Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems

S. RAMANATHAN AND P. VENKAT RANGAN

Multimedia Laboratory, Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093-0114, USA

Future advances in networking and storage will make it feasible to build multimedia on-demand servers that provide services similar to those of a neighbourhood videotape rental store over metropolitan area networks such as B-ISDN. Such multimedia servers can support real-time retrieval of multimedia objects by users onto their ISDN videophones and audiophones for playback. The design of techniques and protocols for providing continuous and synchronous access to multimedia services constitutes the subject matter of this paper. In future integrated networks, mediaphones that possess bare minimum capability to playback media but which lack the sophistication to run elaborate time synchronization protocols, may be connected directly to the network. We present rate-based feedback strategies by which, during retrieval, a multimedia server uses light-weight messages called *feedback units* transmitted periodically by mediaphones, to accurately estimate the playback instants of media units. Using these estimates, the multimedia server detects impending playback discontinuities due to buffer overruns or starvations at mediaphones, and preventively readjusts media transmission so as to avoid either of these anomalies. Given the available buffer sizes at mediaphones, we present methods by which a multimedia server can determine the minimum rate at which feedback units must be transmitted to it by the mediaphones, so as to maintain continuity of media playback. In order to guarantee synchronous playback at mediaphones, we first propose a bounded buffering technique which uses buffering limitations at the slave mediaphones to automatically enforce bounds on the asynchrony among mediaphones. Although simple and easy to implement, this technique may entail a large average asynchrony, in order to avoid which we propose a multiple feedback synchronization technique. We present initial performance comparisons of the effectiveness of these synchronization techniques. The techniques for intra-media continuity and inter-media synchronization presented in this paper form the basis of a prototype multimedia on-demand server being developed at the UCSD Multimedia Laboratory.

Received October 1992

1. INTRODUCTION

1.1. Motivation

Future advances in networking will make it feasible for digital computer networks to support multimedia communication. Coupled with rapid advances in storage technologies, they can be used to build multimedia on-demand services over metropolitan area networks such as B-ISDN, that are expected to permeate residential, organizational and educational premises in a manner similar to existing cable TV or telephone networks [16]. A multimedia on-demand server, which we will refer to as a **Multimedia Server**, provides services similar to those of a neighbourhood videotape rental store [13, 16]. It digitally stores multimedia information such as entertainment movies, educational documentaries, advertisements etc., on a large array of high capacity storage devices, and is connected to media display devices such as ISDN videophones and audiophones belonging to users via an integrated metropolitan area network (see Figure 1). Users can retrieve multimedia objects in real-time from the multimedia server over the integrated network onto their videophones and audiophones (both of which we shall, henceforth, refer to as **Mediaphones**) for playback.

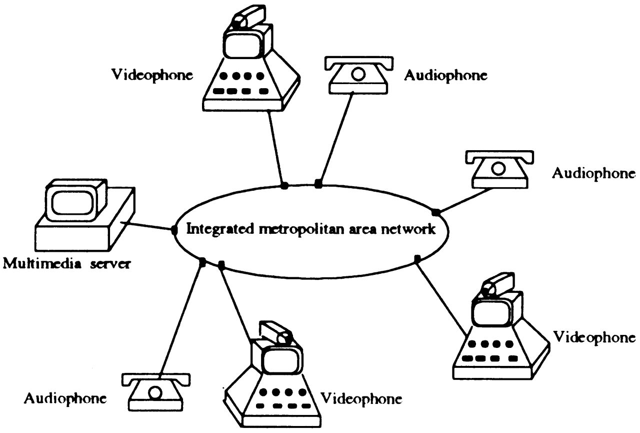

FIGURE 1. A multimedia server connected to mediaphones by an integrated high-speed metropolitan area network.

The development of multimedia-on-demand services as envisaged above offers many research challenges in the areas of storage architectures and communication protocols. Multimedia data is isochronous in nature; that is, each media stream is a sequence of finite sized samples (such as video frames and audio samples) which convey meaning only when presented continuously in time (this is unlike say, a textual object for which spatial continuity is sufficient). In addition, there may be need for synchronization between multiple media streams. Therefore, retrieval[1] of media streams must proceed so as to ensure both continuity of playback of each of the media streams, and also maintain synchronization among different media streams. The design of techniques and protocols for providing continuous and synchronous access to multimedia on-demand services over integrated networks is the subject matter of this paper. Whereas ensuring continuous retrieval amounts to preserving intra-media temporal relationships, synchronization amounts to preserving inter-media temporal relationships.

1.2. Related work

Techniques for specifying, representing, and enforcing temporal relationships between media streams have been receiving attention only recently. Steinmetz [17] presents a set of programming constructs for expressing intermedia relationships. Little *et al.* [9] propose Petri net based models for formally describing synchronization requirements among media streams, and also develop synchronization mechanisms for enforcing these requirements at the time of media retrieval from multiple servers onto a single destination. Both the model and the synchronization mechanisms do not account for variations that may exist in playback rates of different media streams. Both of the above mentioned efforts address the problem of media synchronization at a higher level than us. Nicoloau [10] proposes a two-level scheme for media communication, in which temporal relationships between media units can be specified at a logical data level and implemented at a physical data level. Shepherd *et al.* [15] propose a mechanism using the synchronization marker concept for indication of synchronization points and describe the integration of this mechanism into the OSI model. Anderson *et al.* [1] describe algorithms for recovering from loss of synchrony among interrupt-driven media I/O devices, which are mainly applicable to single-site multimedia workstations.

Protocols for media synchronization in multimedia applications such as tele-conferencing have been presented by Escobar *et al.* [4]. These protocols, which can adapt to dynamic changes in network delays, however, assume the presence of globally synchronized clocks at all times. When the clocks of the source and destination are perfectly matched, continuity of playback at the destination can be guaranteed by simply providing sufficient buffering to counteract network jitter. However, in environments such as the one we consider in this paper, globally synchronized clocks may not exist. This is because mediaphones, which are simple media capture and display subsystems directly connected to the integrated network, may lack the sophistication to run elaborate time synchronization protocols. Furthermore, the mediaphones may belong to different organizations, which may not want to synchronize their clocks across organizational domains. In addition, media streams that may have been recorded at different times may be required to be played back simultaneously (e.g. audio dubbing) and there may not be any commonality in the times of existence of their recording devices. Hence, mismatches in recording rates are inevitable in such a situation and synchronous playback cannot be ensured solely by synchronizing the clocks of the mediaphones used for playback. Discontinuities in media playback and loss of synchronization may also result from dynamic changes in network characteristics, such as congestion, etc. In all such environments, additional mechanisms are essential for maintaining continuity and synchronization.

1.3. Our contributions

In this paper, we present techniques for guaranteeing continuous and synchronous retrieval of multimedia objects from a multimedia server onto mediaphones, in the presence of network delay jitter and non-deterministic variations in rates of recording and playback. In order to ensure continuity of playback at a mediaphone, we present a feedback technique, in which light-weight messages called *feedback units* transmitted periodically by mediaphones back to a multimedia server enable the multimedia server to accurately estimate the playback instants of media units. The multimedia server uses these estimates to detect impending overruns or starvations at mediaphones, and to preventively readjust the transmission rate of media units so as to avoid either anomaly and its adverse effect on playback continuity. Given buffer sizes available at mediaphones, we present methods to determine a *minimum feedback ratio*, which is the minimum rate at which feedback units must be transmitted to maintain continuity of media playback. In order to enforce synchronization, we present a bounded buffer technique in which buffer overruns and starvation at mediaphones are used for synchronizing mediaphones, and a multiple feedback synchronization technique in which the multimedia server uses feedback units transmitted back to it by different mediaphones to monitor their playback and to steer them into mutual synchrony, by speeding up a lagging mediaphone or slowing down a leading mediaphone. We discuss how these techniques can be incorporated into the proposed OSI standard network protocol hierarchy.

The rest of the paper is organized as follows: the system architecture of a multimedia on-demand service is presented in Section 2. Feedback mechanisms for intra-media continuity are presented in Section 3. Tech-

[1] In this paper, we will use the terms retrieval and playback synonymously.

niques for inter-media synchronization are discussed in Section 4. The integration of the continuity and synchronization techniques into existing OSI standards is explored in Section 5. Performance evaluation of the feedback techniques is presented in Section 6. Finally, Section 7 concludes the paper.

2. SYSTEM ARCHITECTURE

The system architecture of a multimedia on-demand service consists of a multimedia server connected to mediaphones $P_1, P_2, \dots, P_m$ of users by an integrated broadband network (see Figure 2). The mediaphones are simple devices capable of digitizing and transmitting, or receiving and playing back media units, but lack the sophistication to run elaborate time synchronization protocols, and hence, may have mismatches in rates of recording and playback. Let θ denote the nominal period of each media unit being generated or played back at any mediaphone, and $\pm\rho$ denote its maximum fractional drift. These drifts are small enough for us to neglect higher powers of ρ and thereby approximate $1/1-\rho$ to $(1+\rho)$ and $1/1+\rho$ to $(1-\rho)$. As a result, the actual period $\theta(n)$ of a media unit n varies between $\theta*(1-\rho)$ and $\theta*(1+\rho)$.

Asynchronous Transfer Mode (ATM) networks, which are emerging as the preferred means for multimedia transmission, can guarantee bounds on the maximum variations in network delays via resource reservation and admission control at the network [6]. Therefore, we assume that the network delays experienced by media units transmitted by the multimedia server to the mediaphones experience non-deterministic communication delays bounded between $\Delta^m_{\min}$ and $\Delta^m_{\max}$ and feedback units transmitted by the mediaphones to the multimedia server experience delays bounded between $\Delta^f_{\min}$ and $\Delta^f_{\max}$. Although in the rest of this paper, the delay bounds are assumed to be fixed at the time of commencement of retrieval, the feedback techniques proposed are applicable even to cases when the delay bounds vary depending on the network load; in order to capture these variations, network delays have to be measured periodically and the delay bounds updated. As a convention, all time measurements are assumed to have been mapped onto the clock of the multimedia server. For time measurements for which there may be uncertainties in exact values (usually, arising out of variations in network delays and playback rates), superscripts e and l are used to denote their earliest and latest estimates. Table 1 summarizes all of the above definitions and conventions.

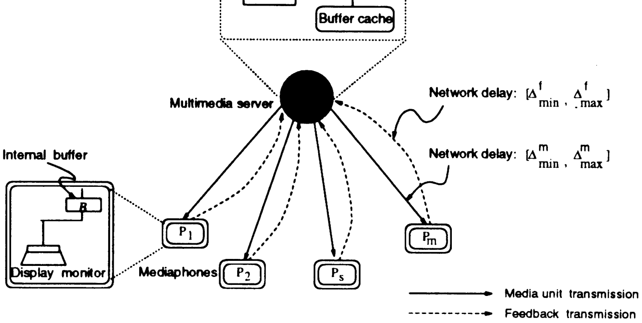

FIGURE 2. System architecture of a multimedia on-demand service.

3. CONTINUITY OF MULTIMEDIA PLAYBACK

One of the distinguishing features unique to playback of digital multimedia is the requirement of continuity. For continuous playback of a media stream at a mediaphone, it is essential to ensure timely availability of all media units of the media stream, i.e. all media units of that stream must be available at the mediaphone prior to their scheduled times of playback. Delayed arrival of media units leads to starvation at the mediaphones, causing discontinuities in playback. The latest instants by which the multimedia server must transmit media units so as to ensure their timely availability at the mediaphone, assuming that the mediaphones commence playback as soon as the first media unit is received, depends on both the network delay jitter and the playback rate variation, and is exactly computed in the following theorem:

THEOREM 1. *In order to guarantee continuous playback of a media stream at a mediaphone, the multimedia server must transmit a media unit μ of the stream to the device no later than $(\mu-1)*\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min})$ after the transmission of the first media unit to that mediaphone.*

Proof. The scheduled playback time $p(\mu)$ of media

TABLE 1. Nomenclature used in this paper

Symbol	*Explanation*	*Unit*
$\Delta^m(\mu)$	Network delay of media unit μ between a mediaphone and the multimedia server	sec
$\Delta^m_{\min}$	Minimum network delay of a media unit	sec
$\Delta^m_{\max}$	Maximum network delay of a media unit	sec
$\Delta^f_{\min}$	Minimum network delay of a feedback unit	sec
$\Delta^f_{\max}$	Maximum network delay of a feedback unit	sec
$\tau(\mu)$	Transmission time of media unit μ from the multimedia server to a mediaphone	sec
$a(f_\mu)$	Arrival time of feedback unit f_μ at the multimedia server	sec
$p(\mu)$	Playback time of a media unit μ at a microphone	sec
θ	Nominal period of recording or playback	sec
ρ	Fractional drift in playback period θ of a media unit	fraction
$\mathscr{B}$	Buffering capacity of a mediaphone	media units
$\mathscr{P}$	Number of media units prefetched prior to commencement of playback	media units
$\mathscr{F}$	Feedback ratio, i.e. ratio of feedback units transmitted to media units played back by a mediaphone	fraction

unit μ at the mediaphone is the sum of the following two factors:

1. $p(1)$: the instant at which the playback of the first media unit commences at the mediaphone; if $\tau(1)$ is the time at which the transmission of the media stream began, $p(1)$ is guaranteed to lie between $\tau(1)+\Delta^m_{\min}$ and $\tau(1)+\Delta^m_{\max}$.
2. the summation of playback periods of media units 1 through $\mu-1$; each of these periods is guaranteed to lie between $\theta*(1-\rho)$ and $\theta*(1+\rho)$.

The above two factors yield the interval $[p^e(\mu), p^l(\mu)]$ within which $p(\mu)$ is guaranteed to lie, where:

$$p^e(\mu)=\tau(1)+\Delta^m_{\min}+(\mu-1)*\theta*(1-\rho) \quad (1)$$

$$p^l(\mu)=\tau(1)+\Delta^m_{\max}+(\mu-1)*\theta*(1+\rho) \quad (2)$$

Suppose that media unit μ is transmitted by the multimedia server at $\tau(\mu)$ after which, it may suffer a network delay of at most $\Delta^m_{\max}$ and arrive at the mediaphone latest by $a(\mu)$, where $a(\mu)=\tau(\mu)+\Delta^m_{\max}$. In order to ensure continuity, its arrival time must not exceed its earliest playback time, which yields that:

$$\tau(\mu)\leqslant p^e(\mu)-\Delta^m_{\max}=\tau(1)+(\mu-1)*\theta*(1-\rho) - (\Delta^m_{\max}-\Delta^m_{\min}) \quad (3)$$

$$\Rightarrow \tau(\mu)-\tau(1)\leqslant(\mu-1)*\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min}) \quad (4)$$

■

It is interesting to note that, by Theorem 1, the second media unit must be transmitted no later than $\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min})$ after the first media unit, which, if it is to be non-negative, the following condition must be satisfied:

$$\theta*(1-\rho)\geqslant\Delta^m_{\max}-\Delta^m_{\min} \quad (5)$$

When Equation (5) is not satisfied, i.e., $\theta*(1-\rho)<\Delta^m_{\max}-\Delta^m_{\min}$, a prefetch of media units in advance of the start of playback is necessary (i.e. it may not be possible to commence playback immediately upon arrival of the first media unit). To see why, notice that the separation between the arrivals of the first and the second media units can be as large as $\Delta^m_{\max}-\Delta^m_{\min}$ (which happens when the first media unit has suffered a delay of $\Delta^m_{\min}$ and the second a delay of $\Delta^m_{\max}$). If this difference exceeds the duration of playback of the first media unit (which can be as low as $\theta*(1-\rho)$), a discontinuity will result at the end of its playback. In order to avoid such a discontinuity, a finite number of media units would have to be prefetched and buffered before the commencement of playback, the exact computation of which is discussed next.

Suppose that the prefetch (in media units) necessary for guaranteeing continuity of playback at a mediaphone is $\mathscr{P}$. Since playback at the mediaphone commences only after $\mathscr{P}$ media units have been received, the earliest possible time of commencement of playback is $\tau(\mathscr{P})+\Delta^m_{\min}$, where $\tau(\mathscr{P})$ is the time at which media unit $\mathscr{P}$ is transmitted by the multimedia server to the mediaphone. Following the same procedure adopted in Theorem 1, it can be determined that the condition for timely availability of any media unit μ, $\mu>\mathscr{P}$ is that the transmission time $\tau(\mu)$ of media unit μ should satisfy the following condition:

$$\tau(\mu)\leqslant\tau(\mathscr{P})+(\mu-1)*\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min}) \quad (6)$$

Thus, the latest instant at which a media unit $\mu>\mathscr{P}$, can be transmitted without causing discontinuity, $\tau^l(\mu)$ is:

$$\tau^l(\mu)=\tau(\mathscr{P})+(\mu-1)*\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min}) \quad (7)$$

Clearly, $\tau^l(\mu)$ should be an increasing function of $\mu\geqslant\mathscr{P}$, that is, given $\mu_2>\mu_1\geqslant\mathscr{P}$, it should be the case that $\tau^l(\mu_2)>\tau^l(\mu_1)$. That this condition is satisfied for all values $\mu_2>\mu_1>\mathscr{P}$ is directly evident from Equation (7). For the case when $\mu_1=\mathscr{P}$, since the necessary prefetch is $\mathscr{P}$, it must be the case that $\tau^l(\mathscr{P}+1)>\tau^l(\mathscr{P})$. The difference $\tau^l(\mathscr{P}+1)-\tau^l(\mathscr{P})$ is obtained from Equation (7) to be:

$$\tau^l(\mathscr{P}+1)-\tau^l(\mathscr{P})\leqslant\mathscr{P}*\theta*(1-\rho)-(\Delta^m_{\max}-\Delta^m_{\min})$$

For $\tau(\mathscr{P}+1)-\tau(\mathscr{P})$ to be non-negative:

$$\mathscr{P}*\theta*(1-\rho)\geqslant\Delta^m_{\max}-\Delta^m_{\min}$$

which yields the minimum prefetch required for continuous retrieval:

$$\mathscr{P}=\left\lceil\frac{(\Delta^m_{\max}-\Delta^m_{\min})}{\theta*(1-\rho)}\right\rceil \quad (8)$$

In all of the above computations, we have assumed that network delay experienced by each of the media units transmitted after the prefetch can be as large as $\Delta^m_{\max}$, and that the playback period of each media unit can be as small as $\theta*(1-\rho)$, since these represent the most stringent requirements for continuity. In practice, network delays experienced by media units may be much smaller than $\Delta^m_{\max}$. Furthermore, playback rate variations may cause periods of media units to be longer than $\theta*(1-\rho)$. Both of these anomalies lead to accumulation of media units at mediaphones, to accommodate which buffering is required. Methods to compute the buffering requirements at mediaphones are elaborated next.

3.1. Buffering requirements of continuity

Buffering at mediaphones serve to counteract variations in network delays and playback rates. Whereas, the minimum required buffering at a mediaphone is equal to the prefetch $\mathscr{P}$ (given by Equation (8)), the maximum buffering required to preserve continuity is precisely computed by the following theorem:

THEOREM 2. *The buffering required at a mediaphone in order to guarantee continuity of playback of a media* *stream containing μ media units is given by:*

$$\left\lceil \frac{2*(\Delta^m_{\max}-\Delta^m_{\min})+2*\theta*\rho*(\mu-1)}{\theta*(1+\rho)} \right\rceil$$

Proof. Suppose that μ' is the latest media unit whose playback has been initiated prior to the arrival of media unit μ at the display device, that is,

$$p(\mu')\leqslant a(\mu)$$

If $\mathscr{P}$ is the prefetch, and $\Delta^m(\mathscr{P})$ is the network delay of the last media unit which is prefetched, the instant at which playback commences is $\tau(\mathscr{P}+\Delta^m(\mathscr{P}))$. Since $\mu'-1$ media units are played back before μ' is played back, the instant of playback initiation of μ' is:

$$p(\mu')=\tau(\mathscr{P})+\Delta^m(\mathscr{P})+\sum_{i=1}^{i=\mu'-1}\theta(i)$$

Therefore, we get

$$\tau(\mathscr{P})+\Delta^m(\mathscr{P})+\sum_{i=1}^{i=\mu'-1}\theta(i)\leqslant a(\mu)$$

Rewriting by using averages to express summation and rearranging, we obtain that μ' is the largest value such that:

$$\mu'\leqslant\frac{a(\mu)-\tau(\mathscr{P})-\Delta^m(\mathscr{P})}{\theta_{\text{avg}}}+1 \qquad (9)$$

The buffering, which is given by $\mu-\mu'$, is maximum when μ' is minimum, which occurs when (1) $a(\mu)$ is minimum, which is obtained by adding $\Delta^m_{\min}$ to $\tau(\mu)$ given by Equation (6) to be: $a(\mu)=\tau(\mathscr{P})+2*\Delta^m_{\min}+(\mu-1)*\theta*(1-\rho)-\Delta^m_{\max}$, (2) $\Delta^m(\mathscr{P})$ is maximum, which is $\Delta^m_{\max}$, and (3) θ_{avg} is maximum, which is $\theta*(1+\rho)$ (that is, the playback period at the mediaphone is uniformly the maximum possible value of $\theta*(1+\rho)$). In other words, buffering is maximum when the playback commences at the latest possible instant and progresses at the slowest possible rate, but μ suffers the minimum network delay. Under these conditions, $\mu-\mu'$ can be obtained to be:

$$\mu-\mu'\leqslant\frac{2*(\Delta^m_{\max}-\Delta^m_{\min})+2*\theta*\rho*(\mu-1)}{\theta*(1+\rho)} \qquad (10)$$

The maximum value of the difference, $\mu-\mu'$ exactly represents the buffering capacity that is required at the mediaphone to avoid loss of media unit μ due to a buffer overrun when it arrives at the mediaphone, and is given by:

$$\mathscr{B}=\left\lceil \frac{2*(\Delta^m_{\max}-\Delta^m_{\min})+2*\theta*\rho*(\mu-1)}{\theta*(1+\rho)} \right\rceil \qquad (11)$$

■

If the buffering available at a mediaphone is less than $\mathscr{B}$ given by Theorem 2, buffer overruns may occur leading to media losses (consequently causing discontinuities in playback). In the expression for $\mathscr{B}$, the first term,

$$\frac{2*(\Delta^m_{\max}-\Delta^m_{\min})}{\theta*(1+\rho)}$$

represents the buffering needed to counteract jitter in network delays and is independent of the size of the media stream. The second term,

$$\frac{2*\rho*(\mu-1)}{(1+\rho)}$$

represents the buffering required to counteract playback rate variations, and increases directly with the size μ of a media stream, and is undesirable in practice. Hence, additional mechanisms are necessary for guaranteeing continuity of playback of media streams.

We now present a constant rate feedback technique, in which the multimedia server uses feedback message units periodically transmitted back to it by mediaphones, to more accurately estimate the actual playback times of media units at those mediaphones; using these estimates, the multimedia server tries to predict impending buffer overruns or starvations at mediaphones, and preventively readjust media transmission rate so as to avoid both anomalies and their adverse effects on playback continuity at those devices.

3.2. Feedback technique for bounded buffer continuity

At the time of playback of media units, a mediaphone transfers the data parts of media units from its buffers to its display monitor. Simultaneously with the transfer of data parts of *selected* media units (but *not* necessarily every media unit) to its display monitor, the mediaphone feeds part of the headers of these selected media units back for transmission as a feedback unit to the multimedia server (see Figure 3). Each feedback unit is, thus, a light-weight message containing only the number of the media unit (but the data part stripped off) that was concurrently played back at the time of the feedback unit's transmission; hence, its transmission imposes little overhead on the network. A binary flag (such as the one proposed in [11]) present in the headers of a media unit distinguishes media units for which feedback units have to be transmitted when playback of those media units is initiated (as we will see shortly, such a flag can be set

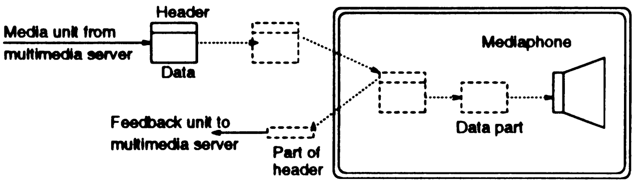

FIGURE 3. Feedback transmission at a mediaphone: The mediaphone transmits part of the headers of selected media units back to the multimedia server concurrently with the playback of data parts of these media units.

a priori by the multimedia server before the commencement of the playback).

Suppose that a mediaphone transmits a feedback unit concurrently with the playback initiation of media unit μ. The multimedia server, upon receiving this feedback unit f_μ at time $a(f_\mu)$, can estimate the earliest and latest possible times, $p^e(\mu)$ and $p^l(\mu)$ at which playback of media unit μ could have been initiated:

$$p^e(\mu) = a(f_\mu) - \Delta^f_{\max} \tag{12}$$

$$p^l(\mu) = a(f_\mu) - \Delta^f_{\min} \tag{13}$$

$I(\mu) = [p^e(\mu), p^l(\mu)]$ is called a **Playback Initiation Interval** of media unit μ. From Equations (12) and (13), we have:

$$p^l(\mu) - p^e(\mu) = \Delta^f_{\max} - \Delta^f_{\min} \tag{14}$$

Since media units are played back periodically, the multimedia server can also estimate the earliest and latest playback times of a subsequent media unit μ':

$$p^e(\mu') = p^e(\mu) + (\mu' - \mu) * \theta * (1 - \rho) \tag{15}$$

$$p^l(\mu') = p^l(\mu) + (\mu' - \mu) * \theta * (1 + \rho) \tag{16}$$

$[p^e(\mu'), p^l(\mu')]$ constitutes a playback initiation interval $I(\mu')$ of media unit μ'.

Having estimated the playback initiation interval of a subsequent media unit μ', the multimedia server can determine the earliest time at which μ' *can* be transmitted to a mediaphone so as to avoid buffer overrun at that mediaphone, as well as the latest time by which μ' *should* be transmitted to the mediaphone so as to avoid starvation at the mediaphone. In order to compute the earliest transmission time $\tau^e(\mu')$ of μ', note that its earliest arrival time $\tau^e(\mu') + \Delta^m_{\min}$ at the mediaphone should not precede the latest playback time of media unit $\mu' - \mathscr{B}$, in order to avoid a buffer overrun. That is:

$$\tau^e(\mu') + \Delta^m_{\min} \geqslant p^l(\mu' - \mathscr{B})$$

$$\Rightarrow \tau^e(\mu') \geqslant p^l(\mu) + (\mu' - \mathscr{B} - \mu) * \theta * (1 + \rho) - \Delta^m_{\min} \tag{17}$$

Similarly, the latest transmission time $\tau^l(\mu')$ of μ' should be such that its latest arrival time at the mediaphone, $\tau^l(\mu') + \Delta^m_{\max}$ does not exceed its earliest playback time, $p^e(\mu')$, in order to avoid starvation:

$$\tau^l(\mu') + \Delta^m_{\max} \leqslant p^e(\mu')$$

$$\Rightarrow \tau^l(\mu') \leqslant p^e(\mu) + (\mu' - \mu) * \theta * (1 - \rho) - \Delta^m_{\max} \tag{18}$$

Equations (17) and (18) define the earliest and latest possible transmission times of a media unit, μ' transmitted after the reception of a feedback unit f_μ, and the difference between them represents the transmission interval of μ':

$$\begin{aligned}\tau^l(\mu') - \tau^e(\mu') &= p^e(\mu) + (\mu' - \mu) * \theta * (1 - \rho) - \Delta^m_{\max} - p^l(\mu) \\ &\quad - (\mu' - \mu - \mathscr{B}) * \theta * (1 + \rho) + \Delta^m_{\min} \\ &= \mathscr{B} * \theta * (1 - \rho) - 2 * \theta * \rho * (\mu' - \mu - \mathscr{B}) \\ &\quad - (p^l(\mu) - p^e(\mu)) - (\Delta^m_{\max} - \Delta^m_{\min})\end{aligned} \tag{19}$$

Thus, the reception of a feedback unit enables the multimedia server to compute the transmission intervals of all subsequent media units. It can be observed from Equation (19) that, the greater the difference between μ and μ' (which is the case when feedback transmission is less frequent), the smaller is the duration of the transmission interval. Eventually, the transmission interval becomes negative, at which point, the multimedia server cannot proceed with any more transmissions without receiving additional feedbacks. The maximum value, $\mu'_{\max}$ for which the transmission interval is non-negative corresponds to the value beyond which the multimedia server *has* to receive a subsequent feedback unit in order to guarantee continuity, and is given by:

$$\mu'_{\max} = \left\lfloor \mu + \mathscr{B} + \frac{\mathscr{B} * \theta * (1 - \rho) - (p^l(\mu) - p^e(\mu)) - (\Delta^m_{\max} - \Delta^m_{\min})}{2 * \theta * \rho} \right\rfloor \tag{20}$$

Substituting for $p^l(\mu) - p^e(\mu)$ from Equation (14) in Equation (20), we have:

$$\mu'_{\max} = \left\lfloor \mu + \frac{\mathscr{B} * \theta * (1 + \rho) - (\Delta^f_{\max} - \Delta^f_{\min}) - (\Delta^m_{\max} - \Delta^m_{\min})}{2 * \theta * \rho} \right\rfloor \tag{21}$$

The latest transmission time of media unit $\mu'_{\max}$ is the deadline for reception of the next feedback unit. Given the earliest playback time $p^e(\mu'_{\max}) = p^e(\mu) + (\mu'_{\max} - \mu) * \theta * (1 - \rho)$, and the maximum network delay $\Delta^m_{\max}$, the latest transmission time of $\mu'_{\max}$ is: $p^e(\mu'_{\max}) - \Delta^m_{\max}$, which is also the deadline for arrival of the next feedback unit. Since the feedback unit itself may experience a delay as high as $\Delta^f_{\max}$, its transmission deadline is given by: $p^e(\mu'_{\max}) - \Delta^m_{\max} - \Delta^f_{\max}$. If the transmission of this feedback unit is to be concurrent with the playback initiation of a media unit $\mu_{\max}$ at the mediaphone, the latest playback time of $\mu_{\max}$ must not exceed the feedback unit's transmission deadline:

$$p^l(\mu_{\max}) \leqslant p^e(\mu'_{\max}) - \Delta^m_{\max} - \Delta^f_{\max}$$

$$\begin{aligned}&\Rightarrow p^l(\mu) + (\mu_{\max} - \mu) * \theta * (1 + \rho) \\ &\qquad \leqslant p^e(\mu) + (\mu'_{\max} - \mu) * \theta * (1 - \rho) - \Delta^m_{\max} - \Delta^f_{\max}\end{aligned}$$

Substituting for $\mu'_{\max} - \mu$ from Equation (21) and for $p^l(\mu) - p^e(\mu)$ from Equation (20), the difference $\mu_{\max} - \mu$ is:

$$\mu_{\max} - \mu \leqslant \frac{\left(\dfrac{\mathscr{B} * \theta * (1 + \rho) - (\Delta^f_{\max} - \Delta^f_{\min}) - (\Delta^m_{\max} - \Delta^m_{\min})}{2 * \theta * \rho}\right) * \theta(1 - \rho) - 2 * \Delta^f_{\max} + \Delta^f_{\min} - \Delta^m_{\max}}{\theta * (1 + \rho)} \tag{22}$$

This difference, $\mu_{\max} - \mu$ represents the maximum number of media units that can be played back at a mediaphone between transmission of successive feedback units, at the same time avoiding discontinuities in media playback owing to buffer overruns or starvation. The minimum value of the ratio of the number of feedback units to the number of media units is given by:

$$\textbf{Minimum Feedback Ratio} = \mathscr{F} = \frac{1}{\mu_{\max} - \mu}$$

Substituting for the difference, $\mu_{\max} - \mu$ from Equation (22) and simplifying the resulting equation, we derive the following theorem which precisely computes the minimum ratio of feedback units transmitted to media units played back at a mediaphone:

THEOREM 3. *In order to guarantee continuity of playback of a media stream at a mediaphone with a buffering capacity of $\mathscr{B}$, the feedback ratio must at least equal:*

$$\mathscr{F} = \frac{2 * \theta * \rho * (1 + \rho)}{\mathscr{B} * \theta * (1 - \rho^2) - \Delta^f_{\max} * (1 + 3 * \rho) + \Delta^f_{\min} * (1 + \rho) - \Delta^m_{\max} * (1 + \rho) + \Delta^m_{\max} * (1 - \rho)}$$

■

Transmission of feedback units at the constant rate of $\mathscr{F}$ enables the multimedia server to guarantee continuity of media playback. All the factors necessary for the computation of $\mathscr{F}$ above are available to the multimedia server at the time of commencement of media retrieval requests by users. Hence, the multimedia server can predetermine the media units $k * (1/\mathscr{F}) + 1$ $(k = 1, 2, \ldots)$ at the time of whose playback, the mediaphones must send feedback; thus the multimedia server can *a priori* set the feedback flags in the headers of such media units before transmitting them to the mediaphones.

Upon receiving a feedback unit, the multimedia server estimates the earliest and latest playback times of media units which are yet to be transmitted to the mediaphones. Based on these estimates, the multimedia server computes the earliest and the latest possible transmission times of these media units. Whereas transmission of media units at their earliest transmission times maintains the buffer occupancy at a high value (close to $\mathscr{B}$), transmission of media units at their latest possible times maintains buffer occupancy at a minimum. Whereas frequent transmission of feedbacks (i.e. a high feedback ratio) enables the multimedia server to make more precise estimates of playback times of media units much more frequently, resulting in a finer adjustment of the transmission rate to match the actual playback rate, it also imposes additional overheads on the network, the mediaphones and the multimedia server. On the other hand, the smaller the frequency of feedback transmission (i.e. the lower the feedback ratio), the smaller is the transmission interval of media units, and the smaller is the flexibility in protocol scheduling and network management.

The applicability of these feedback-based continuity techniques is not limited to stored media applications, but can also be employed for live media communication (such as those between participants in a multimedia conference). The feedback units in this case can be used to adaptively control the rate of generation of media units at the source in a manner similar to that suggested in [3].

3.3. Continuity mechanisms for wide area networks

In the feedback technique for intra-media continuity, described earlier, the entire responsibility of ensuring continuous playback rests on the multimedia server. However, in a wide area network, the network delays may be significantly larger than the corresponding values for a metropolitan area network, thereby making buffering requirements prohibitively high at the mediaphones. In such cases, buffering may be distributed among the intermediate nodes in the path from the multimedia server to the mediaphones. Each intermediate node serves as a pseudo sink for its immediate predecessor and as a pseudo source for its immediate successor in the path from the multimedia server to the mediaphones. Prior to the commencement of playback at the mediaphones, each intermediate node must prefetch and buffer the required number of media units necessary both to counteract network jitter on its link to its immediate predecessor and to satisfy the prefetch requirements of all its successors. Network jitter may be different for different links, and therefore, each node independently determines the rate at which feedback units have to be transmitted back to it by its immediate successor. A similar distributed buffering scheme but targeted to an environment with globally synchronized clocks is presented by Ferrari [5].

4. SYNCHRONIZATION OF MULTIMEDIA PLAYBACK

During the simultaneous retrieval of multiple media streams (such as video and audio) constituting a multimedia object, it is required not only to maintain continuity of playback (using the feedback techniques described in Section 3), but also to preserve the temporal relationships that existed among the media streams at the time of their recording [14]. The different media streams constituting a multimedia object may be transmitted over different network links (since different media have different QOS requirements), and may, hence, experience widely differing network delays. In addition, they may be played back at different mediaphones (e.g., audio at audiophones, video at videophones) whose clocks may not be synchronized, and hence, there may be variations in their playback rates. As a result, the media streams may go out of synchrony soon after the commencement of their playback. We now compute the maximum possible asynchrony between mediaphones, and then present synchronization techniques to maintain the maximum asynchrony within application-designated tolerable limits.

At the commencement of playback, any two mediaphones may be offset by as much as $\Delta^m_{\max} - \Delta^m_{\min}$ in their playback start times, due to network jitter. Thereafter, due to rate mismatches (the maximum fractional values of which is $\pm\rho$), the two mediaphones may playback at the fastest and slowest rates, respectively, in the worst case. Asynchrony between the two mediaphones will increase as playback progresses, reaching a maximum of:

$$\mathscr{A} = \left\lceil \frac{(\Delta^m_{\max} - \Delta^m_{\min}) + 2 * \theta * \rho * n_s}{\theta * (1 - \rho)} \right\rceil \tag{23}$$

where $\mathscr{A}$ represents the maximum asynchrony (in number of media units) between the two mediaphones at the time the slowest mediaphone is playing back media unit n_s [13]. It may be observed from Equation (23) that the maximum asynchrony increases linearly with progression of media playback (i.e., n_s), and is unacceptable in practice. Hence, additional mechanisms are required for enforcing synchronization between media streams.

In order to facilitate resynchronization of playback of media streams at mediaphones, the multimedia server determines and notes down temporal relationships between media streams at the time of their recording. Since the media streams may be recorded at different mediaphones, possibly at different times, but may be required to be played back synchronously, it is convenient to represent their temporal relationships in the form of *Relative Time Stamps* (RTSs). The RTS of a media unit represents its time of playback relative to the commencement of playback. Each media unit is assigned a RTS by the multimedia server at the time of recording. The RTS of a media unit may be as simple as its sequence number (in the case of matching recording rates), or may have to be determined by the multimedia server by comparing its estimates of recording times of media units, relative to the start of recording (techniques for RTS assignment are described in [12]). Simultaneity of playback of a set of media units is indicated by equality of their RTSs.

The multimedia server, since it maintains the RTSs, is best suited to handle synchronization during playback with little additional overhead. In order to resynchronize mediaphones that have gone out of synchrony, the multimedia server may have to speed up some mediaphones and slow down some others, thereby causing breaks in continuity of their playback. The playback of at most one stream, which we shall call the master, can be spared from such discontinuities. Whereas the master always plays back at its natural rate, all other streams, which take on the role of slaves, may be subject to skips and pauses in order to remain synchronized with the master. The choice of the master stream is dependent on the application. For example, when viewing a multimedia document, if smoothness of audio playback is of utmost importance, the audio stream serves as the master and drives the playback. The video stream, being the slave, may be subject to skips or pauses in order to synchronize its playback with that of audio.

In the following subsections, we present two techniques for synchronizing playback at mediaphones. The first technique uses buffering limitations at the slave mediaphones to automatically enforce bounds on the asynchrony between the slave and master mediaphones. (Continuity at the master mediaphone is ensured using the very same feedback technique described in Section 3.) In the second technique, feedback units are transmitted by the slave mediaphones (in addition to the master), using which the multimedia server estimates the playback rates at those mediaphones relative to the master, and speeds them up or slows them down to retain them in synchrony with the master.

4.1. Synchronization using bounded buffering

In earlier sections, we saw how, given a bounded amount of buffering at a mediaphone, feedback units transmitted by that mediaphone can be used by the multimedia server to guarantee continuous playback, even in the face of jitter in network delays and variations in playback rates. At the time of retrieval of multiple media streams, playback at the master is required to be continuous, in order to ensure which, the multimedia server uses feedback units transmitted by the master (as described in Section 3). Playbacks at the slaves are only required to be synchronized with the master. In the face of mismatches in playback rates at the master and slave mediaphones, synchronization can be accomplished only by forcible speeding up of a lagging slave or slowing down of a leading slave. Therefore, breaks in continuity of playback at the slaves are inevitable. Since playbacks at the slaves are not required to be continuous, in this technique, slave mediaphones do not transmit feedback units to the multimedia server. In order to synchronize the slave mediaphones with the master, the multimedia server, when it transmits a media unit to the master, simultaneously transmits media units with the same RTS to all the slaves also. Consequently, if a slave is leading the master, media units may arrive at the slave later than their scheduled times of playback, automatically causing the slave to starve, thereby forcing it to wait until the master catches up. On the other hand, if a slave is lagging the master, media units will arrive at the slave earlier than their scheduled playback times, and will be buffered until the buffer becomes filled. Any further early arrivals will cause the buffer to overrun, automatically causing the lagging slave to skip media units, thereby forcing it to catch up with the master.

This synchronization technique is both simple, and easy to implement. Feedback units are transmitted only by the master, and there are no additional network or computational overheads other than those required for maintaining continuity at the master. However, this technique is not without its drawbacks. Since media units are skipped only when the number of units that need to be buffered at a slave exceeds its buffering capacity, the maximum possible lag of the slave relative to the master can be as high as the buffering capacity of the slave. Likewise, the maximum lead of a slave relative to the master is governed by the buffering capacity at the master. This linking of maximum asynchrony to buffering capacities of mediaphones, which may be determined based on continuity requirements, rather than asynchrony tolerances, restricts the flexibility of this technique. Furthermore, there are situations in which the buffer at either the slave or the master may remain filled, causing the lag or the lead of the slave to reach the maximum and remain at that value, with buffer overrun or starvation at that juncture leading to a temporary reduction of the lag or lead by at most one media unit only[2]. We now propose a synchronization technique which overcomes all of these drawbacks.

4.2. Multiple feedback technique for synchronization

In this technique, both master and slave mediaphones transmit feedbacks to the multimedia server at the time of playback of selected media units. The multimedia server, upon receiving a feedback unit from a master mediaphone, carries out the procedures outlined in Section 3 for maintaining continuity at the master. In addition, when it receives a feedback unit from a slave mediaphone, it carries out the following procedure to synchronize the slave with the master:

- Using Equations (12) and (13), the multimedia server estimates the earliest and latest possible playback times of the media unit corresponding to the feedback unit received from the slave.
- By comparing the estimates of playback times of media units at the master and slave mediaphones, the multimedia server determines sets of media units that are being played back simultaneously at the master and the slaves.
- Mismatches in RTSs of master and slave media units being played back simultaneously are symptomatic of asynchrony; the difference in RTSs reflects the extent of asynchrony between the master and slave mediaphones.
- The multimedia server forces a slave to speed up or slow down by the extent by which it lags or leads, respectively, relative to the master, thereby steering it back to synchrony with the master.

As before, frequent transmission of feedbacks enables more frequent and more precise detection of asynchrony. Given an application-designated tolerable limit on the asynchrony, the minimum rate of feedback transmission can be determined so as to guarantee that the maximum asynchrony does not exceed the tolerable limit [13], thereby enabling the decoupling of asynchrony tolerances from buffer sizes. Such a multiple feedback synchronization technique overcomes all the drawbacks of the bounded buffer technique. This technique is highly flexible, permitting the choice of the asynchrony tolerance limit as well as the master media stream to be changed on-the-fly by the application. The synchronization techniques presented in this section can be extended for wide area network environments in a manner similar to that described in Section 3.3, by distributing the synchronization function among intermediate nodes on the network. In Section 6, we present some initial performance comparisons of the bounded buffer and feedback synchronization techniques.

5. INTEGRATING CONTINUITY AND SYNCHRONIZATION INTO OSI PROTOCOL MODEL

Computer communication software is generally structured as a hierarchy of protocol layers, with each layer building additional functionality on top of services exported by its lower layers. The seven-layered OSI reference model has emerged as the standard for protocol design. Support for handling multimedia must be provided at each of the seven layers. The lower layers, namely, the physical, data link and network layers should provide support for real-time transmission without regard to the type of media, whether audio or video, only considering the extent of delay and bandwidth guarantee requirements [8]. The next higher layer, namely, the transport layer, provides end-to-end delivery of messages, rate-based flow control, sequencing and deadline-based transmission of media units. Above the transport layer is the session layer, which supports negotiation and end-to-end connection establishment. Unlike the session layer and the other layers below it, which deal with media *data* units, the presentation and application layers only handle *information* objects. Whereas the presentation layer performs functions such as compression, error concealment by use of visual redundancy, ciphering etc. [8], the application layer, which interfaces to users, is best suited for specification of intra-media and inter-media temporal relationships. These relationships may be explicitly specified by the creator of the media streams, using tools such as timed petri-nets [9] and synchronization graphs, or may be implicitly determined at the time of recording. Actual enforcement of these higher-level specifications is left to the lower layers.

The session layer, since it deals with setup, management and control of end-to-end connections, is best suited for implementation of the feedback techniques for

[2] Alternative schemes may be employed for buffer management at the mediaphones, but for all such schemes, there are cases in which the asynchrony remains equal to the buffering capacity of the mediaphones.

ensuring intra-media continuity, described in Section 3 (this choice is consistent with the views expressed by Little *et al.* in [9] and by Shepherd *et al.* in [15]). This is because, ensuring continuity of media streams, which may involve slowing down or speeding up media transmission based on the arrival of feedback units, can neither be handled at upper layers where media information objects rather than media data units are handled, nor at the lower layers which do not support control and management of end-to-end connections.

Support for continuous media transmission at the session layer not only requires enhancement of some of the existing session layer primitives but also requires additional primitives performing functions specific to media communication (such as services for remote connection establishment, prefetching etc.) The need for a remote connection establishment/release facility at the session layer in multimedia services has been pointed out by Campbell *et al.* [2]. Such a facility will be especially useful for multimedia on-demand services, in which connections have to be established between the multimedia server and users' mediaphones, rather than users' terminals from which playback requests originate. At the time of connection establishment, admission control decisions will have to be made based on the requested quality of service.

The session layer, as it currently exists, supports the notion of *activities.* Information exchange within an activity is structured as a sequence of media segments called *dialogue units* in OSI terminology [7], which are delineated by major synchronization/continuity points. The continuity sublayer of the enhanced session layer that we propose maps an entire multimedia playback request on to an activity, and provides interfaces to start, end, pause, resume and stop multimedia activities. The positioning of major continuity points is specified by session service users. Actual enforcement of the major continuity points is done by the continuity sublayer of the session layer. Coordination within a dialogue unit is implemented by means of minor continuity points. These minor continuity points correspond to instants at which the media transmission rate from the source is adjusted following the reception of feedback units from the destination, so as to avert the possibility of buffer overrun or starvation at the destination. The instants at which minor continuity points are introduced are determined by the intra-media continuity protocols, based on the buffering available at the destination and the maximum variation in playback rate at the destination.

Unlike continuity techniques, synchronization techniques require cross-coordination between media streams, and cannot be implemented directly at the session layer as it currently exists (since this layer deals primarily with single connections) in the OSI reference model. It is not until the application layer, which provides the multiple association control function (MACF) [15], that any form of coordination between streams is possible. However, the application layer can only handle media information objects and not media data units, and hence, fine-grain synchronization within an object cannot be implemented at this layer. Thus, instead, we have chosen to enhance the session layer, which is the first layer at which fine-grained control such as skipping and pausing at the level of media units is possible, in order to handle media synchronization. Furthermore, since synchronization presumes continuous playback at the master mediaphone, it must be implemented on top of the continuity functions (see Figure 4).

The synchronization sublayer of the enhanced session layer will provide major and minor synchronization points. Major synchronization points indicate the temporal relationships that must exist between dialogue units of different media streams. The application specifies the temporal relationships between dialogue units, which determine the positioning of the synchronization points. Minor synchronization points are used to constantly monitor simultaneous playback of different dialogue units. Unlike major synchronization points, minor synchronization points are not specified by the session user, but may be internally introduced by the synchronization sublayer, if deemed necessary by the synchronization protocol. The synchronization protocol, which uses the synchronization technique described in Section 4, based on the user-specified asynchrony tolerance limits, determines instances at which, during the simultaneous playback of different media streams, feedback units must be received from the destinations, using which the destinations can be synchronized.

Connections to all the slave mediaphones and that to the master with which the slaves are being synchronized, constitute a group of related sessions. The synchronization sublayer exports interfaces that permit a synchronization service user to add a new connection to a group of synchronized connections, delete a connection from an existing group, change the master for the group etc. With each pair of master-slave connections is associated the tolerable lead and lag asynchrony limits, which determine instants at which minor synchronization points result. Table 2 summarizes the additional service

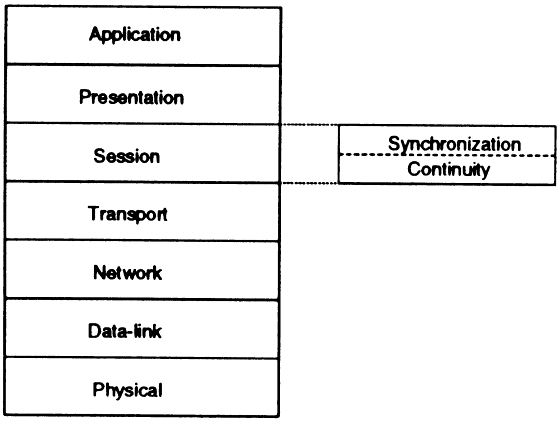

FIGURE 4. Integration of intra-media continuity and inter-media synchronization techniques into the OSI model.

TABLE 2. Service elements provided by continuity and synchronization sublayers

Service element	*Parameters*	*Purpose*
S-REMOTE-CONNECT	Initiator, Source, Mediaphone, QOSParameters	Remote connection Establishment
S-REMOTE-RELEASE	SessionID	Release the remote connection
S-PREFETCH	SessionID, MediaStream, PrefetchAmount	Prefetch media units
S-MAJOR-CONT	SessionID, TimeInterval	Introduce major continuity point
S-MINOR-CONT	SessionID	Internal element for inserting a minor continuity point
S-SYNCH-ADD	SessionID, Group, Master	Add SessionID to group of streams being synchronized with Master
S-SYNCH-DELETE	SessionID, Group	Delete SessionID from Group
S-SYNCH-GROUP	Group, Asynchrony	Request to synchronize Group, with maximum asynchrony between members being Asynchrony
S-CHANGE-MASTER	Group, NewMaster	Change master of Group to NewMaster

elements that we have introduced at the session layer to handle all of the above functions.

6. EXPERIENCE AND PERFORMANCE EVALUATION

At the UCSD Multimedia Laboratory, we are developing a prototype multimedia server, which will serve as a platform for testing the effectiveness of the techniques for continuity and synchronization. The multimedia server is being implemented on a 486-PC with multiple gigabytes of storage, and mediaphones are implemented using PC-ATs equipped with digital video and audio processing hardware, a video camera, and a TV monitor. The audio hardware digitizes audio signals at 8 Kbytes/sec. The video hardware can digitize and compress motion video at real-times rates.

We have carried out preliminary performance simulations of continuity and synchronization techniques for video (and its associated audio) playback at 15 frames/sec (which results in a playback period of 66 ms). Video frames and audio samples are transmitted on a network whose delays are assumed to be exponentially distributed between 40 and 60 ms (as is observed in our network environment). The feedback units are assumed to experience network delays bounded between 1 and 15 ms. The maximum fractional drift in their playback periods is assumed to be $\rho = 10^{-3}$.

Figure 5 illustrates the asymptotic decrease of minimum feedback ratio with buffering capacity of the videophone. At a buffer capacity of 5 frames, the minimum feedback ratio is 0.00043, yielding that the transmission of one feedback unit for every 2325 video frames played back is necessary and also sufficient for guaranteeing continuity of playback at the videophone.

Figure 6 depicts the variation of number of video frames accumulated at the videophone buffers with progression of playback, for a feedback ratio of $0.00043 = \frac{1}{2325}$. As the accumulation approaches the buffering capacity of 5 frames, a feedback unit enables the file server to sense the impending overrun, and appropriately delay transmission of future video frames; in the transmission scheme employed in this simulation, media units are transmitted at the latest possible instants.

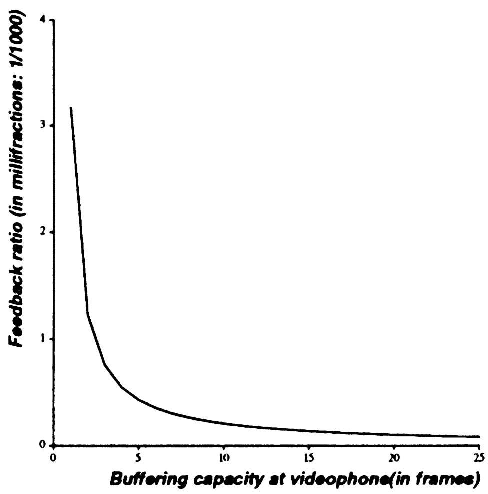

FIGURE 5. Variation of feedback ratio with buffering capacity at the videophone.

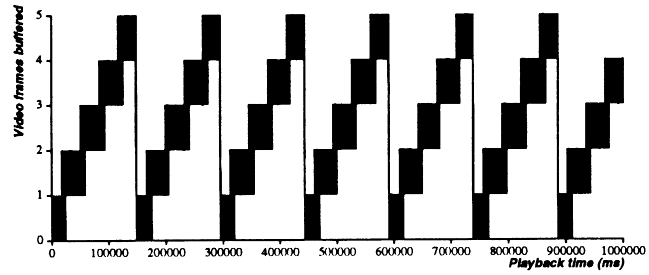

FIGURE 6. Accumulation of video frames in videophone buffers with progression of video playback at a feedback ratio of $0.00043 = \frac{1}{2325}$.

The shaded areas represent dense toggling of buffer occupancy, each toggling consisting of an upward jump due to arrival of a video frame followed by a downward fall due to display of a video frame; the right edge of each shaded area represents the instant when the videophone has accumulated sufficient additional lag to permit an extra arrival before a downward fall, causing the curve to take a step upward.

We have compared the effectiveness of the bounded buffer and multiple feedback synchronization techniques for video and audio playback, with audio as the master stream, and video as the slave. For the case when the slave lags the master, the asynchrony entailed by the bounded buffer synchronization technique steadily increases until it equals the buffering capacity, $\mathscr{B}$ of the slave ($\mathscr{B} = 5$ frames in this case), at which point the buffer at the slave is filled up. Any further arrival of video frames at the slave videophone forces it to skip video frames by deleting them from its buffers. However, since the extent of skipping is limited to one video frame at a time, the asynchrony remains consistently close to the maximum value, toggling down to $\mathscr{B} - 1$ when a frame is removed from the buffer for playback and toggling up to $\mathscr{B}$ when a new frame is received from the multimedia server (see Figure 7). The average asynchrony in this case was 4.6 video frames.

Even in the case when the slave leads the master, there are situations in which the buffer at the master fills up and remains close to full, whereby the lead of the slave toggles between $\mathscr{B}$ and $\mathscr{B} - 1$. This dependence of average asynchrony entailed in the bounded buffer technique on the buffering capacity at the mediaphones restricts the flexibility of this technique.

The average asynchrony entailed by the multiple feedback synchronization technique, on the other hand, is dependent on the feedback ratio, and on the actual rates of playback at the master and slave mediaphones, rather than on the buffering capacities of the mediaphones. Since the multimedia server detects and corrects asynchrony only when it receives feedback units from the mediaphones, the average asynchrony is inversely related to the feedback ratio; the greater the feedback ratio, the greater is the frequency of feedback transmission (and therefore, resynchronization), and hence, the smaller is the average asynchrony. The increase in average asynchrony with decrease in feedback ratio is presented in Table 3. Since the multimedia server can control the feedback ratio dynamically (by means of a binary flag in headers of media units transmitted for playback to the mediaphones), this multiple feedback synchronization technique can permit on-the-fly changes in asynchrony tolerances of subscribers. Such on-the-fly changes in asynchrony tolerances cannot be supported in the bounded buffer scheme, because the asynchrony tolerance is strictly dependent on the buffering at mediaphones.

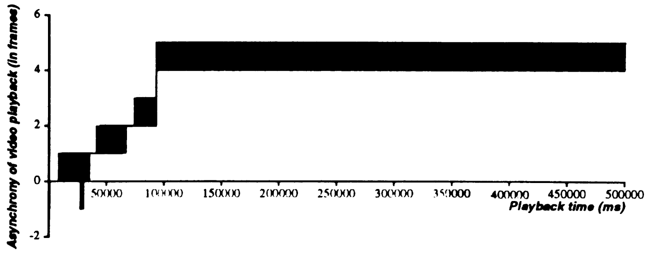

FIGURE 7. Variation of asynchrony with progression of video playback for the bounded buffer synchronization technique when the slave lags the master; Average asynchrony was found to be 4.6 video frames.

TABLE 3. Variation of average asynchrony with feedback ratio for the multiple feedback synchronization technique

Feedback ratio (ratio)	*Average asynchrony* (in video frames)
1/10	0.505
1/100	0.626
1/500	1.216
1/1000	1.689
1/2000	2.560

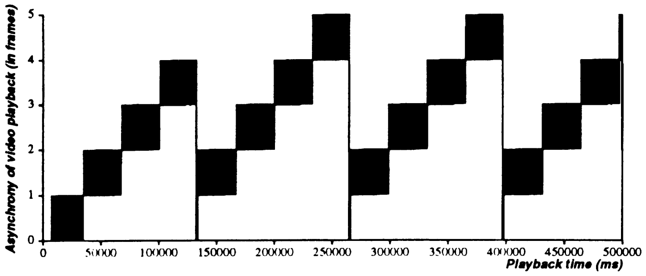

FIGURE 8. Variation of asynchrony with progression of video playback for the feedback synchronization technique when the slave lags the master; Average asynchrony was found to be 2.56 video frames.

Figure 8 illustrates the variation of the lag asynchrony of a slave relative to the master when the multiple feedback synchronization technique is employed, with the feedback ratio being $\frac{1}{2000}$, for the case when the slave plays back at the slowest rate and the master at the fastest. The average asynchrony, in this case, was 2.56 video frames, as compared to 4.6 frames when using the bounded buffer technique.

7. CONCLUDING REMARKS

It is envisaged that future advances in networking and storage technologies will make it feasible to build multimedia on-demand servers providing services similar to those of neighbourhood videotape rental stores. We have investigated the problems of providing continuous and synchronous access to such multimedia on-demand services. In order to ensure continuous retrieval of media streams from a multimedia server onto mediaphones connected directly to the high-speed network, we have proposed a feedback technique in which the multimedia server uses light-weight messages called *feedback units* generated by mediaphones and transmitted back to it to detect impending buffer overruns and starvations, and to preventively readjust media transmission so as to avoid these anomalies and their adverse effects on continuity of playback. Given buffer sizes available at mediaphones, we have presented methods to determine the minimum rate at which feedback units must be transmitted to maintain continuity.

In order to guarantee synchronous playback at mediaphones, we have presented a bounded buffer technique which uses buffering limitations at the mediaphones to automatically limit the asynchrony among the mediaphones. We also outline a multiple feedback synchronization technique in which the multimedia server uses feedback units transmitted back to it by all the mediaphones to monitor their playback and to steer the mediaphones into synchrony by speeding up a lagging mediaphone or slowing down a leading mediaphone. A performance evaluation of the feedback-based techniques reveals that although simple and easy to implement, the bounded buffering technique may entail a much greater average asynchrony than the multiple feedback synchronization technique. Furthermore, when the feedback transmission rate is reduced to the minimum required value, the overheads imposed by feedback transmission over the network become negligible. The techniques developed in this paper form the basis of a prototype multimedia on-demand information server being developed at the UCSD Multimedia Laboratory [13].

REFERENCES

[1] D. P. Anderson and G. Homsy, A continuous media I/O server and its synchronization mechanism, *IEEE Computer, Special Issue on Multimedia Information Systems*, **24**(10), pp. 51–57 (1991).

[2] A. Campbell, G. Coulson, F. Garcia and D. Hutchison, A continuous media transport and orchestration service, In *Proceedings of ACM SIGCOMM'92* (1992).

[3] M. de Prycker, *Asynchronous Transfer Mode—Solution for Broadband ISDN*, Ellis Horwood Series in Computer Communications and Networking (1992).

[4] J. Escobar, D. Deutsch and C. Partridge, A multi-service flow synchronization protocol, *BBN Systems and Technologies Division* (1991).

[5] D. Ferrari, Design and applications of a delay jitter control scheme for packet-switching internetworks, In *Proceedings of Second International Workshop on Network and Operating Systems Support for Digital Audio and Video* (Heidelberg, Germany, Springer Verlag LNCS No. 614, Editor R. G. Herrtwich) (1991).

[6] D. Ferrari and D. C. Verma, A scheme for real-time channel establishment in wide-area networks, *IEEE Journal on Selected Areas in Communications on Multimedia Communication*, **8**(3), pp. 368–379 (1990).

[7] J. Henshall and S. Shaw, *OSI explained: end-to-end computer communication standards*, John Wiley and Sons, Inc., New York (1988). Review: *Computing Reviews*, Vol. 30, No. 8.

[8] G. Karlsson and M. Vetterli, Packet video and its integration into network architecture, *IEEE Journal on Selected Areas in Communications*, **7**(5), pp. 739–751 (1989).

[9] T. D. C. Little and A. Ghafoor, Multimedia synchronization protocols for broadband integrated services, *IEEE Journal on Selected Areas in Communication*, **9**(9), pp. 1368–1482 (1991).

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

[11] K. K. Ramakrishnan and R. Jain, A binary feedback scheme for congestion avoidance in computer networks, *ACM Transactions on Computer Systems*, **8**(2), pp. 158—181 (1990).

[12] P. Venkat Rangan, S. Ramanathan, H. M. Vin and T. Kaeppner, Techniques for media synchronization in network file systems, To appear in *Computer Communications* (March 1993).

[13] P. Venkat Rangan, H. M. Vin and S. Ramanathan, Designing a multi-user multimedia on-demand service, *IEEE Communications Magazine*, **30**(7), pp. 56–65 (1992).

[14] P. Venkat Rangan, H. M. Vin and S. Ramanathan, Communication architectures and algorithms for media mixing in multimedia conferencing, To appear in *IEEE/ACM Transactions on Networking*, **1**(1) (1993).

[15] D. Shepherd and M. Salmony, Extending OSI to support synchronization required by multimedia applications, *Computer Communications*, **7**(13), pp. 399–4067 (1990).

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

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