as Video (Sincoskie, 1991) simply because the services will be similar to those provided by videotape rental

Hossein Saiedian (hossein@aetna.unomaha.edu) is currently an Assistant Professor of Computer Science at the University of Nebraska at Omaha. He received his Ph.D. in Computer and Information Sciences from Kansas State University in 1989. Dr. Saiedian has published over forty technical articles in computing sciences journals and proceedings, including articles in recent issues of *Journal of Systems and Software, Journal of Software and Information Technology, Computer Security, Journal of Microcomputer Applications, Office Systems Research Journal, Journal of Computer Information Systems,* and *Journal of Computer Science Education.* His recent research contributions have been accepted for publication in *Computer Networks and ISDN Systems* and *IEEE Computer.* Dr. Saiedian is a member of the ACM and is currently Chair of the ACM SIGICE (Special Interest Group in Individual Computing Environments).

Mahmoud Awad is currently completing his Master's degree in Computer Science at the University of Nebraska at Omaha. His research areas include networking and synchronization aspects of multimedia systems and database management system. He has published part of his research results in *Journal of Information and Software Technology.*

Internet Research, Vol. 4 No. 2, Summer 1994, pp. 59–66. © 1994 Mecklermedia

stores, but with access through a computer network from a central location. Others (Ramanathan and Rangan, 1993) compare it to existing cable television and telephone network services by referring to it as television.

The simplest definition of On-Demand TV service is a service provided from a centralized location like the local central office (Sincoskie, 1991). A customer can signal the network to select a movie, and the network will start transmission within a reasonable time. The customer can stop or continue the transmission. A coarse (non-visual) rewind and fast forward can also be provided, that is, the customer could signal the network to rapidly move forward or backward some number of basic time units. All customers could choose to watch the same movie, with each customer having an independent "phase-shift" (i.e., each person is watching a different part of the movie). Alternatively, all customers could be watching different movies. Most combinations of numerous customers watching numerous movies with independent phase shifts are also possible.

The following assumptions are fundamental to such a system:

1. The storage devices, such as optical or magnetic disks, used to store such a large number of multimedia data must have an extremely high capacity, which is randomly accessible, with a short seek time.

2. The system must be permanently online.

3. The system must be powerful enough to support continuous interactivity.

4. The network involved must be a very high-speed local (or metropolitan) area network because of the real-time nature of multimedia data exchange.

Although these assumptions sound difficult to accommodate, the necessary technology will be available soon because of very rapid advances in storage and communications. Therefore, the major difficulty facing such a project is not the hardware, but the algorithms and the data delivery techniques that will provide the user with the right data at the right rate. To be more specific, a sound multimedia synchronization technique has to be available for such a system to become a reality.

Multimedia streams constituting a multimedia object must be temporally coordinated because, when they are received from the multimedia server, the data elements experience random delays during transmission and retrieval, which means that playback of these media streams can go out of synchronization. Real-time voice and video data streams are isochronous in nature; that is, they can be thought of as a stream of finite-sized samples that are generated, transmitted, and received at fixed time intervals, imposing a set of timing constraints that must never be exceeded. The delay between the generation of successive samples at the stream's source introduces a *sampling delay;* there is also a *transmission delay* between the generation of a sample and the presentation of the same sample at the stream's end (sink). In general, the transmission delay consists of the packetization delay, the network transmission delay, and the presentation delay (Nicolaou, 1990). The packetization delay is the time taken to generate a sample and transfer it to the network; the network transmission delay is the time taken to transmit the sample over the network; and the presentation delay is the time spent buffering the sample before presenting it to the user. The presentation synchronization, as a separate problem, has been discussed in a previous paper (Saiedian and Awad, 1994). Several researchers have dealt with it thoroughly (Gibbs, Dami, and Tsichritzis, 1991; Steinmetz, 1990; Vazirgeannis and Mourlas, 1993).

All these previous discussions offered their solutions to presentation synchronization in a personal computing environment. In this paper we will first propose a Multimedia On-Demand system architecture; then we will propose a synchronization mechanism that accommodates multimedia transmission over networks, and we will aim, specifically, at On-Demand service systems. The mechanism will make use of global and local buffers located at the Multimedia Server and the local hosts, respectively. The host will take part of the synchronization responsibilities by using its local buffers. In case the local host can no longer synchronize the data flow to the Multimedia Display Unit, the Multimedia Receiver at that host will send a lightweight packet asking the global Multimedia Server to help by slowing down or speeding up one of the data streams.

The rest of this paper is organized as follows: Section 2 introduces the proposed Multimedia On-Demand system architecture. Section 3 explores the problem, that is, synchronizing the data exchange between the Multimedia Receiver and the Display Unit to produce the proper multimedia presentation. The structure of the packets exchanged between the host and the server are also introduced. In Section 4 we present the Distributed Synchronization Responsibility method which will enforce the needed synchronization. We finish with a summary and concluding remarks.

The Multimedia On-Demand System Architecture

The Multimedia On-Demand (MMOD) system architecture is depicted in Figure 1. It demands a multimedia server that has a large amount of storage space (magnetic or optical disks). The server will be connected to a number of subscribers through a very high-speed Metropolitan Area Network (MAN). According to Rangan, Vin, and Ramanathan (1992), the network will be connected to media display devices such as videophones and audiophones belonging to the users. They assumed that these subscribers' mediaphones are simple media capture-and-display subsystems connected to the host computer on one side and to the network on the other side. Since the mediaphones are assumed to be simple devices capable of digitizing and transmitting information, or seen simply as receiving and playback media units, but lacking the sophistication to use elaborate time rate synchronization techniques, these subdevices will have mismatches in their rates of recording and playback.

We propose a slightly different system architecture that gives the local sites more synchronization responsibilities. According to our proposed architecture (Figure 1), the MMOD system is composed of three major constituents:

1. The central location
2. The transmission network
3. The different subscribers' local hosts.

In the following paragraphs, we provide a description of each of these constituents.

The Central Location

The central location consists primarily of a Multimedia Server, storage facilities, and a Multimedia Transmitter. The Multimedia Server is the overall transmission and synchronization manager. It provides storage space via random access disks with a very high storage capacity, for example, CD-ROM or optical technology. To give an example of the storage capacity needed (Rangan et al., 1992), assume that the Multimedia Server is to store 1,000 100–minute-long movies with a data rate of 0.5 megabytes-per-second for real-time motion movie; a disk capacity of 3 terabytes is required.

The Multimedia Transmitter takes care of data transmission and performs the actual multimedia data transmission synchronization functions indicated to it by the Multimedia Server. In addition to data transmission synchronization, the Multimedia Transmitter is to manage local message and data exchange between the Multimedia Server and itself. The Multimedia Server manages audio and video retrieval from Audio Storage and Video Storage. The Multimedia Transmitter receives such data and transfers them to the intended host, and at the same time receives information and feedback from the different hosts about synchronization needs. The transmitter also transfers them back to the Multimedia Server, which analyzes these needs and manages data retrieval and transmission to ensure a synchronized presentation in the host's display system. The global buffer is another small but important part of the Multimedia Transmitter that has a role in multimedia transmission synchronization. It is divided into two parts: the Global Audio Buffer (GAB) and the Global Video Buffer (GVB). The global buffer will be used to guarantee that when a specific media stream is faster than the other stream, it will not flood the network (and therefore the local host's machine), which would "disturb" the multimedia presentation. The appropriate buffer (GAB or GVB) will hold specific multimedia data blocks temporarily to produce some delay for that data stream, so that the other delayed stream will pick up the synchronization "rhythm." Another reason for a global buffer is to handle the differences in disk access and network transmission schedules (Rangan et al., 1992).

The Transmission Network

The MAN interconnects the Multimedia Server with the subscribers' sites. The most suitable network for such a job is a fiber-optic network that offers gigabyte-per-second bandwidths and, in the near future, fiber-optic network offering terabyte-per-second bandwidths. Asynchronous multimedia playback on the host's machine is caused by network delays experienced by media data units transmitted by the Multimedia Server via the Multimedia Transmitter to the subscriber's machine. Since such a delay is non-deterministic, we will assume it is bounded between a minimum limit and a maximum limit.

Subscribers' Local Hosts

Each local host consists of a Multimedia Receiver and a Multimedia Display Unit (MMDU). The Multimedia Receiver will receive data units from the transmission line and will send them back to the

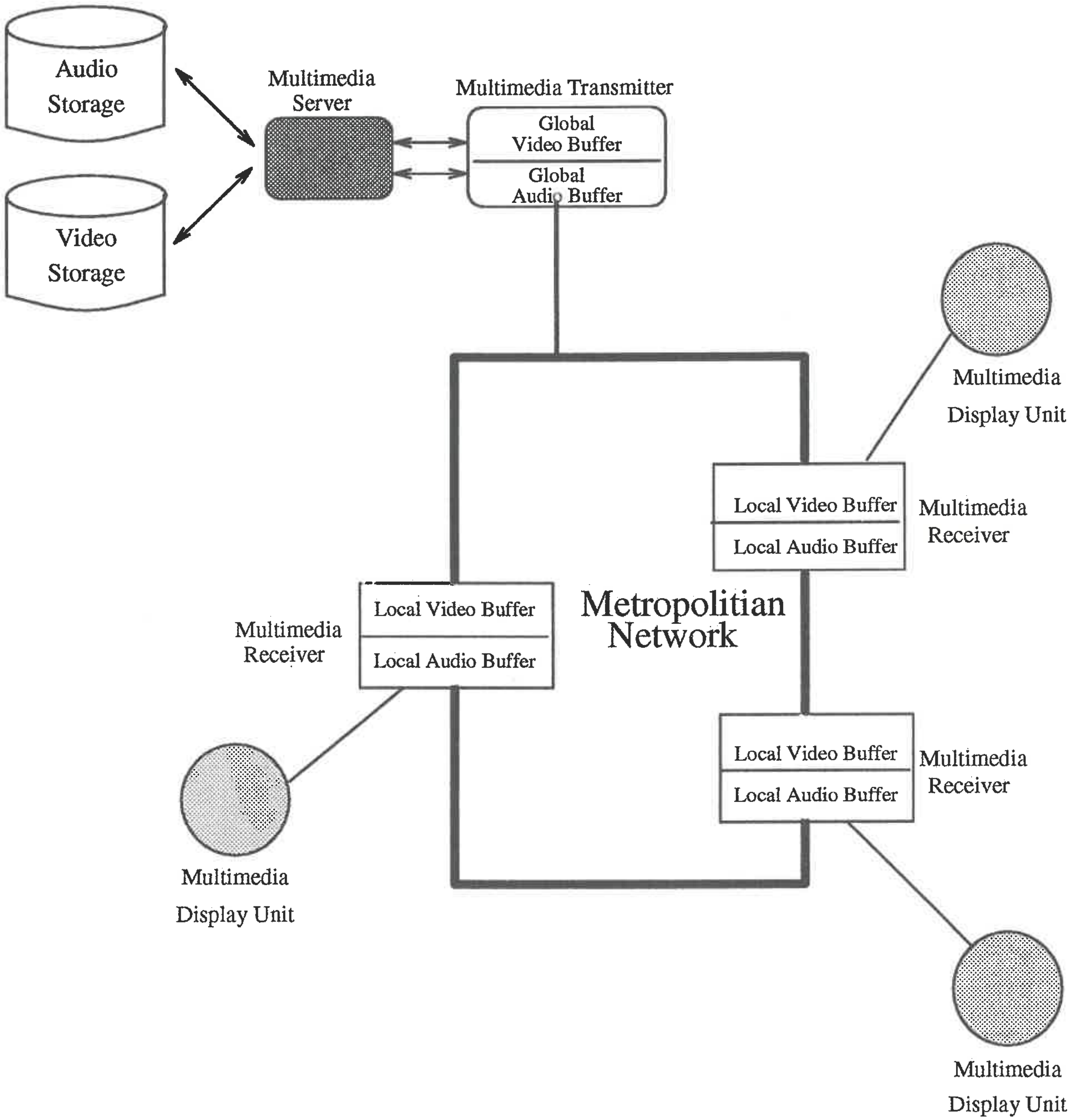

Figure 1. Multimedia On-Demand system architecture

MMDU for playback. The result of any synchronization mechanism that is used to solve network delays must appear clearly in the final playback because the original problem was that the asynchronous multimedia display to the user caused by delays experienced by the media units. Another reason for asynchronous playback is the possible mismatch between the multimedia data rate received by the Multimedia Receiver and the data rate for playback by the MMDU. Such a mismatch could be eliminated by the Local Buffers. Again, to maintain synchronization between the audio stream and the video stream in the MMDU, the local buffers will be used to enforce a delay in one data stream that is proportional to the difference in the transmission and playback data rates.

Synchronization Problem Between Transmission and Playback

The synchronization problem we need to solve, in its simplest definition, is as follows: during the retrieval of multiple media streams (such as video and audio) constituting a multimedia object, it is necessary to preserve the temporal relationships that exist among those media streams. For example, assume a subscriber has just signaled the network asking the Multimedia Server to start the transmission of a specific television show. The audio and video streams representing that show are to be played back on the display unit in perfect synchronization so that a person in a scene speaking has his voice coordinate with the movement of his lips. Notice that in the MMOD environment, the multimedia presentation synchronization problem is not the same as the one presented in the PC environment (Saiedian and Awad, 1994). The reason for this difference is the presence of the overhead of the network transmission delay and the difference in the disk access and the network transmission schedules. Such delays, of course, must be added to the local synchronization problem at the playback display unit, which is equivalent to the only synchronization problem we face in the PC environment.

Before going any further in defining the problem, let us first assume the presence of a perfect network that faces no transmission delay. Let us assume further a perfect match between the disk access and the data transmission, and, moreover, a perfect match between the data rate received by the Multimedia Receiver and the playback rate on the MMDU. Given all this, we propose the following communication scenario that will provide MMOD service. The synchronization problem will be formally presented by adding the delay overhead and the mismatch overhead to the general model.

A subscriber wishing to see a specific television show will do so by signaling his Multimedia Receiver, which will construct a packet and send the packet through the transmission line to the Multimedia Server. Figure 2 shows the structure of the packet that we think is adequate to do the job. It is made as simple and compact as possible to reduce the request transmission time so that the response will also be fast. The following is a brief description of each of the fields in the request packet shown in Figure 2:

- The transmission type is a 1–bit field used to indicate that this packet is intended for the Multimedia Server. This destination is represented by a zero bit. This field serves two purposes:

  1. The other Multimedia Receivers will not "explore" any more of the packet since it is sent to the Multimedia Server as a transmission request.

  2. When the intermediate Multimedia Receivers see the zero bit, they will forward it to the next best Multimedia Receiver leading to the Multimedia Server without the need to include the Multimedia Server address as part of the packet. The zero bit will be enough since the Multimedia Server must have a well-known address.

- The show ID Field is used to identify the show requested. The size of this field depends on the capacity of the storage space; that is, the size of the field depends on the maximum number of shows the Multimedia Server can accommodate.

- The source address identifies the requesting site (host) and is also dependent on the size of the MAN, that is, dependent on the number of subscribers the Multimedia Server can serve.

Transmission Type	Show ID	Source Address

Figure 2. Transmission request packet

The packet sent back to the source host from the Multimedia Server has the structure depicted in Figure 3; its fields are described as follows:

- The transmission type field will be a 1 if the packet contains data.

- The destination address is dependent on the number of subscribers in the MMOD service network. Any Multimedia Receiver receiving a packet will first check the transmission type field. If it is 1, then it checks the destination

Transmission Type	Destination Address	Data

Figure 3. The respond packet

address to see where the packet is heading and forwards it to the best successive Multimedia Receiver.

- The data field contains the data requested in terms of multimedia blocks. Playback of media streams proceeds in terms of quanta such as video frames. If each block is assumed to contain a quantum, then retrieval of a fraction of a block cannot be used for playback, causing the playback to starve until the remaining fraction arrives, possibly in the next round. To avert such situations, the number of blocks retrieved in each service round must be an integer for all requests.

We will assume that data are continuously transmitted to the requesting site until the end of the television show.

Since it is unrealistic to assume a delay-free, mismatch-free network, we assume the presence of random network delays due to the inherently asynchronous nature of the packet network and the latencies associated with storage devices. For continuous, real-time streams of data, the problem is to ensure the proper time of playback (presentation time) of each data element in spite of random network delays. In addition to the network transmission delay, there is a problem due to the different natures of audio and video data; the video display unit might not start displaying until several frames (blocks) have been received, whereas the audio device starts playing audio samples as soon as they are received. This difference will also cause asynchrony. Synchronization problems will also occur if the actual rates at which each physical device displays data differ even slightly. If the data rates do not exactly match, audio and video output will not stay in synchrony over long periods. Another important problem mentioned in Anderson and Homsy (1991) is that scarcity of system resources (for example, network bandwidth and CPU time) may cause a stream to "starve." If input/output continue on another stream, any skew in the output will increase.

In the next section we propose a new method to solve the synchronization problem, and we call it the Distributed Synchronization Responsibility (DSR) method.

Enforcing Synchronization Using the DSR Method

The Multimedia Server will continuously perform part of the synchronization duties without the need for the Multimedia Receiver to send any requests to do so. When the data are first recorded, the Multimedia Server maintains a log of break points at which one of the data streams is to be speeded up or slowed down. Figure 4 explains this first and basic technique. It shows a time line starting with 0, marking the beginning of transmission, and ending with the time it takes to transmit the entire show. When the show is first recorded, the audio part of a scene might be faster than the corresponding video part, and consequently, the movement will not "rendezvous" with the voice intended for that movement. For these points of mismatch, the Multimedia Server will keep a log such as that shown in Figure 4. The example shown in the figure indicates that, after three time units of data transmission, the audio data rate is to be increased; that is, the audio data stream is to be speeded up. This action will continue for two units of time. Between 5 and 10 time units, the transmission rate will be the same for both the audio and video data. Notice that at time 22, the video data stream is to be speeded up for two units of time. The reason why we did not slow down the audio part instead is the desire to maintain the consistency of the multimedia presentation. You would speed up the video data stream and not slow down the audio stream simply because there is no meaning (in terms of presentation) in slowing down the voice. To slow down a video stream, you can simply slow down the video data transmission, and for presentation purposes, you will repeat the last video frame that appeared on the screen. To speed it up, you can increase the speed of video frame presentation on the screen, but the increase is usually small enough that the user does not recognize it. To speed up the audio stream, you can do the same with the voice, but, again, you cannot slow down the voice data stream.

The heart of our synchronization mechanism is that the Multimedia Server will do anything it can to coordinate data transmission and real-time presentation. This purpose is accomplished by solving any other problem as it occurs, so that the Multimedia Server will keep slowing down and speeding up data flow according to demand. The problems to be solved include the network delay problem, mismatch between any local Multimedia Receiver and its corresponding (MMDU), and the mismatch in the Multimedia Server transmission due to the latency caused by the fact that Audio Storage and Video Storage units have a high seek time. To solve any of these problems, the Multimedia Server will also have to increase or decrease the transmission rate, but these problems are only temporary. Therefore, solving any of them will give the Multimedia Server the time to solve the inherent problem of mismatch in recording (explained above).

The Multimedia Server is responsible for servicing requests from any Multimedia Receiver, and possibly from all of them at the same time. For this reason, it is unreasonable to give it complete responsibility for synchronization. We propose a new approach that will distribute the synchronization responsibility between the Multimedia Server and the local hosts. In short, the Multimedia Receiver will perform part of the job until it is unable to handle it. Then it will send the necessary information back to the Multimedia Server to synchronize its original transmission, so that the Multimedia Receiver will retain control of the data transfer to the MMDU in order to recover from the asynchrony in presentation.

We will present the synchronization technique by showing how it works when there is a local data rate mismatch between the Multimedia Receiver and its MMDU. We mentioned earlier that the Multimedia Receiver receives packets containing blocks of multimedia data and sends them up to the MMDU. Assume that the data rate the MMDU uses to display the multimedia data is less than the the rate at which the Multimedia Receiver receives data. If the Multimedia Receiver is flooded by either type of data (audio or video) because the MMDU has a slow "consumption" rate, the Multimedia Receiver has to get rid of the overloaded blocks of data.

To solve this problem, we present the following solution. The Multimedia Receiver maintains two local buffers: the Local Audio Buffer and the Local Video Buffer. These buffers will be used to hold a number of video and audio data blocks. The buffers will keep filling up until they reach some threshold value; then the Multimedia Receiver will have to send a special packet asking the Multimedia Server to slow data transmission rate. If the MMDU data "consumption" rates are different for audio and video data, the data type with the lower consumption rate will have its Multimedia Receiver buffer full before the other type. In this case, the Multimedia Receiver has to identify which stream of data is the one it is flooded with. On the other side of the network, the Multimedia Server will receive the "SOS" packet and will respond to it by decreasing the data transmission rate. To slow down or speed up one data type and not the other, or even to synchronize the transmission of both at the same time, two buffers are needed. The buffers are exactly like those used at the local host, but they will be referred to as Global Audio Buffer (GAB) and Global Video Buffer (GVB). When the audio data, for example, is the stream to be slowed down, the GAB will be used to maintain the current data blocks received from Audio Storage.

Two points must be mentioned here:

1. The buffers are actually used to guarantee the availability of a number of data blocks in case the data stream is required to be speeded up at a later time. Assume that the audio data consumption by the MMDU is slow and the Local Audio Buffer is flooding. If, after some time, the MMDU needs audio data at a rate higher than the Multimedia Transmitter can send, the Multimedia Receiver will be ready

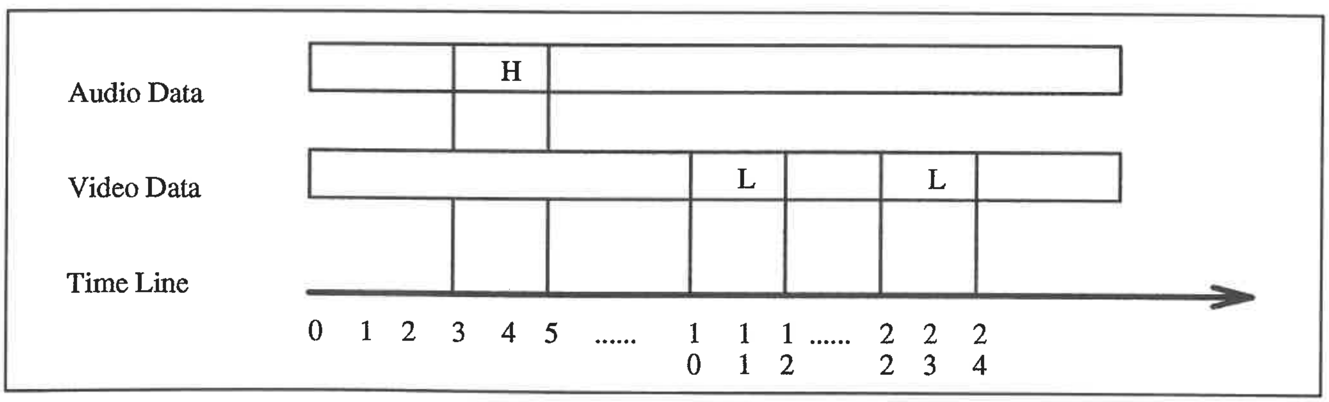

Figure 4. Multimedia server time line showing recording break points

with its previously saved data blocks. The same is true with Audio Storage, Video Storage, and the Multimedia Server, which also justifies the use of the global buffers.

2. By the data rate change we mean the unavailability of data to be transmitted or received because of transmission line delay, the high seek time of the data disks, and the like.

Using this procedure, any kind of transmission delay will be taken care of directly by either the Multimedia Server (in case of a mismatch between the Multimedia Transmitter transmission rate and the disk data retrieval rate) or the Multimedia Receiver with the help of the Multimedia Server (in case of a mismatch between the MMDU and the Multimedia Receiver transmission rates).

Summary and Concluding Remarks

In this paper we proposed a Multimedia On-Demand (MMOD) system architecture capable of solving presentation synchronization problems. Synchronization problems include the mismatch between the data transmission rate of the Audio Storage, the Video Storage, and the Multimedia Server; the mismatch between the data rate of the Multimedia Receiver and the Multimedia Display Unit (MMDU); and the transmission line delay. Using the local and global buffers at the host and the Multimedia Server sites, respectively, these problems will be solved by the Multimedia Receiver doing its part in synchronizing data transmission and presentation rates. When it can no longer do so, it will send a lightweight packet to the Multimedia Server, asking it to slow down or speed up one data stream or the other.

Our Distributed Synchronization Responsibility (DSR) technique, created to solve the synchronization problem, must be tested on real multimedia data to prove its effectiveness in enforcing presentation synchronization. There are, however, no real MMOD systems in the market that we can use for testing our architecture. Most papers published about this topic are only proposals, and none describes a real MMOD system in detail. Our objective has been to propose a model for synchronization of Multimedia On-Demand Systems. The model that we have proposed is fairly simple but practical. Naturally, in the course of implementation, many efficiency issues have to be considered. For example, data compression must be considered to achieve acceptable data communication, storage, and retrieval response times.

Acknowledgment

We would like to acknowledge the anonymous reviewers who provided us with numerous helpful comments.

References

Anderson, D., & Homsy, G. (1991). A continuous media I/O server and its synchronization mechanism. *IEEE Computer, 24*(10), 51–57.

Gibbs, S., Dami, L., & Tsichritzis, D. (1991). An object-oriented framework for multimedia composition and synchronization. In L. Kjelldahl (Ed.), *Multimedia: Systems, Interaction and Applications, 1st Eurographics Workshop* (102–111). Berlin and NY: Springer-Verlag.

Nicolaou, C. (1990). An architecture for real-time multimedia communication system. *IEEE Journal on Selected Areas in Communications, 8*(3), 391–400.

Ramanathan, S., & Rangan, P. (1993). Feedback techniques for inter-media continuity and inter-media synchronization in distributed multimedia systems. *The Computer Journal, 36*(1), 19–31.

Rangan, P., Vin, H., & Ramanathan, S. (1992). Designing an on-demand multimedia service. *IEEE Communications Magazine, 30*(7), 56–64.

Saiedian, H., & Awad, M. (1994). Managing synchronization and time factors in multimedia presentations. *Journal of Information and Software Technology, 35*(12), 653–657.

Sincoskie, W. D. (1991). System architecture for a large scale video on-demand service. *Computer Networks and ISDN Systems, 22*(2), 155–162.

Steinmetz, R. (1990). Synchronization properties in multimedia systems. *IEEE Journal on Selected Areas in Communications, 8*(3), 401–412.

Vazirgeannis, M., & Mourlas, C. (1993). An object-oriented model for interactive multimedia presentations. *The Computer Journal, 36*(1), 78–86.