Adaptive Feedback Techniques for Synchronized Multimedia Retrieval over Integrated Networks*

Srinivas Ramanathan and **P. Venkat Rangan**

Multimedia Laboratory
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093-0114
E-mail: {sramanat, venkat}@cs.ucsd.edu, Phone: (619) 534-5419, Fax: (619) 534-7029

Abstract

Recent advances in networking, storage and computer technologies are stimulating the development of multimedia on-demand services providing services similar to those of a neighborhood videotape rental store over metropolitan area networks. In this paper, we develop inter-media synchronization techniques for multimedia on-demand retrieval over integrated networks in the absence of global clocks. In these techniques, multimedia servers use light-weight messages called *feedback units* transmitted by media display sites (such as audiophones and videophones, generically referred to as *mediaphones*) to detect asynchronies among those sites. We present strategies by which the multimedia server, based on its most recent estimate of how soon playback may go out of synchrony at each mediaphone, can adaptively control the feedback transmission rate from that mediaphone, so as to minimize the associated overheads without permitting the asynchrony to exceed tolerable limits. We propose various resynchronization policies ranging from *conservative* (which reacts only when it is guaranteed that playback is asynchronous) to *aggressive* (which reacts as soon as there is even a slight chance that playback is asynchronous). Between the extremes of aggressive and conservative policies, *probabilistic* policies can be employed for resynchronization when statistical distributions of network delays and playback rate variations are known. We compare the performance of these policies for video/audio playback, and show that the conservative policy performs well at lower levels of asynchrony but declines in effectiveness at higher levels. In contrast, the aggressive policy exhibits oscillatory behaviour at lower levels of asynchrony, but outperforms its conservative counterpart at at higher levels. Both the policies become ineffective in high network jitter environments; only the probabilistic policies continue to perform uniformly well.

Keywords: *Media Synchronization, Multimedia On-Demand Services over Integrated Networks*

MAJOR AREA: MULTIMEDIA SERVICES, TERMINALS, AND APPLICATIONS

---

*To appear in IEEE/ACM Transactions on Networking, 1993

1 Introduction

1.1 Motivation

Technological advances in networking are making integrated networks pervasive. Coupled with the development of large capacity storage devices, these advances are making it feasible to design multimedia on-demand services catering to the educational, commercial and entertainment needs of a variety of clientele ranging from individual households to entire community neighbourhoods and organizations [9, 7]. In such multimedia on-demand services, media objects are stored at multimedia servers equipped with high capacity storage devices, and retrieved onto end-users' display sites over integrated networks [8]. Multimedia objects, in general, may be composed of multiple media streams such as audio and video, whose retrieval must proceed so as to not only maintain continuity of playback of each of the constituent media streams, but also preserve the temporal relationships among them. The design of mechanisms and protocols for providing synchronous access to multimedia services over integrated networks constitutes the subject matter of this paper.

In general, the different media streams constituting a multimedia object may be captured or played back at different sites (such as telephones, videophones, cameras, digital HDTVs, audio speakers, etc., generically referred to as *mediaphones* in the rest of this paper) on the network. For example, video may be played back at HDTV display sites and audio at CD-quality speakers, each of which may be connected directly to the network via digitizers (see Figure 1). When network delays are deterministic or constant (as in analog transmission over cable TV networks), and recording/playback rates of all the users' media capture and display sites are perfectly matched, synchronous playback is easily ensured; all that is required of a multimedia server is that it (1) instruct the mediaphones to commence playback after preset delays following the reception of the first media unit, and then (2) transmit media units to those mediaphones at their playback rate. However, in future integrated networks, factors such as congestion and queueing at network nodes are expected to introduce non-deterministic delays. Media objects may be recorded at one set of mediaphones (such as cameras belonging to video publishing and distribution houses), then stored at the multimedia server, and later played back at a different set of mediaphones (such as digital HDTVs belonging to residential consumers); and there may not be any commonality in the time of existence of connections to media recording and playback sites, rendering synchronization between clocks virtually impossible among those sites. In all such environments, additional mechanisms are essential for enforcing synchronization between media.

1.2 Relation to Previous Work

The problems in media synchronization are just being recognized. Whereas Nicolaou [5] mainly proposes mechanisms and abstractions for representation of synchronization requirements at logical and physical protocol levels, the algorithms of Anderson et al. [1] for recovering from loss of synchronization between interrupt-driven media input/output devices is mainly applicable to single-site multimedia workstations. Little et al. [4] propose a protocol for ensuring synchronous retrieval of media streams from storage servers to a single destination, in which case there is no chance of any mismatch in playback rates. Ferrari [2] proposes a jitter control scheme that can also ensure media synchronization in environments in which delays are bounded and clocks are synchronized.

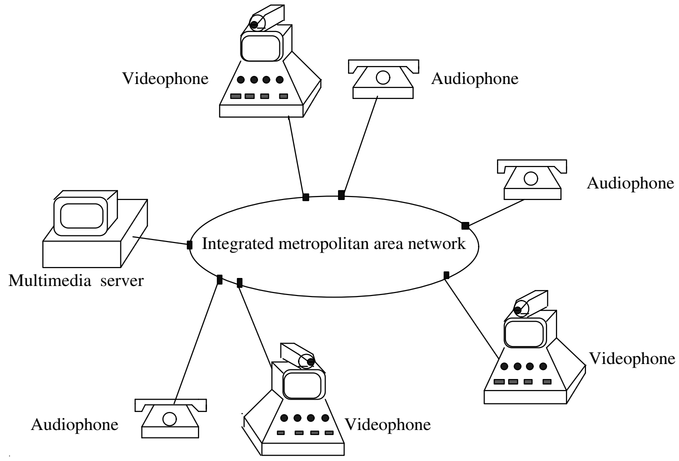

Figure 1: *Configuration of a multimedia on-demand service: a multimedia server is connected to users' mediaphones via an integrated metropolitan area network.*

1.3 Our Contributions

In this paper, we address the problem of providing synchronous access to multimedia on-demand services over integrated networks *in the absence of synchronized clocks*. We develop a inter-media synchronization technique in which, during retrieval of a multimedia object from a multimedia server to mediaphones for playback, the multimedia server uses light-weight messages (called *feedback units*) transmitted by the mediaphones to estimate the playback instants of media units at the mediaphones and detect asynchronies among them. The multimedia server then corrects the asynchrony so detected by speeding up or slowing down each mediaphone by the amount by which that mediaphone may be lagging or leading, respectively.

We propose resynchronization policies ranging from *conservative* (which reacts only when it is guaranteed that playback is asynchronous) to *aggressive* (which reacts as soon as there is even a slight chance that playback is asynchronous), that the multimedia server can use to detect and correct asynchronies between mediaphones. Between the extremes of aggressive and conservative policies, *probabilistic* policies can be employed for resynchronization when statistical distributions of network delays and playback rate variations are known. We compare the performance of the resynchronization policies for video audio playback, and show that the conservative policy performs well at lower levels of asynchrony but declines in effectiveness at higher levels. In contrast, the aggressive policy exhibits oscillatory behaviour at lower levels of asynchrony, but outperforms its conservative counterpart at at higher levels. Both the policies become ineffective in high network jitter environments; only the probabilistic policies continue to perform uniformly well.

A high feedback transmission rate, although it enables more precise and frequent estimates of playback rates at mediaphones thereby enabling asynchronies to be detected at the earliest, imposes additional overheads on the network, the multimedia server, and the mediaphones. We propose strategies by which the multimedia server, based on its most recent estimate of how soon playback may go out of synchrony at each mediaphone, can adaptively control the feedback transmission rate from that mediaphone, so as to minimize the associated overheads without permitting the asynchrony to exceed tolerable limits.

The rest of the paper is organized as follows: In Section 2, we formulate the synchronous retrieval problem, and in Section 3, we present the feedback technique for detecting asynchrony. Section 4 presents resynchronization policies, Section 5 develops adaptive feedback strategies, and Section 6 presents their extension to wide area networks. Section 7 presents a performance evaluation of the adaptive feedback techniques, and finally, Section 8 concludes the paper.

2 Formulating the Problem of Inter-Media Synchronization

2.1 System Architecture

Providing a multimedia on-demand service over an integrated metropolitan area network is a multimedia server, that stores multimedia objects on a large array of high capacity disks. Subscribers to the service can retrieve media units (such as video frames and audio samples) belonging to multimedia objects in real-time from the multimedia server over the network, and play the media units back at their mediaphones (see Figure 2). The integrated network that interconnects the server and subscribers' mediaphones is assumed to impose delays bounded between Δ_{min} and Δ_{max} for each media or feedback unit transmitted. Whereas Δ_{min} is close to the smallest propagation delay of the network, Δ_{max} must not exceed a few hundred milliseconds if the network is to support real-time, interactive multimedia applications. Bounds on network delays can be guaranteed via resource reservation at the time of start of playback of a multimedia object, and in particular, by employing admission control, real-time scheduling and buffer reservation schemes (such as those proposed by Ferrari and Verma [3]) at network nodes, at the multimedia server and at the mediaphones.

The mediaphones are simple display sites that are capable of receiving and playing back media units as well as transmitting feedback units (which are replicas of media units, except that they are devoid of data). Since the mediaphones are assumed not to possess globally synchronized clocks, there may be variations in their playback rates, with the maximum fractional drift in the playback period, θ of a media unit at any mediaphone being bounded by $\pm\rho$ (which is small enough for us to neglect higher powers of ρ and thereby approximate $\frac{1}{1-\rho}$ to $(1+\rho)$ and $\frac{1}{1+\rho}$ to $(1-\rho)$). Table 1 defines the symbols used in this paper.

2.2 Emergence of Asynchrony during Playback

During the retrieval of a multimedia object consisting of multiple media streams (such as video and audio), it is not only necessary that playback of each of the individual media streams be continuous, but it is also necessary that the playback of these media streams be mutually synchronized. Drifts in the playback rates at the mediaphones may introduce asynchrony between their playbacks; some mediaphones may playback at the fastest rate whereas some others at the slowest rate, and as a result, causing them to go out of synchrony soon after commencement. The asynchrony may accumulate as playback progresses, and the maximum asynchrony that may exist between

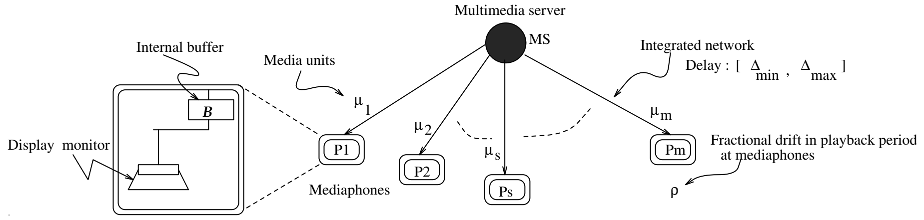

Figure 2: *Retrieval of media streams from a multimedia server to mediaphones on an integrated network*

Symbol	*Explanation*	*Unit*
$p_i(\mu)$	Playback time of media unit μ at mediaphone P_i	sec
$[p_i^e(\mu_i), p_i^l(\mu)]$	Earliest and latest playback initiation times of media unit μ at mediaphone P_i	sec
$I_i(\mu)$	Playback initiation interval of media unit μ at mediaphone P_i	sec
$\theta(\mu)$	Playback period of media unit μ	sec
f_μ	Feedback unit corresponding to media unit μ	number
θ	Nominal playback period	sec
ρ	Fractional drift in playback period at mediaphones	fraction
$\Delta(\mu)$	Network delay of media unit μ	sec
Δ_{min}	Minimum network delay of media units	sec
Δ_{max}	Maximum network delay of media units	sec
$a(f_\mu)$	Arrival time of feedback unit f_μ at the multimedia server	sec
$\mathcal{A}_{max}$	Maximum tolerable asynchrony	media units
$\mathcal{W}$	Window of precision separating media units being played back concurrently at the mediaphones	sec
β	Residual asynchrony immediately following resynchronization	media units
P_{lead}, P_{lag}	Probabilities of existence of lead and lag asynchronies	fraction
P_{thres}	Threshold probability used in the probabilistic resynchronization policy	fraction

Table 1: Symbols used in this paper

any two mediaphones is computed by the following theorem. It is assumed that the multimedia server instructs the mediaphones to commence their playback after a sufficient number of media units have been prefetched so as to maintain continuity of playback in spite of network jitter.

Theorem 1 *Although retrieval of media streams for playback at different mediaphones commences simultaneously, the maximum extent by which the slowest mediaphone may lag behind the fastest mediaphone at the instant the slowest mediaphone has played back media unit μ is given by:*

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

Proof: Let μ_m be the media unit which is being played back at the fastest mediaphone simultaneously with the playback of μ at the slowest mediaphone. Asynchrony between these two mediaphones, given by $\mu_m - \mu$, increases directly with the value of μ_m, which is maximum when both of the following conditions are satisfied:

1. the time since the start of retrieval till playback of μ is the longest possible; this happens when playback at the slowest mediaphone commences at the latest possible instant (which is given by $\tau + \Delta_{max}$, where τ denotes the instant at which the multimedia server instructs the mediaphones to commence their playback) and progresses at the slowest possible pace (i.e., with the longest period, $\theta * (1 + \rho)$);
2. playback at the fastest mediaphone commences at the earliest possible instant (which is $\tau + \Delta_{min}$), and progresses at the fastest possible pace (i.e., with the least period, $\theta * (1 - \rho)$);

In order to compute μ_m, notice that it is the last media unit whose playback may have been initiated prior to the completion of playback of μ, that is:

$$\tau + \Delta_{min} + \mu_m * \theta * (1 - \rho) \leq \tau + \Delta_{max} + \mu * \theta * (1 + \rho)$$

$$\Rightarrow \mu_m \leq \frac{(\Delta_{max} - \Delta_{min}) + \mu * \theta * (1 + \rho)}{\theta * (1 - \rho)}$$

The maximum asynchrony (in terms of media units) is given by:

$$\mathcal{A}_{max} = \mu_m - \mu \leq \frac{(\Delta_{max} - \Delta_{min}) + \mu * \theta * (1 + \rho)}{\theta * (1 - \rho)} - \mu$$

$$\Rightarrow \mathcal{A}_{max} = \left\lceil \frac{(\Delta_{max} - \Delta_{min}) + 2 * \theta * \rho * \mu}{\theta * (1 - \rho)} \right\rceil \tag{1}$$

□

The first term in Equation 1, $\frac{\Delta_{max} - \Delta_{min}}{\theta * (1 - \rho)}$ is the contribution of network jitter to the maximum asynchrony. The second term, $\frac{2 * \theta * \rho * \mu}{\theta * (1 - \rho)}$ is the contribution of playback rate mismatches, and increases linearly with progression of media playback (i.e. μ). Such a linear dependence of asynchrony on the length of a media stream is undesirable in practice, and hence, additional mechanisms are necessary for enforcing synchronization between media streams.

The multimedia server, at which the temporal relationships among media streams are stored, is best suited to handle synchronization during retrieval 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 will call as the *master*, can be spared from such discontinuities. While 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 be 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 deletions or duplications of frames in order to synchronize its playback with that of audio.

We now propose a feedback-based synchronization technique in which the multimedia server uses feedback units transmitted back to it by the mediaphones to estimate the actual playback times of media units at those mediaphones. By comparing these estimates, the multimedia server tries to detect occurrences of playback asynchronies between master and slave mediaphones. The multimedia server then tries to steer slave mediaphones back to synchrony with the master by instructing the slaves to skip or or pause media units, depending on whether they are slower or faster, respectively, relative to the master.

3 Feedback Approach for Detecting Asynchrony

Mediaphones generate feedback units concurrently with the playback of *selected* media units (but *not* necessarily with playback of every media unit), and transmit them back to the multimedia server (see Figure 3). Each feedback unit is a light-weight message containing only the number of the media unit that was concurrently played back at the time of the feedback unit's generation; hence, its transmission imposes little overhead on the network. As we will see shortly, media units corresponding to which a mediaphone is required to generate feedback units can be pre-determined by the multimedia server; hence, such media units can be distinguished by means of a binary flag set in their headers at the time of transmission by the multimedia server.

Suppose that the master mediaphone P_m transmits the first feedback unit after it has played back μ_m media units and concurrently with the playback initiation of media unit $\mu_m + 1$ (see Figure 4). The multimedia server, upon receiving feedback unit f_{μ_m+1} (corresponding to media unit $\mu_m + 1$) at time $a(f_{\mu_m+1})$, can estimate the earliest and latest possible times, $p^e_m(\mu_m + 1)$ and $p^l_m(\mu_m + 1)$ at which playback of media unit $\mu_m + 1$ could have been initiated at the master mediaphone to be (see Figure 5):

$$p^e_m(\mu_m + 1) = a(f_{\mu_m+1}) - \Delta_{max} \tag{2}$$

$$p^l_m(\mu_m + 1) = a(f_{\mu_m+1}) - \Delta_{min} \tag{3}$$

$I_m(\mu_m + 1) = [p^e_m(\mu_m + 1), p^l_m(\mu_m + 1)]$ is called a **Playback Initiation Interval** of media unit $\mu_m + 1$ at the master mediaphone. Similarly, if a slave mediaphone P_s transmits a feedback unit f_{μ_s+1} after having played back μ_s media units, the multimedia server, when it receives f_{μ_s+1}, can compute the playback initiation interval

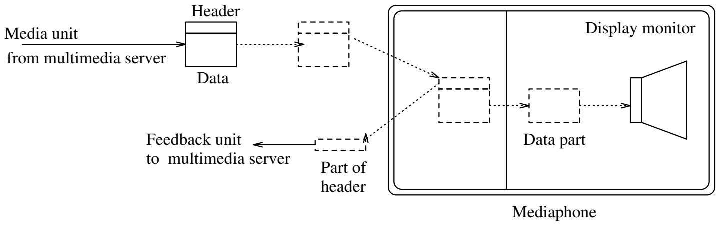

Figure 3: *Feedback transmission at a mediaphone: The mediaphone transmits part of the headers of selected media units back to the multimedia server concurrently with the transfer of data parts of these media units to the display monitor for playback.*

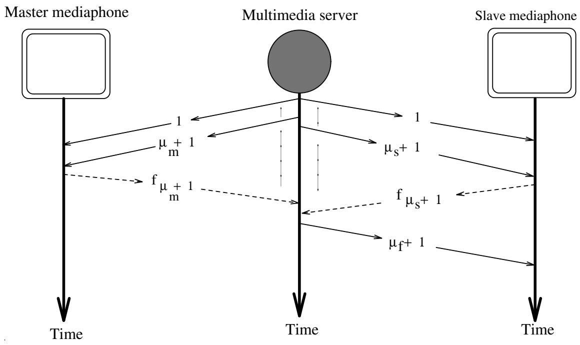

Figure 4: *Sequence of transmissions of media and feedback units between the multimedia server and master/slave mediaphones*

of $\mu_s + 1$. By comparing the playback initiation intervals of the same numbered media units at the master and the slave, the multimedia server can detect asynchronies.

The multimedia server can then resynchronize by deleting or duplicating media units (depending upon whether the slave is lagging or leading the master) from the media stream being transmitted to the slave mediaphone. The applicability of this scheme, which is simple to implement (since the mediaphones themselves need not be aware of any resynchronization), is however, restricted to local area environments with small network delays. When network delays are large, as is the case in a wide area network, a large number of media units may have been already transmitted in the intervening period between the transmission of a feedback unit (by a slave mediaphone) and its subsequent reception at the multimedia server. All of these media units would have to be played back before resynchronization by way of duplications or deletions in the transmission queue at the multimedia server becomes effective, not only causing large delays in the correction of asynchrony, but also making the delay dependent on the rate of media transmission. Since such a scheme couples resynchronization with media transmission, it also precludes the handling of resynchronization functions at network nodes other than the multimedia server[1].

Alternatively, the multimedia server can instruct slave mediaphones to skip or pause media units (depending upon whether they are lagging or leading relative to the master). In this case, the maximum latency between transmission of a feedback unit (by a slave mediaphone) and the instant at which resynchronization becomes effective is close to twice the maximum network delay, $2 * \Delta_{max}$. Thus, if a slave mediaphone transmits a feedback unit f_{μ_s+1} (at the time of playback of media unit $\mu_s + 1$), resynchronization can become effective as early as the time of playback of $\mu_f + 1 = \mu_s + 1 + \lceil \frac{2*\Delta_{max}}{\theta*(1-\rho)} \rceil$ media units.

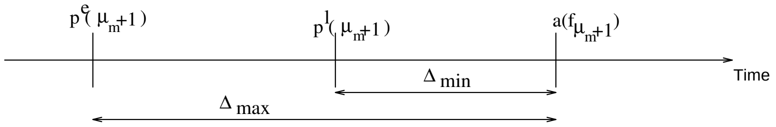

Figure 5: *Estimating earliest and latest playback times of media units*

In fact, since the maximum drift of the clocks of the mediaphones is bounded, when the multimedia server receives f_{μ_s+1}, it can, not only determine the asynchrony that existed at the time of playback of f_{μ_s+1}, but also *predict* the maximum asynchrony that is expected to exist at the time at which resynchronization will become effective, which is the time of playback of media unit $\mu_f + 1$. In order to do so, notice that, playback of successive media units at a mediaphone can be ensured to be continuous (i.e., without the mediaphone being starved of the availability of the next media unit in its buffers at the time of completion of the preceding media unit's playback) if either: (1) the network jitter ($\Delta_{max} - \Delta_{min}$) does not exceed the smallest playback period, $\theta * (1 - \rho)$, or (2)

[1]As we shall see in Section 6, the ability to handle resynchronization at other nodes is desirable for extensibility to wide area network environments.

the largest number of media units whose playback period equals network jitter are prefetched to the mediaphone's buffers before commencement of playback. This number is given by: $\frac{\Delta_{max}-\Delta_{min}}{\theta*(1-\rho)}$ (see [6] for detailed techniques and protocols for playback continuity). Since the period of each media unit is bounded between $\theta * (1-\rho)$ and $\theta * (1+\rho)$, continuity of playback enables the multimedia server to estimate the playback initiation intervals for all media units $\mu > \mu_s + 1$, starting from the playback initiation interval of media unit $\mu_s + 1$. Thus, the earliest and latest playback initiation times of the media unit $\mu_f + 1$ at the slave mediaphone can be predicted to be:

$$p_s^e(\mu_f + 1) = p_s^e(\mu_s + 1) + (\mu_f - \mu_s) * \theta * (1 - \rho) \tag{4}$$

$$p_s^l(\mu_f + 1) = p_s^l(\mu_s + 1) + (\mu_f - \mu_s) * \theta * (1 + \rho) \tag{5}$$

$[p_s^e(\mu_f + 1), p_s^l(\mu_f + 1)]$ constitutes a playback interval, $I_s(\mu_f + 1)$ of media unit $\mu_f + 1$. By applying similar techniques, given the playback initiation interval of media unit $\mu_m + 1$ (for which the master transmitted a feedback unit f_{μ_m+1}), the multimedia server estimates playback initiation intervals of master's media units $\mu_m + 2$, $\mu_m + 3$, ... By comparing these intervals with that of the slave media unit $\mu_f + 1$, the multimedia server determines media units of the master that may be played concurrently with $\mu_f + 1$, and if their unit numbers are different from $\mu_f + 1$, asynchrony (equal to their difference) is determined to exist. Based on the asynchrony so predicted, the multimedia server instructs the slave to resynchronize with the master at the time when the slave plays back media unit $\mu_f + 1$, by skipping or pausing an appropriate number of media units.

In this technique, since the multimedia server does not have a precise knowledge of the playback initiation *instants* of media units (owing to non-deterministic network delays and playback rate mismatches), it has to use estimates of playback initiation *intervals* instead. Hence, estimates of asynchronies between master and slave mediaphones, which are based on the differences between playback initiation intervals at those mediaphones, may also have a range of possibilities. Depending on whether the multimedia server takes a conservative view of considering the least possible asynchrony, or an aggressive view of considering the largest possible asynchrony, several policies are possible for resynchronization, which we explore next.

4 Resynchronization Policies

Given the playback initiation intervals of media units at master and slave mediaphones, asynchrony between them can take values ranging all the way from the minimum of their differences to the maximum. Whereas *conservative* policies resynchronize by the minimum absolute value of the difference, which is the asynchrony that is guaranteed to exist, *aggressive* policies resynchronize by the maximum absolute value of the difference, which is the largest likely asynchrony. Between these two extremes are an entire spectrum of resynchronization policies. In addition, in environments in which statistical distributions of network delays and playback rate variations are known a priori, probabilistic resynchronization policies, which resynchronize when the the probability of asynchrony exceeds a given threshold level, can be employed. We investigate each of these policies in the following subsections.

4.1 Conservative Resynchronization

In the conservative approach to resynchronization, the multimedia server, when it receives feedback units from both master and slave mediaphones, uses the estimates of their playback initiation intervals to determine media units that are *guaranteed* to be played back concurrently at the master and slave mediaphones. Here, concurrently means within a window of time $\mathcal{W}$, where $\mathcal{W}$ is the precision beyond which separations between playback instants of media units cannot be distinguished or determined, possibly due to inherent uncertainties in measurements. Mismatches in unit numbers of media units being played concurrently indicate that asynchrony is *guaranteed* to exist between the master and slave mediaphones, to alleviate which, remedial action must be initiated by the multimedia server.

In the scenario in Figure 4, any resynchronization effected by the multimedia server following the reception of feedback unit f_{μ_s+1} from the slave, is guaranteed to become effective only after media unit μ_f has been played back (i.e. at the scheduled playback initiation time of media unit $\mu_f + 1$). Therefore, in order to resynchronize the slave with the master, upon receiving f_{μ_s+1}, the multimedia server must determine the master media unit(s) that may possibly be played back concurrently with slave media unit $\mu_f + 1$. This method is developed in the following theorem.

Theorem 2 *If f_{μ_m+1} is the last feedback unit received from the master mediaphone prior to the reception of a feedback unit f_{μ_s+1} from a slave mediaphone and if μ_f is the latest media unit that could possibly be played back at that slave mediaphone before resynchronization can be effected, the range of all media units that are guaranteed to be played back at the master mediaphone concurrently with $\mu_f + 1$ at the slave is given by $[\mu_m + \delta^e, \mu_m + \delta^l]$, where:*

$$\delta^e = \left\lceil \frac{(p_s^l(\mu_s + 1) - p_m^e(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 + \rho) - \mathcal{W} + \theta * (1 - \rho)}{\theta * (1 - \rho)} \right\rceil$$

and

$$\delta^l = \left\lfloor \frac{(p_s^e(\mu_s + 1) - p_m^l(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 - \rho) + \mathcal{W} + \theta * (1 + \rho)}{\theta * (1 + \rho)} \right\rfloor$$

Proof: Media unit $\mu_m + \delta$ is guaranteed to be played back concurrently with media unit $\mu_f + 1$ if and only if the separation between the playback instants of $\mu_m + \delta$ and $\mu_f + 1$ does not exceed the window of precision $\mathcal{W}$, i.e. if and only if both of the following conditions are satisfied (see Figure 6):

$$p_m^e(\mu_m + \delta) \geq p_s^l(\mu_f + 1) - \mathcal{W} \tag{6}$$

and

$$p_m^l(\mu_m + \delta) \leq p_s^e(\mu_f + 1) + \mathcal{W} \tag{7}$$

The playback interval of $\mu_f + 1$ can be obtained from that of $\mu_s + 1$ by using Equations 4 and 5; the playback interval of $\mu_m + \delta$ can also be obtained from that of $\mu_m + 1$ in a similar manner. Substituting these in

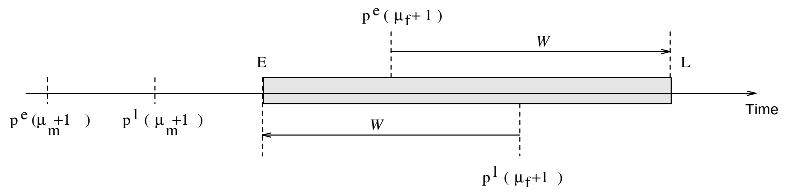

Figure 6: *Determination of a media unit* $(\mu_m + \delta)$ *which is guaranteed to be played back concurrently with* $\mu_f + 1$*; The playback interval of* $(\mu_m + \delta)$ *must be within the interval* $[E, L]$ *where* $E = p_s^l(\mu_f + 1) - \mathcal{W}$ *and* $L = p_s^e(\mu_f + 1) + \mathcal{W}$.

Equations 6 and 7, together with the observation that δ must be an integer, we obtain that, $\delta^e \leq \delta \leq \delta^l$, where:

$$\delta^e = \lceil \frac{(p_s^l(\mu_s + 1) - p_m^e(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 + \rho) - \mathcal{W} + \theta * (1 - \rho)}{\theta * (1 - \rho)} \rceil \tag{8}$$

and

$$\delta^l = \lfloor \frac{(p_s^e(\mu_s + 1) - p_m^l(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 - \rho) + \mathcal{W} + \theta * (1 + \rho)}{\theta * (1 + \rho)} \rfloor \tag{9}$$

□

Asynchrony is guaranteed to exist if $\mu_f + 1 \notin [\mu_m + \delta^e, \mu_m + \delta^l]$. The slave lags or leads the master depending on whether $\mu_f + 1 < \mu_m + \delta^e$ or $\mu_f + 1 > \mu_m + \delta^l$. In order to steer the slave back to synchrony, when the slave lags the master, the multimedia server instructs the slave to skip $(\mu_m + \delta^e) - (\mu_f + 1)$ media units from the slave's internal buffers, and when the slave leads the master, the multimedia server instructs the slave to pause for $(\mu_f + 1) - (\mu_m + \delta^l)$ media units.

Immediately following a resynchronization, there may remain a *residual asynchrony* of at most $\mathcal{W}$:

$$\beta_{conservative} = \lceil \frac{\mathcal{W}}{\theta * (1 - \rho)} \rceil \; media \; units \tag{10}$$

As we will see in Section 5, the difference between the residual asynchrony (β) and the maximum tolerable asynchrony ($\mathcal{A}_{max}$) determines the frequency at which the slave mediaphones are resynchronized; smaller the difference, greater the frequency of resynchronization.

4.2 Aggressive Resynchronization

In sharp contrast to the conservative resynchronization policy, an *aggressive* policy reacts (by skipping or pausing media units) whenever it detects even a *slight possibility* (but may not be a certainty) of asynchrony. In order to implement this policy, the multimedia server, when it receives feedback units f_{μ_m+1} and f_{μ_s+1} from the master and slave mediaphones, respectively, determines all media units of the master stream that may possibly be played back concurrently with the playback of media unit $\mu_f + 1$ at the slave mediaphone, and triggers resynchronization if it finds any mismatch between the two. This method is developed in the following theorem.

Theorem 3 *If f_{μ_m+1} is the last feedback unit received from the master mediaphone prior to the reception of a feedback unit f_{μ_s+1} from a slave mediaphone and if μ_f is the latest media unit that could possibly be played back at that slave mediaphone before resynchronization can be effected, the range of all possible media units that may be played back at the master concurrently with $\mu_f + 1$ at the slave is given by $[\mu_m + \delta^e, \mu_m + \delta^l]$, where:*

$$\delta^e = \lceil \frac{(p^e_s(\mu_s + 1) - p^l_m(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 - \rho) - \mathcal{W} + \theta * (1 + \rho)}{\theta * (1 + \rho)} \rceil$$

and

$$\delta^l = \lfloor \frac{(p^l_s(\mu_s + 1) - p^e_m(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 + \rho) + \mathcal{W} + \theta * (1 - \rho)}{\theta * (1 - \rho)} \rfloor$$

Proof: A media unit $\mu_m + \delta$ has a chance of being played back concurrently with media unit $\mu_f + 1$ if the minimum difference in their playback initiation times is at most $\mathcal{W}$. Thus, using the estimates of playback initiation intervals, the following conditions are satisfied (see Figure 7):

$$p^l_m(\mu_m + \delta) \geq p^e_s(\mu_f + 1) - \mathcal{W} \tag{11}$$

and

$$p^e_m(\mu_m + \delta) \leq p^l_s(\mu_f + 1) + \mathcal{W} \tag{12}$$

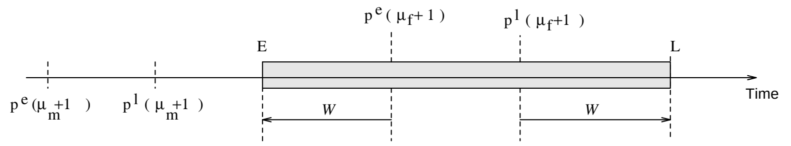

Figure 7: *Determination of a media unit $(\mu_m + \delta)$ that may be played back concurrently with $\mu_f + 1$; The playback interval of $(\mu_m + \delta)$ must be within the interval $[E, L]$, where $E = p^e_s(\mu_f + 1) - \mathcal{W}$ and $L = p^l_s(\mu_f + 1) + \mathcal{W}$*

Substituting (as was done in the case of Theorem 2) for the estimates of $p^e_s(\mu_f + 1)$, $p^l_s(\mu_f + 1)$, $p^e_m(\mu_m + \delta)$ and $p^l_m(\mu_m + \delta)$, we obtain that, $\delta^e \leq \delta \leq \delta^l$, where:

$$\delta^e = \lceil \frac{(p^e_s(\mu_s + 1) - p^l_m(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 - \rho) - \mathcal{W} + \theta * (1 + \rho)}{\theta * (1 + \rho)} \rceil \tag{13}$$

and

$$\delta^l = \lfloor \frac{(p^l_s(\mu_s + 1) - p^e_m(\mu_m + 1)) + (\mu_f - \mu_s) * \theta * (1 + \rho) + \mathcal{W} + \theta * (1 - \rho)}{\theta * (1 - \rho)} \rfloor \tag{14}$$

□

The multimedia server triggers resynchronization if there is even one media unit $\mu_m + \delta \in [\mu_m + \delta^e, \mu_m + \delta^l]$ such that $\mu_m + \delta \neq \mu_f + 1$. The difference $(\mu_m + \delta^l) - (\mu_f + 1)$ represents the highest possible lag of the slave (relative to the master), and $(\mu_f+1)-(\mu_m+\delta^e)$ represents the highest possible lead. Adopting a "high risk" policy, the multimedia server resynchronizes by the maximum of the lag and lead. Such a policy may lead the multimedia server to initiate resynchronization even when there is none (i.e., even when $\mu_f+1 \in [\mu_m+\delta^e, \mu_m+\delta^l]$) thereby accentuating (rather than alleviating) asynchrony, and represents the opposite end of the spectrum as compared to the conservative policy, which adopts a "no risk" approach by initiating resynchronization only when asynchrony is guaranteed to exist.

The residual asynchrony following a resynchronization is the maximum difference between playback intervals of $\mu_m+\delta^l$ and μ_f+1, which can be shown to be:

$$\beta_{aggressive} = \lceil \frac{2*(\Delta_{max}-\Delta_{min})+2*\theta*\rho*(\delta^l-1)+\mathcal{W}+2*\theta*\rho*(\mu_f-\mu_s)}{\theta*(1-\rho)} \rceil \; media\ units \quad (15)$$

4.3 Probabilistic Resynchronization

Between the extremes of conservative and aggressive policies, probabilistic resynchronization can be employed when the statistical distributions of both the network delays and the playback period variations are known a priori. Potentially, probabilistic policies, since they take into account the distributions of network delays and playback rate variations, can be more effective than purely deterministic ones. In probabilistic policies, the arrival of a feedback from a slave causes the multimedia server to compute the probability distribution of asynchrony at the slave. The multimedia server then resynchronizes the slave by the maximum value of asynchrony for which the probability exceeds a given threshold value. To see how this policy can be implemented, consider the scenario of Figure 4, in which the multimedia server receives a feedback unit f_{μ_s+1} from a slave mediaphone at time $a(f_{\mu_s+1})$, and μ_f is the last media unit that may be played back at the slave before resynchronization is effected. The playback initiation instant of μ_f+1 at the slave can be computed to be:

$$p_s(\mu_f+1) = a(f_{\mu_s+1}) - \Delta(f_{\mu_s+1}) + \Sigma_{i=\mu_s+1}^{\mu_f}\theta_s(i) \quad (16)$$

Similarly, if the most recent feedback unit received from the master is f_{μ_m+1}, the playback initiation instant of a media unit $\mu_f+1\pm\alpha$ at the master (α being any non-negative integer, and represents the difference relative to first media unit μ_f+1 yet to be played back at the slave) can be computed to be:

$$p_m(\mu_f+1\pm\alpha) = a(f_{\mu_m+1}) - \Delta(f_{\mu_m+1}) + \Sigma_{i=\mu_m+1}^{i=\mu_f\pm\alpha}\theta_m(i) \quad (17)$$

The probability that the slave lags the master by at least α is given by:

$$\mathcal{P}(\mathcal{A}\geq\alpha) = \mathcal{P}(p_m(\mu_f+1+\alpha)\leq p_s(\mu_f+1))$$

and that the slave leads the master by at least α is given by:

$$\mathcal{P}(\mathcal{A}\geq\alpha) = \mathcal{P}(p_m(\mu_f+1-\alpha)\geq p_s(\mu_f+1))$$

Substituting for $p_s(\mu_f+1)$ and $p_m(\mu_f+1\pm\alpha)$ from Equations 16 and 17, respectively, and defining the random variables:

$$\mathcal{V}_{lag} = \Delta(\mu_s+1) - \Delta(f_{\mu_m+1}) + \Sigma_{i=\mu_m+1}^{i=\mu_f+\alpha}\theta_m(i) - \Sigma_{i=\mu_s+1}^{\mu_f}\theta_s(i)$$

and

$$\mathcal{V}_{lead} = \Delta(\mu_s + 1) - \Delta(f_{\mu_m+1}) + \Sigma_{i=\mu_m+1}^{i=\mu_f-\alpha}\theta_m(i) - \Sigma_{i=\mu_s+1}^{\mu_f}\theta_s(i)$$

we obtain the probabilities of lag and lead to be:

$$\mathcal{P}_{lag}(\mathcal{A} \geq \alpha) = \mathcal{P}(\mathcal{V}_{lag} \leq a(f_{\mu_s+1}) - a(f_{\mu_m+1}))$$

and

$$\mathcal{P}_{lead}(\mathcal{A} \geq \alpha) = \mathcal{P}(\mathcal{V}_{lead} \geq a(f_{\mu_s+1}) - a(f_{\mu_m+1}))$$

Given the distributions of network delays and playback periods, the distributions of $\mathcal{V}_{lag}$ and $\mathcal{V}_{lead}$ can be obtained by $(2 * \mu_f \pm \alpha - \mu_m - \mu_s + 2)$ convolutions involving those distributions. However, convolutions can be computationally expensive, to bypass which, the use of normal approximation can be employed. In this approximation, network delays are assumed to be normally distributed with mean and variance λ_Δ and σ_Δ^2, respectively, and so are playback periods with mean λ_θ and variance σ_θ^2. In such a case, the random variable $\mathcal{V}_{lag}$ can be shown to be normally distributed with mean $\lambda_\mathcal{V}$ given by:

$$\lambda_\mathcal{V} = \lambda_\Delta - \lambda_\Delta + (\mu_f + \alpha - \mu_m) * \lambda_\theta - (\mu_f - \mu_s) * \lambda_\theta$$

$$\Rightarrow \lambda_\mathcal{V} = (\alpha - \mu_m + \mu_s) * \lambda_\theta$$

and variance $\sigma_\mathcal{V}^2$ given by:

$$\sigma_\mathcal{V}^2 = 2 * \sigma_\Delta^2 + (\mu_s + \alpha - \mu_m) * \sigma_\theta^2$$

The distribution of $\mathcal{V}_{lead}$ can be obtained in a similar fashion (by using $-\alpha$ in place of α).

Knowing the arrival times of f_{μ_s+1} and f_{μ_m+1} at the multimedia server, and the mean and variance of $\mathcal{V}_{lag}$ and $\mathcal{V}_{lead}$, the probability of the slave lagging or leading the master by various values of α can be determined from standard normal distribution function (available in most statistical software packages). The goal of a probabilistic resynchronization policy is, given a threshold probability, P_{thres}, to enable the multimedia server to determine the largest lag or lead between the slave and the master with a probability of at least P_{thres}. That is, the multimedia server computes α such that:

$$P_{lag}(\mathcal{A} \geq \alpha - 0.5) \geq P_{thres} > P_{lag}(\mathcal{A} \geq \alpha + 0.5) \tag{18}$$

or

$$P_{lead}(\mathcal{A} \geq \alpha - 0.5) \geq P_{thres} > P_{lead}(\mathcal{A} \geq \alpha + 0.5) \tag{19}$$

The multimedia server can then instruct the slave mediaphone to resynchronize, by skipping (in case of lag) or pausing (in case of lead) α media units. However, when no value of α satisfying Equations 18 and 19 can be determined, the multimedia server has to wait for receipt of subsequent feedback units from slave mediaphones before resynchronization. After a resynchronization, the playback at the slave is set to within half a media unit (lag or lead) of the playback at the master with a probability of P_{thres}; hence, the residual asynchrony immediately following a resynchronization is at most:

$$\beta_{prob} = 0.5\, media\ units \tag{20}$$

with a probability of P_{thres}.

5 Adaptive Feedback Techniques

The conservative, aggressive, and probabilistic policies discussed in previous sections enable a multimedia server to resynchronize a mediaphone (that may have fallen into asynchrony) as soon as a feedback unit is received from that mediaphone. Immediately following a resynchronization, each mediaphone may have a residual asynchrony that is at most β (given by Equations 10, 15, and 20). Starting from this residual value, asynchrony of a mediaphone may again increase with progression of playback, owing to playback rate variations. Such an increase may be allowed upto a maximum tolerable limit, $\mathcal{A}_{max}$, by when the multimedia server must have received the next feedback unit from the mediaphone and initiated the next resynchronization, bringing the asynchrony back to within β. In order to minimize additional overheads due to feedback transmission, it is desirable to maximize the interval between the previous and the next resynchronizations (and, hence, the interval between transmission of their corresponding feedback units); the maximum allowable interval is directly determined by the extent by which the asynchrony can be allowed to increase, that is, by the difference between the maximum tolerable asynchrony ($\mathcal{A}_{max}$) and the residual asynchrony (β) at the end of previous resynchronization.

Whereas the maximum tolerable asynchrony, $\mathcal{A}_{max}$ is specified by the application under consideration, the residual asynchrony, β at the end of a resynchronization depends on the network delay bounds, the playback period of media units, and the resynchronization policy being employed. Furthermore, for the conservative policy, β depends on the relative separation of feedback units (from master and slave mediaphones) corresponding to the previous resynchronization. Thus, at the time of each resynchronization, the multimedia server can adaptively set the maximum interval that can elapse before the transmission of next feedback by a mediaphone. The exact method for computation of such maximum intervals between transmission of successive feedback units is presented next.

5.1 Adaptive Determination of Maximum Feedback Intervals

Consider the situation when the i^{th} feedback unit from a slave reaches the multimedia server. The multimedia server initiates the necessary resynchronization causing the slave to playback media unit ν_s^i. Suppose that following the i^{th} resynchronization, μ_s^i media units are played back by the slave before it transmits the $(i+1)^{th}$ feedback unit (see Figure 8). The largest value of μ_s^i, which denotes the interval between transmission of i^{th} and $(i+1)^{th}$ feedbacks by the slave, without permitting the asynchrony of the slave to increase beyond the maximum tolerable value, $\mathcal{A}_{max}$ (generally specified by the application) is precisely computed in Theorem 4 for the case when the slave lags the master; the case when the slave leads the master is similar.

Theorem 4 *The maximum duration of the interval between the i^{th} resynchronization and the transmission of the $(i+1)^{th}$ feedback unit from a slave mediaphone so as to bound the lag of the slave mediaphone relative to master to within tolerable limits is given by:*

$$\mu_s^i = \left\lfloor \frac{\mathcal{A}_{max} * \theta * (1-\rho) - \beta^i * \theta * (1+\rho) - 4 * \rho * \Delta_{max} * (1+\rho)}{2 * \theta * \rho} \right\rfloor$$

where $\mathcal{A}_{max}$ denotes the maximum tolerable asynchrony and β^i denotes the maximum residual asynchrony by which the slave may lag the master immediately following the i^{th} resynchronization.

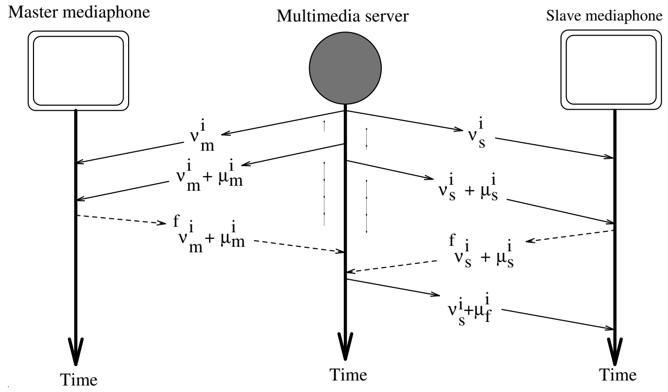

Figure 8: *Sequence of exchange of media and feedback units between the multimedia server and mediaphones*

Proof: Consider the accumulation of the lag of a slave mediaphone relative to the master mediaphone between the i^{th} and $(i+1)^{th}$ resynchronizations. Starting from a residual value of β^i immediately following the i^{th} resynchronization (the value of β^i being dependent on the resynchronization policy), the lag accumulates most rapidly when playback at the slave progresses at the slowest possible pace, but that at the master progresses at the fastest possible pace. Whereas the slave sends the $(i+1)^{th}$ feedback immediately after playing back $\nu_s^i + \mu_s^i$, $(i+1)^{th}$ resynchronization takes place at the time of playback of $\nu_s^i + \mu_f^i$, with the relation between μ_f^i and μ_s^i being given by

$$\mu_f^i = \mu_s^i + \lceil \frac{2 * \Delta_{max}}{\theta * (1-\rho)} \rceil \tag{21}$$

Following a procedure similar to that adopted in Theorem 1, it can be shown that the maximum lag of the slave just prior to the playback of $\nu_s^i + \mu_f^i$ is given by:

$$\mathcal{A}_{max} = \lceil \frac{\beta^i * \theta * (1-\rho) + 2 * \theta * \rho * \mu_f^i}{\theta * (1-\rho)} \rceil \tag{22}$$

Substituting for μ_f^i in terms of μ_s^i (using Equation 21), and solving for μ_s^i, we obtain:

$$\mu_s^i \le \lfloor \frac{(\mathcal{A}_{max} - \beta^i) * \theta * (1-\rho) - 4 * \rho * \Delta_{max} * (1+\rho)}{2 * \theta * \rho} \rfloor \tag{23}$$

which represents the maximum interval (in terms of media units) between two successive feedbacks from a slave.

Using the above equation, at the time of the i^{th} resynchronization (for any integer of $i \ge 1$), the multimedia server can determine the the next media unit $\nu_s^i + \mu_s^i$ corresponding to which a mediaphone must send the $(i+1)^{th}$ feedback unit, so as to never let the asynchrony exceed the maximum permissible value of $\mathcal{A}_{max}$ before $(i+1)^{th}$ resynchronization takes effect. When $i = 0$, Equation 23 can be directly used to determine the instant at which the first feedback unit has to be transmitted back to the multimedia server, with the initial residual asynchrony (between playback start times at the master and the slave), $\beta^0 * \theta * (1 - \rho)$ being equal to the network delay-jitter, $\Delta_{max} - \Delta_{min}$.

□

The above scheme is said to be *adaptive* since, the largest permissible interval, μ_s^i between the i^{th} and the $(i + 1)^{th}$ resynchronizations is dynamically determined by the multimedia server based on the residual asynchrony β^i that existed immediately following the i^{th} resynchronization. Having determined the interval, μ_s^i, the multimedia server can then insert a binary flag in the header of such a media unit $\nu_s^i + \mu_s^i$ to indicate to the mediaphone that a feedback unit must be transmitted corresponding to that media unit.

It should, however, be observed that in the computation of μ_s^i using Equation 23, all the factors except β^i can be precomputed before the commencement of playback. Even for the computation of β^i for the conservative and aggressive policies, all factors except the difference between playback time estimates of the feedback units from the master and slave can be precomputed before the commencement of playback; and during playback, computation of the playback time estimates upon reception of a feedback involve about four additions/subtractions, and the computation of β^i would involve about two more additions/subtractions and two multiplications/divisions. Similar is the case for determination of asynchronies using either of the sets of Equations (8, 9) or (13, 14). For the probabilistic policy, the computation of β^i involves one-time cost of a couple of additions and multiplications/divisions, and the computation of asynchrony (Equations 18 and 19) involves a couple of function calls to standard statistical packages, and a few additions/multiplications of mean and variances of normal distributions. Hence, all in all, both the one-time computational overheads associated with precomputed factors and the per-feedback unit computational overheads associated with dynamic factors together sum up to a few arithmetic operations, for each slave mediaphone being synchronized. This level of overhead is even smaller than those associated with checksumming and other routine per network-message operations in common systems.

To minimize the frequency of feedback transmission, it is desirable to maximize μ_s^i, which, in turn, is possible if the residual asynchrony, β^i is minimized. However, from Equation 15, β^i for the case of aggressive policy increases directly with the separation between arrival times of feedback units from master and slave mediaphones, which must, hence, be minimized. For the conservative policy, the effect of a large separation is even worse: there is a chance that this policy will never be able to resynchronize in such a case. For the probabilistic policy, larger the separation, larger is the number of convolutions and the variance, and less sharp is the probability distribution, smaller is the likelihood of finding a solution to Equations 18 and 19. In order to minimize the separation between arrival of master and slave feedbacks and thereby overcome the above drawbacks, we now propose a feedback transmission scheme in which the master mediaphone transmits feedback units in short bursts around the time when feedback units from slave mediaphones are expected.

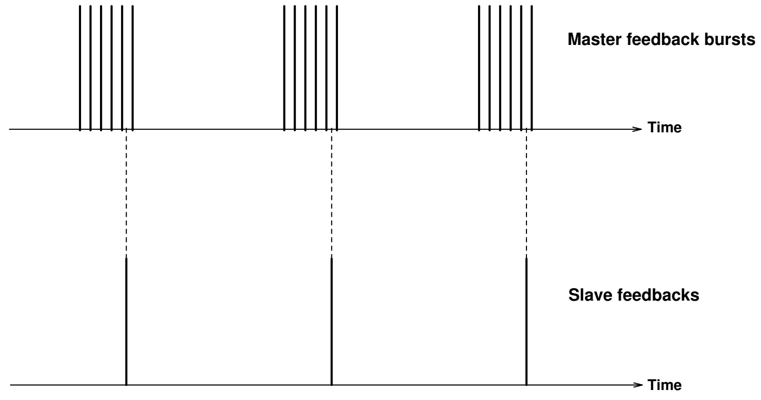

Figure 9: *Bursty Feedback Transmission scheme: master transmits feedbacks in bursts, while slave transmits at minimum rate*

5.2 Bursty Feedback Transmission Scheme

In this scheme, whereas slave mediaphones transmit feedback units at intervals given by Equation 23, the master mediaphone transmits feedback units in short bursts (see Figure 9). The commencement and termination of each feedback burst from the master mediaphone, which can be completely controlled by the multimedia server through the setting of a binary flag in the headers of media units corresponding to which the feedbacks are to be transmitted, are chosen such that the start of the burst coincides with the earliest possible arrival of a feedback from any of the slave mediaphones, and the end of the burst coincides with the latest possible arrival. This way, the maximum separation between arrival of feedbacks from a master and slave is bounded by the separation between two successive feedbacks within a burst from the master, and consequently, the residual asynchrony following a resynchronization also becomes bounded.

In order to precisely compute the length of the burst needed to so bound the residual asynchrony, suppose that, after the i^{th} resynchronization, μ_s^i media units are played back by a slave mediaphone prior to the transmission of the next feedback unit. The smallest interval between the time when the i^{th} resynchronization is effected at the slave mediaphone and the time when the $(i+1)^{th}$ feedback unit from the slave is received by the multimedia server is $\mu_s^i * \theta * (1-\rho) + \Delta_{min}$. In comparison, the master's playback may have a residual lag of at most β^i relative to the slave immediately following the i^{th} resynchronization and thereafter progress at the slowest possible rate, because of which, the $(i+1)^{th}$ feedback burst (which starts after an interval of μ_m^i media units) could arrive as late as $(\beta^i + \mu_m^i) * \theta * (1+\rho) + \Delta_{max}$ after the i^{th} resynchronization. In order to guarantee that the arrival of the $(i+1)^{th}$ burst from the master precedes the arrival of the $(i+1)^{th}$ feedback unit from the slave, it must be the case that:

$$(\beta^i + \mu_m^i) * \theta * (1+\rho) + \Delta_{max} \leq \mu_s^i * \theta * (1-\rho) + \Delta_{min}$$

$$\Rightarrow \mu_m^i \leq \frac{\mu_s^i * \theta * (1-\rho) - (\Delta_{max} - \Delta_{min})}{\theta * (1+\rho)} - \beta^i \tag{24}$$

The largest value of μ_m^i that satisfies the above equation determines the start of the $(i+1)^{th}$ feedback burst from the master, after which for each successive media unit played back at the master, the multimedia server sets the binary feedback flag to trigger transmission of a feedback unit from the master, until the $(i+1)^{th}$ feedback unit is received from *each* slave, at which time the burst from the master is ended.

Feedback units, since they contain only sequence numbers of media units that were played back concurrently with their transmission, are light-weight. Furthermore, since playback of not all but only selected media units triggers feedback transmission, the network overhead entailed by the adaptive feedback technique is small compared to that of media transmission[2]. Given that the fractional variations in playback periods are generally very small, for instance, the clock drift among mediaphones may be of the order of a few seconds per day, the length of a burst is usually small, and the interval between two feedbacks of a slave or feedback bursts of the master is large. As we will see in Section 7, even when network delays are of the order of hundreds of milliseconds, for typical media playback rates, the feedback burst lengths are only of the order of a few tens of media units over an interval of about a few thousands of units, indicating that overheads due to their transmission at mediaphones and reception at the multimedia server are small.

6 Extension to Wide Area Networks

In networks that are geographically distributed over a wide area, delays between the multimedia server and mediaphones can be large, thereby increasing both the latency ($\mu_f - \mu_s$) between detection and correction of asynchrony, and the residual asynchrony (β). Consequently, the interval between two successive feedbacks decreases, causing the number of feedback receptions and resynchronization computations at a multimedia server to increases with geographical distribution. Thus, additional mechanisms are essential for keeping the overheads due to the feedback technique small.

Notice that, since in our feedback technique, resynchronization is effected by skips and pauses at mediaphones rather than by deletions and duplications in media unit transmissions at the multimedia server, all the computations and actions associated with each resynchronization can be carried out by any node other than the multimedia server. Furthermore, resynchronizations associated with different slave mediaphones can be handled by different nodes serving as *synchronizers*. The slave mediaphones can thus be partitioned into disjoint subsets, and for each subset, a node close to the subset can serve as the synchronizer. Whereas the slave mediaphones transmit feedbacks only to the synchronizer associated with the subset to which they belong, the master mediaphone multicasts its feedbacks to all the synchronizers. By so distributing the function among a number of synchronizers, the overhead due to the feedback technique at each of the synchronizers can be kept small.

[2]In order to avoid back-to-back reception of feedback units and the consequent possibilities of their losses at the multimedia server's network interface, feedback transmissions from different mediaphones can be slightly staggered in time.

In some situations, the master and slave mediaphones themselves may be widely separated from each other. In such cases, a hierarchical architecture in which mediaphones closer to each other are better synchronized than those separated farther apart may be desirable. Nodes serving as synchronizers are configured in a *hierarchy* with the mediaphones as leaf nodes (see Figure 10). At each internal node, within the subtree rooted at that node, one of the mediaphones functions as the *subtree master*, relative to which all other mediaphones in the subtree are synchronized by the node. Each node transmits only the feedback units of its subtree master to its parent node. The master at the root of the tree serves as the *global master*. Whereas the global master always plays at its natural rate, subtree masters may have to pause or skip relative to the subtree master at their higher level. When the global master slows down or speeds up, resynchronization actions propagate down the tree.

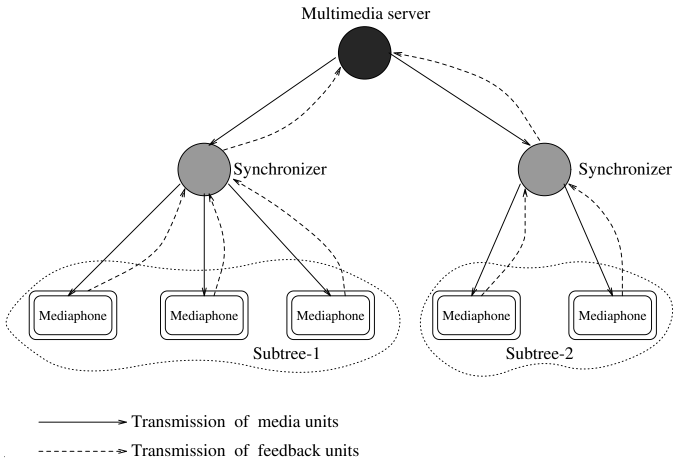

Figure 10: *Distribution of feedback synchronization function among multiple nodes in a wide area network*

The above decentralization of the feedback technique eliminates the need for the multimedia server (which is the root of the tree) to receive feedback units all the way back from each leaf mediaphone and resynchronize them all. The computational and feedback transmission overheads of resynchronization are distributed among all the intermediate nodes, which can as well be routing nodes, thereby fully integrating synchronization with routing.

7 Experience and Performance Evaluation

In order to experimentally evaluate the performance of the adaptive feedback techniques for synchronous multimedia retrieval, we are developing a prototype multimedia on-demand server at the UCSD Multimedia Laboratory (see Figure 11). The multimedia server is being implemented on a 486-PC with multiple gigabytes of storage, and each display device consists of a PC-AT 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-time rates.

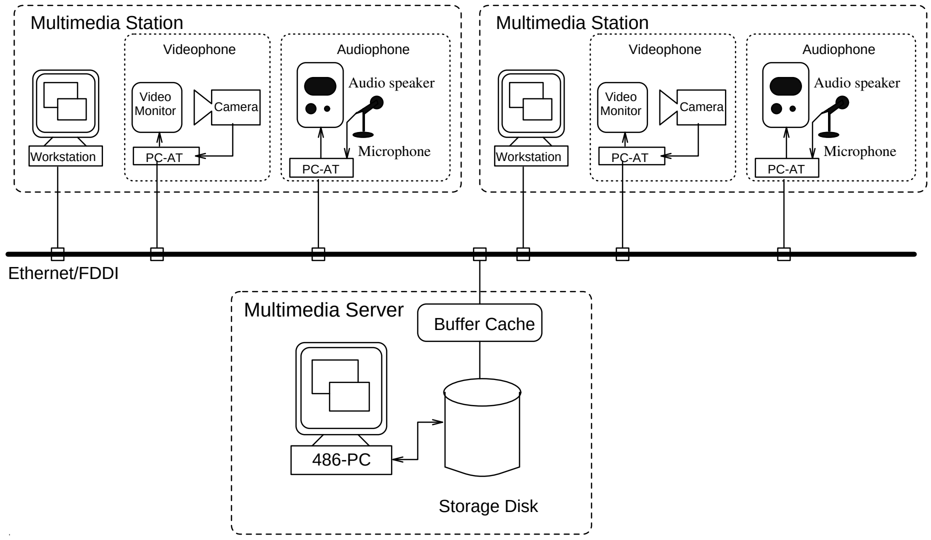

Figure 11: *Hardware configuration of the multimedia network*

We have carried out preliminary performance simulations of the resynchronization policies for video (and its associated audio) playback at 60 frames/sec (which results in a nominal playback period of 16.67 ms), with the audiophone serving as the master and the videophone as the slave. In the simulations, 100,000 media units are assumed to be transmitted by the multimedia server to the mediaphones, first in a local and metropolitan area network environment, and then in a wide area network environment. The network delays are approximated to be normally distributed, with 99.99% of the delays lying in the range $[\Delta_{min}, \Delta_{max}]$, whose values are assumed to be $[40ms, 50ms]$ for the local/metropolitan area network environment, and $[100ms, 200ms]$ for the wide area network environment (these are the transmission plus queueing delay ranges observed for a video frame in our 10 Mb/s Ethernet and 2.4 Mb/s wide area network environments). The maximum fractional drift in the playback periods at both the videophone and the audiophone is assumed to be: $\rho = 10^{-3}$. A maximum tolerable asynchrony of $A_{max} = 5\ media\ units$ (83.33 ms) was used throughout.

The deterministic resynchronization policies, namely, the conservative policy and the aggressive policy, are effective only when the network jitter is sufficiently small such that, in comparison with the network jitter, the maximum tolerable asynchrony is large enough to clearly stand out. In our simulation experiments this is indeed the case for the local/metropolitan area network environment. For such an environment, we compare the effectiveness of conservative and aggressive policies for two scenarios that give an idea of the range of the performance spectrum of the two policies: (1) an ideal scenario in which the playback rates of the master and slave mediaphones are identical, and (2) an extreme scenario in which, whereas the master mediaphone plays back at the fastest rate, the slave mediaphone plays back at the slowest rate. Both conservative and aggressive policies fail when network jitter becomes comparable to tolerable asynchrony. However, the probabilistic policy, since it uses statistical distributions of network delays and playback periods, continues to be effective in resynchronizing the mediaphones even in those cases.

Several measures of effectiveness of resynchronization are reported by the simulations for the various policies: (1) the maximum, minimum, and average absolute asynchronies, (2) the number of skips and pauses incurred at the slave, and (3) the *misfire ratio*, which is the number of incorrect resynchronization decisions (which actually accentuate the asynchrony instead of alleviating it) as a fraction of the total number of resynchronization decisions taken. Tables 2-4 and Figure 12 summarize these results.

In the ideal scenario, the conservative policy does not trigger any resynchronization (see Table 2). This is understandable, because, the conservative policy triggers resynchronization only if asynchrony is guaranteed to exist. In sharp contrast, the high-risk aggressive policy overreacts, leading to a high average asynchrony. This is because, even though there are no playback rate mismatches, the network jitter by itself results in differential delays between master and slave feedbacks, thereby falsely causing the aggressive policy to trigger resynchronization, as is evident in its large misfire ratio. In fact, the skips and pauses introduced by the aggressive policy are large and almost equal in number, demonstrating its oscillatory behaviour (see Figure 12); this oscillatory behaviour is more evident for larger feedback frequencies, i.e. for smaller values of $\mathcal{A}_{max}$.

	Conservative policy	Aggressive policy
Maximum asynchrony (ms)	2.95	2.95
Minimum asynchrony (ms)	2.95	-13.88
Average absolute asynchrony (ms)	2.95	7.8
Media units skipped	0	25
Media units paused	0	24
Misfire ratio	0	0.51
# Master feedbacks	406	419
# Slave feedbacks	45	49

Table 2: Performance of the conservative and aggressive policies in the ideal scenario

In the extreme scenario, the conservative policy entails larger asynchronies than its aggressive counterpart (see Table 3). This is because, since the conservative policy reacts only when it can conclude with certainty the existence of asynchrony, it waits until the asynchrony builds up to surpass the jitter so as to clearly stand out in comparison. The aggressive policy, however, because of its fast reactivity, has the potential to detect asynchrony at the earliest instant and then resynchronize by the maximum possible extent, yielding a much lower asynchrony. Thus the aggressive policy outperforms the conservative policy in this case, showing that it is not without advantages.

Both aggressive and conservative policies lose their effectiveness when the maximum network jitter becomes comparable to, and hence, difficult to distinguish from, the maximum tolerable asynchrony. Whereas the aggressive policy tends to react too frequently (because of false resynchronizations triggered by the large jitter), the

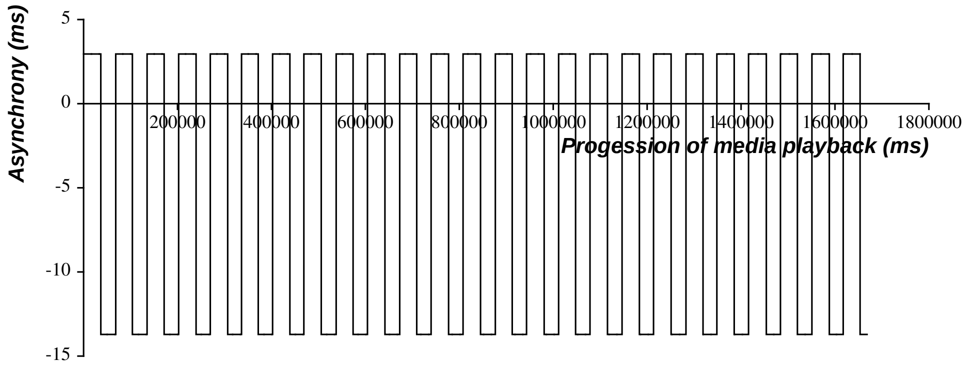

Figure 12: Oscillatory behavior of the aggressive policy in the ideal scenario

	Conservative policy	Aggressive policy
Maximum asynchrony (ms)	92.33	76.22
Minimum asynchrony (ms)	-7.43	-23.55
Average absolute asynchrony (ms)	41.50	20.82
Media units skipped	199	201
Media units paused	0	0
Misfire ratio	0	0
# Master feedbacks	603	662
# Slave feedbacks	43	55

Table 3: Performance of the conservative and aggressive policies in the extreme scenario

conservative policy, being unable to assuredly separate the effects of asynchrony from those of jitter, tends not to react at all. In such cases, the probabilistic policy, since it uses statistical distributions of network delays and playback periods, succeeds in detecting asynchrony and carrying out resynchronization with a high probability (see Table 4), thereby avoiding both (1) the over-reactivity and oscillatory behavior of the aggressive policy, and (2) the sluggishness of the conservative policy.

	Ideal scenario	Extreme scenario
Maximum asynchrony (ms)	29.52	130.60
Minimum asynchrony (ms)	-3.80	-24.10
Average absolute asynchrony (ms)	6.75	57.95
Media units skipped	2	197
Media units paused	0	0
Misfire ratio	0.02	0
# of master feedbacks	928	1162
# of slave feedbacks	46	45

Table 4: Performance of the probabilistic resynchronization policy (90% threshold) when the network jitter is high, as in a wide area environment with $[\Delta_{min}, \Delta_{max}] = [100ms, 200ms]$. The conservative and aggressive policies become ineffective in this case, and are hence, not shown in the table; whereas the conservative policy never reacts, the aggressive policy yields a feedback ratio of one for each media unit and goes into a highly oscillatory mode.

From the measurements, it may also be observed that the maximum number of feedbacks range around a thousand per 100,000 media units. The burst lengths (which can be computed as the ratio of the number of master feedbacks to the number of slave feedbacks in Tables 2-4) are in the range of a few tens of media units, with each burst occurring at most once in a thousand media units. Thus, the total number of feedback units as a fraction of the total number of media units is about 0.01, confirming that the overheads due to feedback transmission are likely to be very small.

8 Concluding Remarks

We have developed adaptive, feedback techniques for synchronous multimedia retrieval from multimedia on-demand servers to mediaphones in future integrated networks, in the presence of network delay jitter and non-deterministic playback rate mismatches. Resynchronization policies employed by the multimedia server can range from being conservative so as to react only when it is guaranteed that playback is asynchronous, to being aggressive so as to react as soon as there is even a slight chance that playback is asynchronous. Between these two extremes, probabilistic policies can be employed when statistical distributions of network delays and playback rate variations are known. A performance evaluation of these policies reveals that whereas conservative policies perform best at lower levels of asynchrony, aggressive policies perform best at higher levels of asynchrony. Both conservative and aggressive policies decline in effectiveness in environments with high network jitter; only the probabilistic policies continue to perform uniformly well, by avoiding both the over-reactivity of the aggressive policy and the sluggishness of the conservative policy. We present strategies by which the multimedia server can adaptively control the feedback transmission rate from a mediaphone. In order to further minimize asynchronies without increasing the rate of feedback transmission, we propose short, bursty transmission of feedback units. Initial simulation of these techniques for video audio playback yields an average feedback rate of one feedback unit for about 1000 media units to keep the asynchrony within about 80 ms, showing that the overhead due to feedback transmissions entailed by these techniques is very small.

REFERENCES

[1] D. P. Anderson and George Homsy. A Continuous Media I/O Server and Its Synchronization Mechanism. *IEEE Computer, Special Issue on Multimedia Information Systems*, 24(10):51–57, October 1991.

[2] 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.

[3] 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):368–379, April 1990.

[4] T.D.C. Little and A. Ghafoor. Multimedia Synchronization Protocols for Broadband Integrated Services. *IEEE Journal on Selected Areas in Communication*, 9(9):1368–1482, December 1991.

[5] Cosmos Nicolaou. An Architecture for Real-Time Multimedia Communication System. *IEEE Journal on Selected Areas in Communication*, 8(3):391–400, April 1990.

[6] P. Venkat Rangan and Srinivas Ramanathan. Rate-Based Feedback Techniques for Continuity and Synchronization in Multimedia Retrieval over High-Speed Networks. *Technical Report No. CS92-230, Dept. of Computer Science and Engineering, University of California, San Diego*, February 1992.

[7] P. Venkat Rangan and D. C. Swinehart. Software Architecture for Integration of Video Services in the Etherphone Environment. *IEEE Journal on Selected Areas in Communication*, 9(9):1395–1404, December 1991.

[8] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan. Designing an On-Demand Multimedia Service. *IEEE Communications Magazine*, 30(7):56–65, July 1992.

[9] W. D. Sincoskie. System Architecture for a Large Scale Video on Demand Service. *Computer Networks and ISDN Systems, North-Holland*, 22:155–162, 1991.