The Use of Network Delay Estimation for Multimedia Data Retrieval [1]

J.F. Gibbon[2] and T.D.C. Little

Multimedia Communications Laboratory
Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
jgibbon@esscom.com, tdcl@bu.edu

MCL Technical Report 06-15-1996

Abstract–Multimedia data have specific temporal presentation requirements. For example in video conferencing applications voice and images of participants must be delivered and presented synchronously. These requirements can be achieved by scheduling or managing system resources.

We present a technique called limited *a priori* scheduling (LAP) to manage the delivery channel from source to destination for digital multimedia data. By using delay estimation a LAP scheduler can retrieve stored digital media spanning arbitrary networks with unspecified delays. The use of delay estimation also facilitates selective degradation of service in bandwidth and buffer limited situations. Such degradation enables the continuous real-time playout and synchronization of various media arriving from different sources. The performance of the LAP scheduler is described based on implementation and experimentation using Ethernet.

Keywords: Digital media, multimedia, network delay modeling, flow control, bandwidth allocation, resource management, real-time networks, real-time scheduling.

[1]in *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, September 1996, pp. 1376-1387. This work is supported in part by the National Science Foundation under Grant No. IRI-9211165.

[2]Currently with Essential Communications of Albuquerque, NM.

1 Introduction

Data objects involved in multimedia presentations can have strict temporal playout requirements. We define a multimedia data object as data such as graphics, text, video, audio, or similar items. Each object has a specific relationship with the other objects in an author's presentation. For example, a multimedia slide presentation requires visual and aural components to be played out for a predefined sequence and duration (Fig. 1).

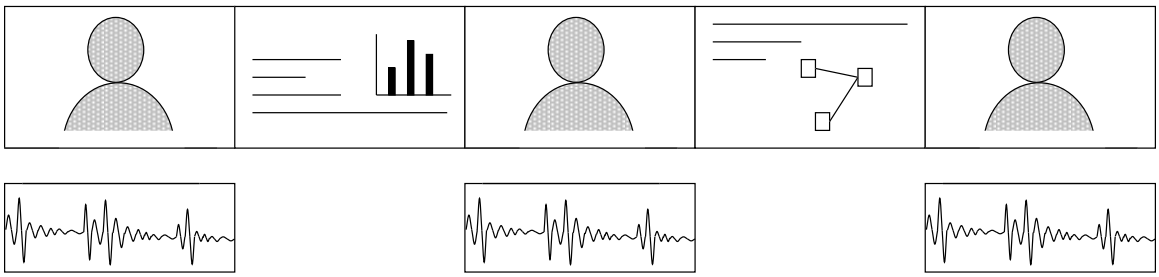

Figure 1: An Example of Multimedia Data Presentation

If these temporal relationships are violated, the result can be noticeable to the end-user. For example, in "talking head" applications such as videoconferencing, a skew in audio and video playout greater than 160 ms is easily detectable and is annoying [31]. The relationships between components of the same object or medium are equally important. For example the constant frequency of audio samples of an audio object must be preserved during playout.

Given a temporal specification for the playout of a collection of multimedia objects, the correct sequencing and timing of playout can be achieved through the support of a computer operating system and associated interconnected computer network. Therefore, these subsystems must be managed and scheduled to yield the timely delivery of the multimedia objects.

One of the most difficult subsystems to schedule is the communication channel. This is due to the high data rate of a multimedia presentation relative to most systems' performance and to the shared nature of computer networks. A multimedia scheduler must also respond to dynamic user requests caused by user interaction and must deal with limitations in available memory. In summary, a multimedia scheduling system must accommodate the temporal requirements of a multimedia presentation, adapt to dynamic user input, consider memory limitations, and respond to changes in network utilization.

The approach described in this paper, called limited *a priori* scheduling (LAP), conforms to these requirements. The essence of the technique is as follows: After receiving information about the characteristics of the stored data from multiple sources, the LAP scheduler, residing at the receiver, creates a data retrieval schedule based on delay estimation and using real-time scheduling. The LAP scheduler then sends a unique data retrieval schedule to each source. Each retrieval schedule; however, is only valid for a limited period due to the dynamic behavior of the network loading and the user's interaction with the system. The LAP scheduler creates a schedule for a new period when the current schedule expires or when required by dynamic user input or changes in network performance. Bandwidth and memory limitations are also considered by the LAP scheduler.

There are many approaches to accommodating the delivery requirements of time dependent data such as digital media. Real-time network communications as described by Ferrari and Verma [13] and Lazar et al. [22] provide performance guarantees which can be either absolute, through deterministic scheduling and resource allocation, or approximate by using statistical approaches. In these types of networks, the user can request specific time-based communication requirements. If adequate resources exist to accommodate these requirements, the network grants a connection. A similar approach is used for RSVP (Resource ReSerVation Protocol) [5]. The RSVP reservation model uses flow and filter specifications (flows spec and filter spec). The flow spec defines the requested quality of service (QOS) while the filter spec associates data packets in a session to a given flow spec.

Unlike research in real-time network communications, there have been systems designed based on pre-existing network protocols to accommodate the requirements of multimedia data delivery. One popular network technology for digital media delivery is the Asynchronous Transfer Mode due in part to its bandwidth allocation policies [8, 20, 27, 28]. Another is the IEEE 802.5 Token Ring LAN which is has a maximum token holding time and a data priority system that allow for bounded transmission delays [3]. Studies have also been performed on other ring networks such as the Cambridge Fast Ring [1] and the Fiber Data Distributed Interface network [23].

One such system, using the Capacity Based Session Reservation Protocol (CBSRP), maintains temporal data relationships of multimedia presentations by using the characteristics of an FDDI network and a real-time operating system called ARTS [32, 7]. Similar to the LAP scheduler, resource scheduling and management is performed to ensure that the proper resources are available for data playout. Unlike the LAP scheduling technique, it uses a real-time operating system and a specific communications technology to manage all active

sessions.

There has also been considerable research on the delivery of multimedia data through general packet-switched networks. Elliot et al. [10] constructed a global videoconferencing environment based on a packet-switched network with bandwidth reservation and multicasting protocols. Other systems use flow control techniques rather than built-in bandwidth reservation protocols [2, 11, 17, 19, 21]. These approaches adjust the transmission rates of multimedia data in accordance with network performance.

Another method to transmit digital media across general packet-switched networks is by using a "best-effort" technique. Jeffay uses a best-effort approach for the delivery of audio and video based on a combination of transport, display, and operating system processes [18]. The transport mechanisms include techniques to avoid network congestion, provide forward error correction, manage the data leaving the server, and vary the synchronization between audio and video so that continuous audio is achieved. The combined use of several types of processes creates a effective system for delivering digital media data across packet-switched networks; however, the authors admit some shortcomings. The transport mechanism is not as effective when the number of intermediate networks increases. Also, the system only considers audio and video streams, unlike the LAP approach which is designed for orchestrated continuous and discrete multimedia data.

There are related approaches to media delivery which are specifically designed to support large user populations via multicasting. In the system of Bolot et al. [4], each receiver indicates to the sender what quality of video it is receiving based on its rate of packet loss. The sender can then increase or decrease the amount of data it introduces into the network by making adjustments to its encoding process.

Another protocol designed specifically for data delivery through multicasting is the Multimedia Multicast Channel (MMC) [29]. In the MMC work, the sender and receiver are only loosely coupled. The "open-loop" interaction between the sender and receivers is well suited for the multicast model of real-time continuous media. Each receiver filters the multicasted data to suit its specific application.

The LAP scheduling technique described in this paper is distinct from the aforementioned approaches for a variety of reasons. The LAP scheduler has been implemented and demonstrated in conjunction with an Ethernet network and has been shown to be effective for an FDDI network as well [14, 15]. Moreover, it can function in a variety of single protocol and integrated network environments because of its use of a generic delay model and a delay esti-

mation technique. Real-time network communications and the approach proposed by Elliot for packet-switched networks require the establishment of new low-level network protocols. Moreover, the use of existing protocols (e.g., token ring) satisfies LAN-based applications but fails when multiple interconnected networks are used.

The LAP also deals with multiple sources. Flow control mechanisms are usually designed for the retrieval of data from a single source and do not provide mechanisms to selectively degrade media delivered from a variety of distinct sources. In contrast, multicast algorithms are created to deliver to many destinations. While users can decide whether or not to receive a stream (i.e., to receive audio but not video of a conference) and can filter received data, they cannot individually control the stream emanating from the source. The LAP approach uses a one-to-one system which manages the data from the receiver. It can degrade the delivered stream if required by bandwidth limitations and can be easily expanded to handle multiple streams from multiple sources.

The remainder of this paper is organized as follows. In Section 2 the retrieval delay model and corresponding estimation techniques are described. Section 3 provides a description of the LAP scheduler including the algorithms used for service degradation under bandwidth and buffer-limited situations. Results from the implementation and experimentation using Ethernet are discussed in Section 4. Section 5 concludes the paper.

2 Retrieval Delay Model and Delay Estimation

The transmission of time-dependent data on computer networks is ideally be performed using network performance guarantees, but on most networks today there are no such guarantees. Therefore, the retrieval of multimedia data require an assessment of the current network or communication channel (henceforth, "channel") performance. With delay modeling and delay estimation, the retrieval of multimedia objects can be scheduled so that objects arrive at the playout system before their presentation times, but not so early as to overflow the allocated buffer space of the playout system. The delay model used by the LAP scheduler in a multihop network is described first, followed by the window-based probabilistic delay estimation technique.

2.1 Delay Model

A network delay characterization can be used by the scheduler to overcome delay and bandwidth limitations by choosing bounds on retrieval delay and bandwidth usage. As illustrated in Fig. 2, a *control time* T_1 can be selected corresponding to a likelihood F_1 that a packet arrives on time. Unfortunately, the network delay characterization is not stationary and such a measurement is valid for a limited period. Our approach adapts to these changes by monitoring the channel delay distribution and adjusting the retrieval schedule accordingly.

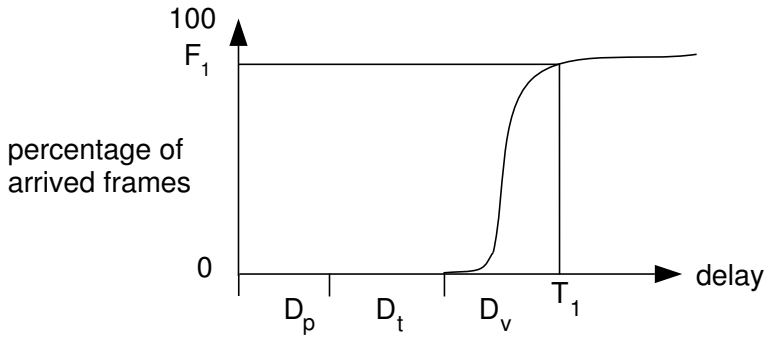

Figure 2: Probability Distribution Function for Single Packet Retrieval Delay

When an object composed of r packets is retrieved across a network, the delay estimate for this retrieval D_O consists of a constant overhead (propagation) delay D_p, a transmission delay D_t, and a variable delay D_v^O:

$$D_O = D_p + rD_t + D_v^O.$$

The variable delay D_v^O for a multihop network is modeled considering the variable portion of the total *trip time* from source to destination of one packet, D_v^T, and the variable portion of the time for the remaining packets to arrive after the first packet has arrived, D_v^{CI}. The variable portion of the time for the remaining packets to arrive is found by summing the variable interarrival delays between packets, $D_v^{CI} = \sum_{j=1}^{r-1} D_{v_{j,j+1}}^I$. The variable delay model for a single hop network, the trivial case of the multihop network model, is described in references [16, 24].

Table 1: Symbols Used in the Delay Model

D_p	propagation delay in retrieval
D_t	transmission delay in retrieval
$D_{v_i}^{CI}$	cumulative variable interarrival delay for object i
$D_{v_i}^{O}$	variable delay in retrieval of object i
D_{O_i}	total delay in retrieval of object i
T_{O_i}	control time for retrieval of object i
$T_{v_i}^{O}$	variable delay component of T_{O_i}
D_v^T	variable portion of total packet trip
D_v^I	variable portion of arrival time between packets
F_a	desired percentage of on-time arrivals for medium a
$\mu_{D_v^I}, \mu$	mean of interarrival delay
$\sigma^2_{D_v^I}, \sigma^2$	variance of interarrival delay

2.2 Delay Estimation

Because D_v^{CI} represents the sum of the variable portion of differences in packet arrival times that are assumed to be independent and identically distributed, the mean $\mu_{D_v^I}$ and variance $\sigma^2_{D_v^I}$ of the variable delay distribution approximate (by the Central Limit Theorem) the mean and variance of the delay distribution for retrieving an r packet object, $\mu_{D_v^{CI}} = r\mu_{D_v^I}$ and $\sigma^2_{D_v^{CI}} = r\sigma^2_{D_v^I}$. Note that the independence assumption yields an approximation that is an important component of a network delay model that is shown to be effective (Section 4. The correspondence between the actual variable delays of network packets for a given session varies due to network conditions: the busier the network, the closer the variable delays are to being independent. This is due to the additional effects of other sessions using the same channel, i.e., effects of the the predecessor and successor packets are diminished.

The use of a normal approximation in a network model was suggested by De Prycker et al. [9]. It is used to estimate the variable portion of a retrieval delay in a single-hop case by Little and Ghafoor [24] and in a multi-hop case by Gibbon [16]. A benefit of using a normal approximation when estimating retrieval delay times is the simplicity of using the error function $erf(g)$ to select a control time corresponding to the desired likelihood of on-time arrival. Furthermore, only two parameters need to be estimated: the mean μ and variance σ^2.

For the LAP scheduler, estimates of these parameters are calculated using a window based approach. If $(D_{v_1}^I, D_{v_2}^I, D_{v_3}^I, ..., D_{v_w}^I)$ are the last w recorded variable interarrival delays, independent and identically distributed, and are from a variable delay distribution, then $\hat{\mu}$

is an unbiased and consistent estimator of the mean μ if it is defined as [30]:

$$\hat{\mu} \stackrel{\text{def}}{=} \frac{1}{w} \sum_{i=1}^{w} D_{v_i}^I.$$

The estimator $\hat{\sigma^2}$ is an unbiased and consistent estimator of variance σ^2 if it is defined as [30]:

$$\hat{\sigma^2} \stackrel{\text{def}}{=} \frac{1}{n-1} \sum_{i=1}^{n} (D_{v_i}^I - \hat{\mu})^2.$$

An advantage of this window estimator is that it lends itself to a technique that determines whether the network load has recently changed. We define a test for this condition based on a likelihood ratio factor defined below:

$$\text{likelihood ratio factor } (lrf) \stackrel{\text{def}}{=} \frac{1}{w} \sum_{i=1}^{w} \frac{(D_{v_i} - \hat{\mu}_s)^2}{\hat{\sigma^2}_s}.$$

Using the mean $\hat{\mu}_s$ and variance $\hat{\sigma^2}_s$ estimates used to create the current schedule, an evaluation, called the likelihood ratio test, is employed to determine if the more recent arrivals are part of the delay distribution modeled by $\hat{\mu}_s$ and $\hat{\sigma^2}_s$ or if they represent a new delay distribution.

The likelihood ratio test is defined as $lrf \gg 1.0$. If the retrieval delay distribution has changed, then the current mean $\hat{\mu}_s$ and variance $\hat{\sigma^2}_s$ estimates no longer effectively model the current delay distribution and the likelihood ratio factor will grow to a number much greater than one.

Because the variance estimate (denominator) is defined in terms of the mean estimate and the original delay samples, and the numerator involves the mean estimate and the new delays; the lrf ratio will be approximately one if the new packets continue to experience the original network conditions. If new network conditions exist then the likelihood ratio factor will be significantly greater than one. This signals the LAP to create a new retrieval schedule using new estimates of μ and σ^2. The new estimates will be based on the w_{small} ($w_{small} < w_{max}$) most recent packet arrivals to increase the likelihood that all the retrieval delays used reflect the new delay distribution. As more packets arrive after the detected change in distributions, the number of packets used to estimate μ and σ^2 returns to w_{max}.

The types of changes in the delay distribution that the likelihood ratio test will most likely detect are discrete or long-lived. These types of changes will occur if there is an

increase or decrease in the number of multimedia sessions utilizing the network for retrieving digital video.

A different network dynamics scenario is characterized by slow drifts of the network delay distribution. This type of network behavior requires a window size (w_{max}) and lrf_{limit} small enough so that performance estimates are always within a tolerable region of accuracy. If there are short-lived and frequent changes in the network load due to activities such as file transfers, w_{max} should be large enough to encompass delays characteristic of these periods. If bursts are infrequent yet still potentially detrimental to scheduling in the network, a portion of the window can be set with constant values characteristic of delay retrieval during burst activities.

The LAP scheduler does not assume that all packets must be transmitted on the same physical path through the network. It is possible that the characteristics of a retrieval delay distribution are due in part to packets traversing different paths through the network. If there is a change in the manner in which packets are being routed, resulting in a change in the delay distribution, the LAP scheduler can adapt to the new condition.

3 Limited *A Priori* Scheduler

In this section we describe the limited *a priori* (LAP) scheduler. By using network delay estimation and real-time scheduling, the LAP scheduler, residing on a receiving (presentation) machine, manages the timing of retrieval for multimedia objects. The LAP scheduler is comprised of two main components: the *Schedule Creator*, corresponding to a static resource reservation mechanism, and the *Delay Estimator*, corresponding a dynamic delay modeling mechanism.

Fig. 3 shows the relationship between these components. Their behavior is as follows. After the LAP Scheduler receives information about the data to retrieve, the Schedule Creator generates retrieval schedules that utilize the delay estimates maintained by the LAP Delay Estimator. Once a retrieval schedule is created, it is sent to the data server (source) where it is used as the script for timing the transmission of the selected objects. At the destination, the Delay Estimator receives the multimedia objects, collects statistics on arrival times, and forwards the objects to the appropriate playout subsystem. The management of the various playout subsystems is not performed by the LAP scheduler in our implementation; however, these components have more well defined timing behaviors.

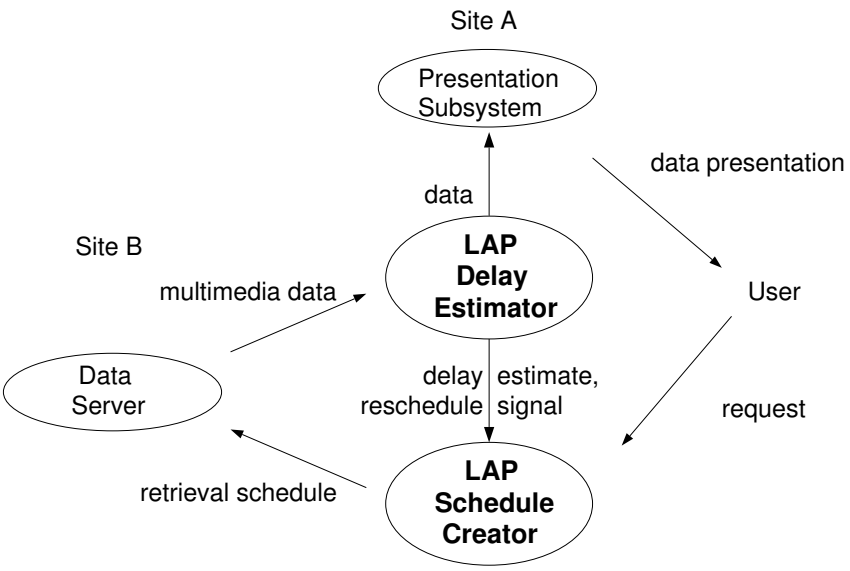

Figure 3: Components and Data Flow of the LAP Scheduler

A function of the LAP scheduler is the creation of schedules for future durations, or periods, of the multimedia presentation. A new period is enacted either at the end of the current period or any time when required by change in system state. State changes are caused by user requests (user interaction) or by significant changes in the current delay or bandwidth of the channel. Figure 4 illustrates the playout of a period $n + 1$ and the scheduling for period $n + 2$ that is interrupted by a "fast-forward" command issued by a user. The subsequent period, $n + 6$, is then scheduled and enacted based on this change in system state. Details of the operation of the Schedule Creator are described below.

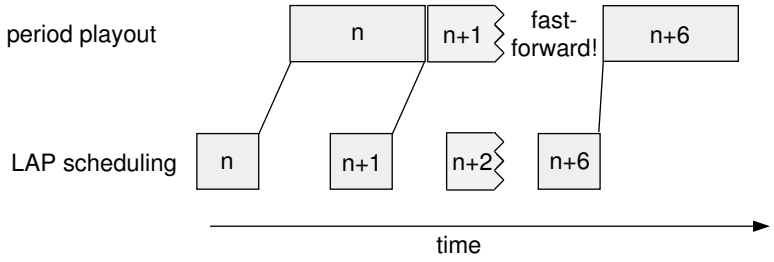

Figure 4: The Creation of Schedules for Multiple Periods

3.1 Schedule Creator

The Schedule Creator produces a retrieval schedule each period of the playout schedule. It schedules the retrieval of an object according to:

- The time to retrieve the object,
- the available bandwidth of the channel,
- the available memory at the receiver allocated to buffering, and
- the interaction and competition for resources of the object with others from the same session.

Prior to addressing these techniques we introduce a set of *media presentation factors* used by the LAP scheduler. These factors allow us to consider the special delivery characteristics of each medium.

3.1.1 Media Presentation Factors

By using a normal approximation and corresponding error function $erf(g)$[1] a control time T_O can be selected that corresponds to a percentage of on-time arrivals. As proposed by Montgomery in a study of packet voice synchronization, a percentage of on-time arrivals can be selected which is suitable to an application and provides a bounded retrieval delay [26]. When there are several media whose playouts need to be synchronized, it is difficult to determine the appropriate values that should be used for factors such as percentage of on-time arrivals. Steinmetz and Engler contend that the acceptable difference between the playout time of a video image and its corresponding audio in a "talking head" application is less than 160 ms [31]. A similar characterization, including specifications for delay, throughput, and reliability, is proposed by Ferrari for the communication of various real-time end-user applications [12].

The LAP scheduler uses two media presentation factors. Media are degraded in relation to their minimum acceptable presentation percentage P_M and retrieved with a probability of on-time arrival of F_m. The parameter P_M represents the smallest fraction of complete objects must be played-out for a given media. The penalty for dropping a degradeable media object

[1] $erf(g) = \frac{1}{\sqrt{2\pi}} \int_0^x exp(-\frac{1}{2}t^2)dt$

(e.g., a video frame) is assumed to be inversely proportional to the defined P_M corresponding to the medium. Note that some media such as graphic and text objects cannot tolerate losses and we call them non-degradeable.

We assume that there exists a relationship between perceived quality and these parameters; however, no specific criteria are suggested for selecting values for F_m and P_M since appropriate values vary for different media, applications, end-users, and compression standards. For example, when video is compressed using the JPEG standard, losses to individual frames do not affects other frames in the stream. When interframe compression approaches are used (e.g., MPEG I), a single lost packet can cause the loss of many subsequent video frames. The use of the media presentation factors is, at best, a coarse method for mapping media presentation requirements to resource availability and requires considerable further investigation.

3.1.2 Retrieval Before Playout and Bandwidth Averaging

The Schedule Creator schedules the retrievals for multimedia objects comprised of non-degradeable media as well as degradeable (lossy) media. The following description focuses on the scheduling of two media: a non-degradeable medium A and a degradeable medium B. Here we assume that there is at least sufficient bandwidth available to retrieve all of the non-degradeable media, although possibly at the expense of the degradeable media.

The LAP scheduler creates the retrieval schedule as follows. It begins with the last object of the period and calculates the time T_v^O necessary for object i to be retrieved with an on-time arrival probability F_m. This is performed using the error function $erf(g)$ and estimates $\hat{\mu}$, $\hat{\sigma^2}$, and $D^{\hat{T}}_{v_{F_m}}$: $T_v^O = D^{\hat{T}}_{v_{F_m}} + \sum_{j=1}^{r-1} D^I_{v_{j,j+1}}$ where $\sum_{j=1}^{r-1} D^I_{v_{j,j+1}}$ is estimated by $(r_i - 1)\hat{\mu} + g_m\sqrt{(r_i - 1)\hat{\sigma^2}}$. An $erf(g)$ table is used to find g_m for $F_a - 0.5$. The number of packets, r, needed for transmitting an object is found by dividing the object size by the packet size and rounding up, i.e., $r_i = \lceil(|x_i|/S)\rceil$. To maintain accurate statistics, we use a packet size less than or equal to the smallest transport packet size to prevent additional packet fragmentation.

Once T_v^O is established, the total object delay T_O is calculated for the object being retrieved: $T_O = D_p + rD_t + T_v^O$. The retrieval time ϕ is then set based on the playout time, π, of the object. In our implementation, an additional time, T_{OS}, is subtracted from the final retrieval time to compensate for additional processing time due to algorithm execution and operating system overheads: $\phi_i = \pi_i - T_{O_i} - T_{OS}$. The time execute the algorithm is

Table 2: Symbols Used to Describe the LAP Scheduling Mechanism

ϕ_i	object i retrieval time
π_i	object i playout time
S	packet size (fixed)
$\|x_i\|$	size (bits) of object i
C	channel capacity
Q	buffer space
L	maximum period length
g_m	input value for error function
$\hat{\mu_s}, \hat{\sigma_s^2}$	interarrival delay estimates used in current schedule
P_M	minimum playout percentage
U	change in minimum playout percentage for lossy media
lrf	likelihood ratio factor
lrf_{limit}	likelihood ratio factor limit
t	clock time
Z	network bandwidth usage reduction factor

bounded by the the number of items in the scheduling period. Though we recognize that the operating system overhead is variable we do not attempt to model this here.

The retrieval time for the predecessor object is similarly calculated, but now considering competition for use of the channel by other object in the same session. If the retrieval time of the predecessor object plus the time it takes to send the object is later than the scheduled retrieval time of the object ($\phi_{i-1} + r_{i-1}(\hat{\mu} + D_t) > \phi_i$), then the predecessor is rescheduled in relation to the successor: $\phi_{i-1} = \phi_i - r_{i-1}(\hat{\mu} + D_t)$. Here $r_{i-1}(\hat{\mu} + D_t)$ defines the average time it takes to put object $i-1$ onto the channel. In this manner a form of bandwidth averaging is introduced whereby objects are retrieved earlier than needed in order to compensate for the variations in the playout schedule [24]. In Fig. 5 the third object of medium B (B_3) is retrieved for playout earlier than needed so that the second object of medium A (A_2) can be retrieved on time.

3.1.3 Object Degradation to Accommodate Bandwidth Constraints

Bandwidth averaging can be used to overcome variations in the resource requirements of a multimedia presentation; however, if the average network bandwidth needed by the presentation is higher than the current available network bandwidth, the retrieval and subsequent playout of the media must be degraded. By not retrieving some objects during a period (e.g.,

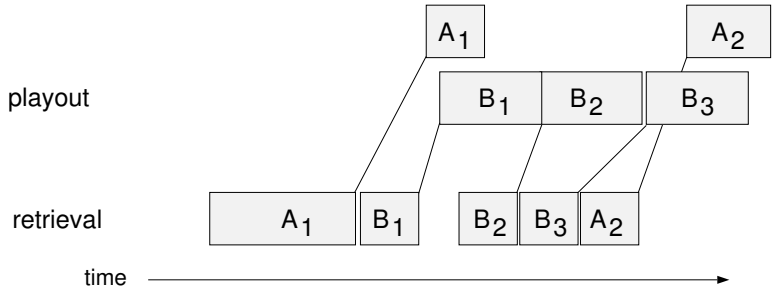

Figure 5: Bandwidth Averaging Through Manipulation of Retrieval Times

video frames), the resources requirements of a presentation can be reduced to fit within available channel resources.

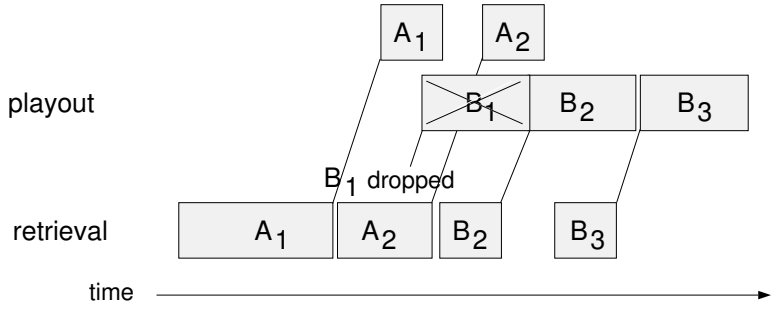

Figure 6: Degradation of Lossy Medium (B) by Dropping Object B_1

Figure 6 illustrates the dropping of object B_1 from the retrieval schedule (and subsequently playout schedule) due to bandwidth limitations. The decision to drop an object is achieved by the Schedule Creator which uses the network capacity ($C^L = |\mathrm{x}|/(D_t + \mu)$) and the total volume of data for the period to determine if degradation is required. If degradation is necessary, the various media comprising the presentation are degraded based on their media presentation factors.

3.1.4 Degradation to Accommodate Buffer Space Limitations

Another reason a retrieval schedule might be degraded is the limited availability of buffer space at the receiver. In some scenarios it is possible that several objects can be in the channel at the same time. These scenarios include high-capacity networks (e.g., gigabit networks) or networks with highly variable transmission delays. In the later case, an object

must be scheduled to consider very long delays while at at the same time buffer space must be allocated to deal with early arrivals due to short delays. Moreover, because most objects (e.g., frames) must be received in their entirety before being useful for playout, the buffer must be able to accommodate the largest object of the presentation.

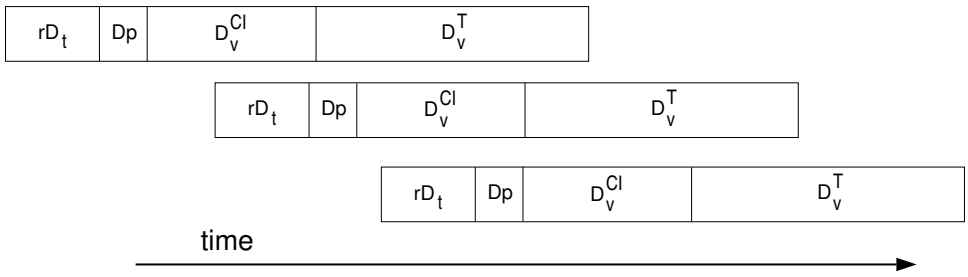

Figure 7: Network Delay during Retrieval of Three Objects Across a Multihop Network

The largest volume of data that can arrive occurs when the variable transmission delay and the variable interarrival delay equal zero (i.e., $D_v^T = D_v^{CI} = 0$). This is illustrated in Fig. 7. Here the object remains in the buffer for the time allocated for the variable delay. If other objects also experience a zero variable delay, then data arrive at a maximum rate proportional to the percentage of network capacity being utilized. This percentage is expressed in terms of the minimum distance between retrieval times $min(\phi_j - \phi_{j-1})$ for objects that arrive after the current object arrives but before it is played out. The required buffer space ($Q_{maximum\_required}$) is therefore calculated with the equation:

$$Q_{maximum\_required} = |x_i| + C(D_v^T + D_v^{CI})(rD_t)/(min(\phi_i - \phi_{i-1})).$$

When maximum network resources are being used $min(\phi_j - \phi_{j-1}) = rD_t$.

When this required buffer space exceeds the available buffer space (i.e., $Q_{required} > Q_{max}$), then the retrieval schedule is degraded. Figure 8 illustrates the case where B_3 is dropped because non-degradeable objects A_1 and A_2 would otherwise occupy the buffer at the same time as B_3; however, there is insufficient space for all of the objects.

In the LAP implementation, a list is maintained to determine the current buffer occupation. This is done by considering each object's playout time and earliest possible arrival time $t_{earliest\_arrival}$. A conservative approach to determining $t_{earliest\_arrival}$ assumes that the entire object requires buffering as soon as the first packet can arrive (i.e., $t_{earliest\_arrival} =$

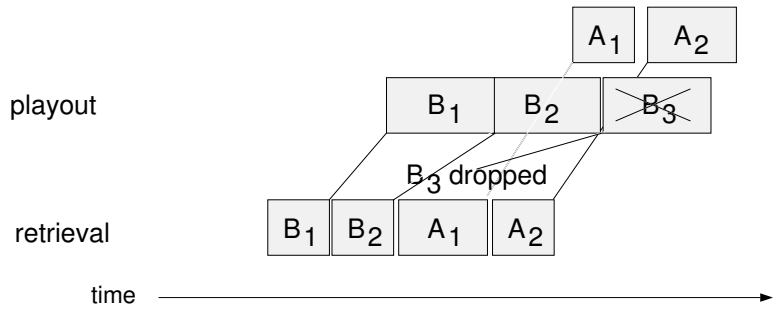

Figure 8: Degradation of Lossy Medium (B) by Dropping Object B_3 from the Schedule

$\phi_i + D_p + D^{\hat{T}}_{v_{min}}$, where $D^{\hat{T}}_{v_{min}}$ is used to compensate for asynchronous clocks). A less conservative estimate can be made assuming a double buffering scheme. In this case $t_{earliest\_arrival}$ represents the earliest time that all packets comprising the object might be received and therefore transferred to the second buffer (i.e., $t_{earliest\_arrival} = \phi_i + D_p + D^{\hat{T}}_{v_{min}} + r(D_t + rD^{\hat{I}}_{v,min})$).

3.2 Delay Estimator

The LAP Delay Estimator uses a window estimation technique to determine the network performance. When a new packet arrives the estimates for $\hat{\mu}$ and $\hat{\sigma^2}$ are updated according to whether the estimator window has already reached the size w_{max} or if it is still expanding. If the time remaining in the current schedule is less than or equal to the time needed to create, request, and process a new schedule, then the Delay Estimator requests a new schedule from the Schedule Creator. If the lrf exceeds lrf_{limit} or the user requests a presentation change, then the estimator window shrinks to the w_{small} lastest arrivals; $lrf, D^2_v, \hat{\mu}$, and $\hat{\sigma^2}$ are recalculated; and a new schedule is requested.

The Network Delay Estimation algorithm is shown below. It is presented in a generalized form which is valid for both the asynchronous and synchronous clock cases. The propagation delay and retrieval delay are set to zero ($D_p = D_t = 0$) and are encompassed by the variable retrieval delay (i.e., D_t for the first packet, D_p in $D^T_{v_1}$, and D_t for the remaining packets in $\sum_{j=1}^{r-1} D^I_{v_{j,j+1}}$: $D_O = \sum_{j=1}^{r-1} D^I_{v_{j,j+1}} + D^T_{v_1}$). The recorded delay values can be of arbitrary size (even negative) due to the difference in the source and destination clocks; however, delay estimation is not affected by these anomalies because clock skews are negated when the retrieval times created by the LAP scheduler (on the receiver) are used by the data server (source).

Network Delay Estimation Algorithm

1. If new packet i arrives
   (a) Determine total trip delay for new packet:
   $D^T_{v_i} = t_{stamped_i} - t_{arrived_i}$
   (b) If not first packet to arrive for new object:
   $D^I_{v_{i-1,i}} = t_{arrived_i} - t_{arrived_{i-1}}$
   (c) If window is not expanding ($w = w_{max}$)
      i. Calculate new mean estimate:
      $\hat{\mu} = ((\hat{\mu} * w) - D^I_{v_{i-1-w,i-w}} + D^I_{v_{i-1,i}})/w$
      ii. Calculate new D^2_v estimate:
      $\hat{D^I_v}^2 = ((\hat{D^I_v}^2 * w) - {D^I_{v_{i-1-w,i-w}}}^2 + {D^I_{v_{i-1,i}}}^2)/w$
   (d) If window is expanding ($w < w_{max}$)
      i. Calculate new mean estimate:
      $\hat{\mu} = ((\hat{\mu} * w) + D^I_{v_{i-1,i}})/(w+1)$
      ii. Calculate new D^2_v estimate:
      $\hat{D^I_v}^2 = ((\hat{D^I_v}^2 * w) + {D^I_{v_{i-1,i}}}^2)/(w+1)$
      iii. $w = w + 1$
   (e) Calculate new variance estimate:
   $\hat{\sigma^2} = \hat{D^I_v}^2 - \hat{\mu}^2$
   (f) Calculate current likelihood ratio factor:
   $lrf = ((lrf * w * \hat{\sigma^2_{ia_s}}) - (D^I_{v_{i-w}} - \hat{\mu_{ia_s}})^2 + (D^I_{v_i} - \hat{\mu_{ia_s}})^2)/(w * \hat{\sigma^2_{ia_s}})$
2. If user requests a change in presentation or current schedule about to expire (time left in current schedule is near time it takes to create, send, and process new schedule) then request new schedule from Schedule Creator.
3. If user requests change, or likelihood ratio test indicates that estimates used for current schedule do not accurately model the current network performance, $lrf > lrf_{limit}$, then:
   (a) Request next schedule from Schedule Creator,
   (b) Shrink window: $w = w_{small}$, and
   (c) Recalculate $lrf, {D^I_v}^2, \hat{\mu}, \hat{\sigma^2}$. for new window.

3.3 Extensions

The Schedule Creator and Schedule Executor algorithms presented assume a single data source; however, both can be extended for multiple sources. When two (independent) sources contend for a common channel, then a single LAP schedule can be created for its management. This requires maintaining retrieval delay parameters for each data source. Similarly, different estimates must be maintained when different media are retrieved from the same source but via different channels or protocols. For example, different protocols might be used to transmit media requiring reliable transmission, while the same channel supports a protocol suitable for a loss-tolerant medium (e.g., TCP vs. UDP). When different sources do not share common resources then different LAP schedulers are enacted. This occurs when one source is remote and the other is local storage. In this situation the schedules must be coordinated.

A further enhancement to the LAP scheduler is the use of a *bandwidth usage reduction factor*, Z, that is inversely proportional to the percentage of lost (discarded) packets. In the implementation Z is assigned a static value that is used in channel capacity calculations to reduce the percentage of channel bandwidth utilized by the LAP scheduler. By using Z to limit the use of channel capacity, the likelihood of channel bottlenecks and subsequent lost packets decreases. In the current implementation the available channel capacity is determined by the equation, $C^L = Z * |\mathrm{x}|/(D_t + \mu)$ where $0 < Z < 1$.

4 Implementation and Experimentation

The LAP scheduler is implemented in C under Unix. Communication between processes on the same machine is performed using semaphores and shared memory. Control information sent across the network, including retrieval schedules, uses TCP, whereas multimedia data are delivered using UDP.

The scheduler was evaluated using two Sun IPX workstations configured with motion-JPEG decompression boards by Parallax Graphics, Inc., and interconnected by 10 Mbit/s Ethernet. Being a part of a university-wide network, the network segment used for experimentation did not have well-behaved traffic. The multimedia presentation used for evaluation was a 436-frame motion-JPEG-compressed video sequence. The average frame size was approximately 14.5 Kbytes. In addition, some of the trials used large (≈145 Kbytes) non-degradeable multimedia objects in the presentation schedule. The packet size was chosen to

be 1,400 bytes so that the network protocols would not fragment our transmitted units (the maximum size used by the network was 1,500 bytes).

Data were recorded to characterize the performance of several different multimedia sessions. In the following subsections, we describe each session and the corresponding results obtained during experimentation.

4.1 Session A

For Session A, a sequence of digital video was played at 5 frames per second (f/s) to demonstrate the basic operation of the LAP scheduler. Figure 9 illustrates the network performance by indicating the change in the estimated interarrival mean. As discussed in Section 2.1 the delay for retrieving the object from the source is $D_O = D_p + rD_t + D_v^O$ and the variable retrieval delay is expressed as $D_v^O = D_{v_1}^T + \sum_{j=1}^{r-1} D_{v_{j,j+1}}^I$. Because the source and destination clocks were not synchronized the propagation delay and constant portion of the transmission delay cannot be distinguished from clock skew. Therefore these delays were set to zero ($D_p = D_t = 0$) and were encompassed by the variable retrieval delay (i.e., D_p and D_t for the first packet being represented in $D_{v_1}^T$ and D_t for the remaining packets in $\sum_{j=1}^{r-1} D_{v_{j,j+1}}^I$). The asynchronous clock case (which encompasses the synchronous clock case) has two time components, the time to "fill up the pipe," $\hat{D_{v_1}^T}$, and the time for the "pipe to drain," $\sum_{j=1}^{r-1} \hat{D}_{v_{j,j+1}}^I$. The total retrieval delay is estimated as: $T_O = \hat{D_{v_1}^T} + \sum_{j=1}^{r-1} \hat{D}_{v_{j,j+1}}^I$.

In Session A the retrieval delay for frame 200 was estimated as 61.8 s. This calculation was performed in the following manner. The transmission time $D_{v_1}^T$ was approximated for the 90th percentile (90% of the values are less than this value) as 61.77 s. This estimate is large because it encompasses the difference between the source and destination clocks. The interarrival time $\sum_{j=1}^{r-1} D_{v_{j,j+1}}^I$ was estimated by $(r_i - 1)\mu + g_m\sqrt{(r_i - 1)\sigma^2}$ to be 0.0291 s.

In this example, the delay estimation was performed using the asynchronous clock model. Therefore, the retrieval delay is comprised of the variable transmission and variable interarrival delays (i.e., $T_O = \hat{D_{v_1}^T} + \sum_{j=1}^{r-1} \hat{D}_{v_{j,j+1}}^I = 61.77 + 0.0291 = 61.80$ s). While the actual transmission time was unknown due to the difference between the clocks, the interarrival time indicates that the bandwidth provided to the session is approximately 14.5 Kbytes/0.0291 s or 0.498 Mbyte/s (less than half of the theoretical maximum Ethernet bandwidth).

Although the the performance of the network varied (Fig. 9), the scheduler successfully delivered all frames without missing a playout deadline. Because we added a one second

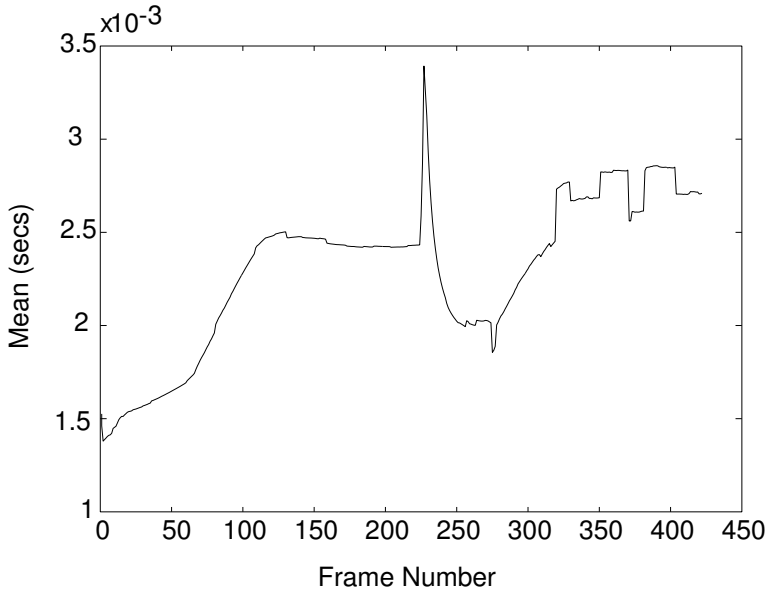

Figure 9: The Estimated Interarrival Mean for Session A

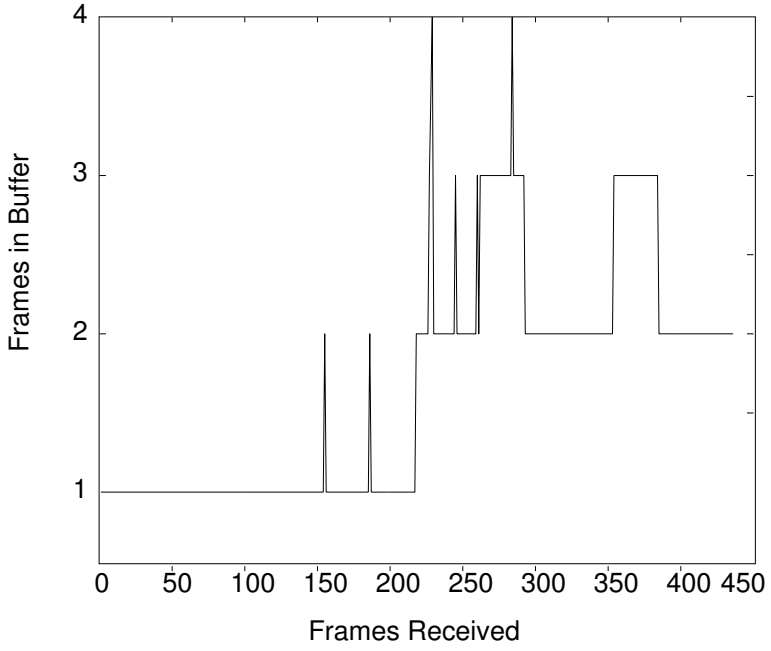

Figure 10: The Buffer Occupancy for Session A

operating system overhead time (T_{OS}) to the retrieval delay of the session (and all sessions), there is a one second margin for "on-time" playout extending one second before scheduled playout to the actual playout time. Therefore, for this case the the buffer must be sized to hold up to five frames as indicated by Fig. 10.

From the figure, it is apparent that the playout process began to fall behind and frames accumulated in the buffer. This could have been caused by other processes requesting CPU resources and the subsequent decrease in the priority of the playout process. In this experiment, there was always at least one frame in the buffer. Late arriving frames were not recorded in these measurements.

4.2 Session B

Session B is used to demonstrate the use of the likelihood ratio factor for degrading a retrieval schedule under bandwidth limitations. Here we used video identical to Session A but with a frame rate of 10 f/s. Therefore, this session required more processing and network resources. In addition, the network resources were further limited by allowing the network to be scheduled at only 40% of the allocated bandwidth for the presentation (i.e., $C^L = Z * |x|/(D_t + \mu)$ where in this case $Z = 0.4$).

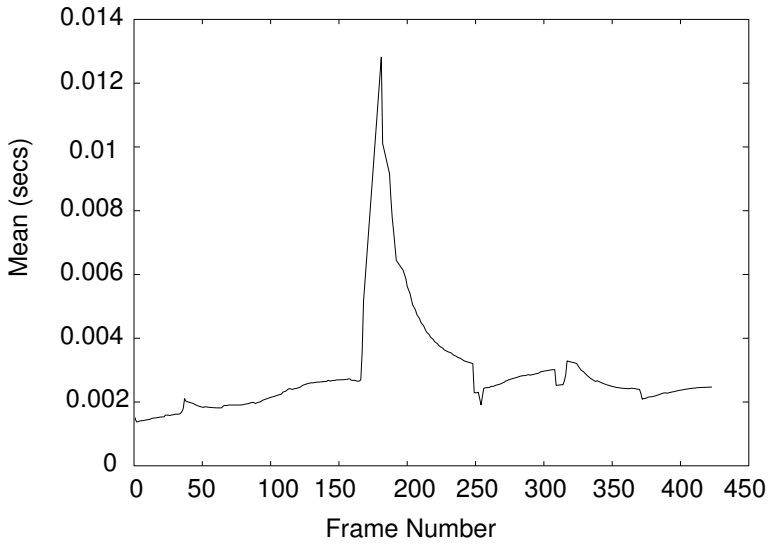

Figure 11: The Interarrival Mean for Session B

As seen in Fig. 11 there was a significant change in network performance in the region near the arrival of frame 175. The likelihood ratio test indicates (Fig. 12) that a new schedule

was needed after the arrival of frame 175 (the maximum value of the mean estimate in Fig. 11) and frame 181 when the network performance increased again. Because the schedule created at frame 175 was quickly superseded by the one created at frame 181, we examine the schedule created at 181.

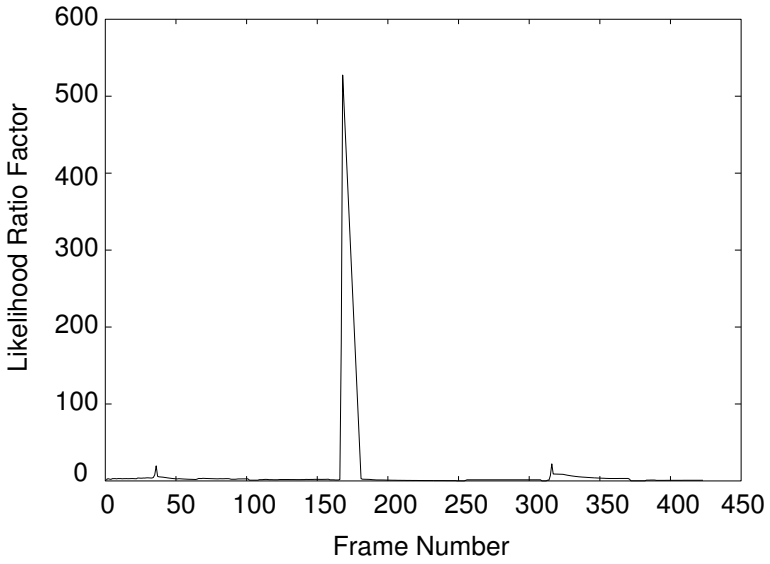

Figure 12: The LRF Value for Session B

When the schedule beginning with 181 was created, the interarrival mean μ was estimated as 0.0056 s. This implies that the network capacity allocated for the transmission of data for the period equals 100 Kbytes/s (i.e., $C^L = Z * |x|/(D_t + \mu) = 0.4 * 1.4$ Kbyte/0.0056 s = 100 Kbytes/s). The bandwidth needed to transmit 10 f/s approximately equals 145 Kbytes/s (i.e., 10f/s * 14.5 Kbytes/f = 145 Kbytes/s). Therefore, the LAP scheduler reduced the frames requested to 70% of the full playout rate, or 7 f/s (7f/s $*$ 14.5 Kbytes/frame = 102 Kbytes/s). The advantage here, in spite of the degradation, is the real-time, continuous playout of frames.

Figure 13 illustrates the average rate of scheduled frame playouts (and retrievals) as the session progresses. Notice that the presentation was scheduled to achieve a retrieval rate of approximately 7 f/s from frame 181 until a new schedule was created with new delay estimates at frame 254. The capacity calculated at frame 254 was 243 Kbytes/s (i.e., $C^L = 0.4 * 1400/0.0023$ s = 243 Kbytes/s) which is greater than the 145 Kbytes/s needed for 10 f/s and therefore no degradation was required. The irregularities in the curve between frame 181 and 254 are a result of the averaging technique that only considers entire frames.

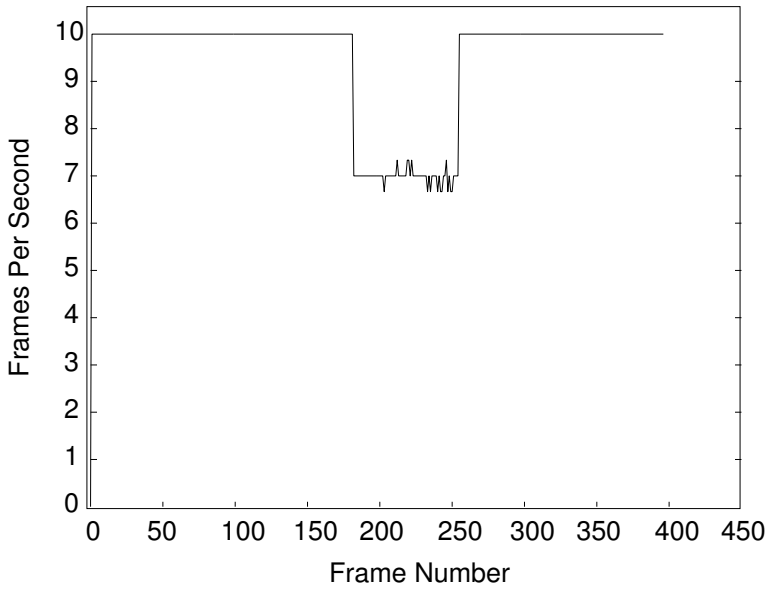

Figure 13: The Scheduled Retrieval/Playout Rate for Session B

4.3 Session C

Session C was designed to demonstrate the performance of the LAP scheduler when independent media are to be retrieved. In this session two large non-degradeable objects (≈145 Kbytes) were retrieved near frame 100 and frame 250 during the delivery of the previous video. The requested playout rate of video was 20 f/s with a 30% requested bandwidth capacity use ($Z = 0.3$).

Figure 14 illustrates the decrease in the rate of retrievals scheduled for the continuous (degradeable) medium in the region of the retrieval of the non-degradeable objects. The retrieval of the large non-degradeable object reduced the bandwidth available for the video frames as is seen near frame 100. The area surrounding frame 100 has a lower-than-requested scheduling rate for the continuous medium to accommodate the non-degradeable object. The effect of bandwidth averaging is seen here as the degradation occurs equally before and after the large object retrieval.

When the large object (appearing near frame 250) was retrieved, sufficient bandwidth existed so that the continuous media could be retrieved at the requested 20 f/s. Degradation of the continuous media occurred approximately 20 frames after the retrieval of the large object. This degradation was in response to a reduction in the network performance, possibly caused by the recent transmission of a large object.

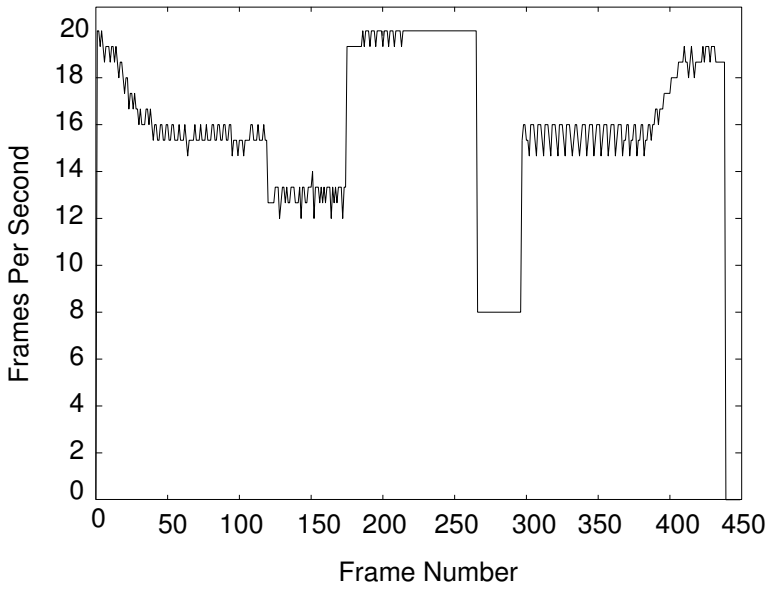

Figure 14: The Scheduled Playout/Retrieval Rate for Session C

4.4 Session D

Session D demonstrates the ability of the LAP scheduler to deal with the limitations of available memory for buffering at the receiver. Figure 15 illustrates buffer occupancy for a session comprised of video retrieved at 10 f/s with a target buffer limit of 100 Kbytes. Notice that this limit was exceeded on two occasions but was typically maintained below the 100 Kbyte limit.

In general, the maximum buffer occupancy for a session presenting 10 f/s is 140 Kbytes. This value is derived from the T_{OS} of one second which allows a frame that arrives at the scheduled time to remain in the buffer for one second (i.e., 10 f/s $*$ 1 s $*$ 14 Kbytes/f = 140 Kbytes). Buffer occupancy can rise as high as 300 Kbytes when objects arrive earlier than expected during an increase in channel capacity or when the playout process takes longer than the estimated T_{OS} of 1 second.

4.5 Discussion

Results from the implementation of the LAP scheduling mechanism illustrate several of its features. As calculated for Session A, the delay model achieves an approximation of the performance of the channel. This is further confirmed by the model's determination that

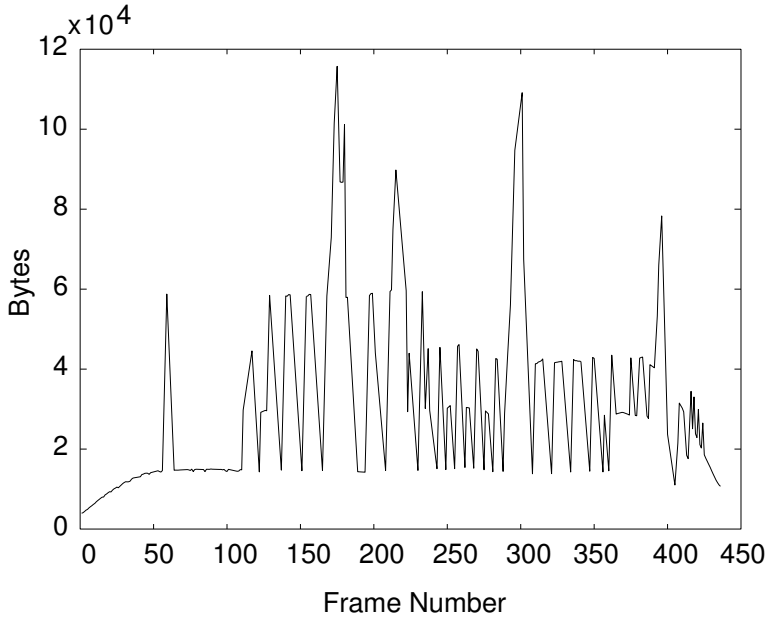

Figure 15: Buffer Occupancy for Session D

of approximately half (0.498 Mbyte/s) of the maximum capacity of the Ethernet network is available to the LAP scheduler. This is a reasonable estimation since the LAP scheduler is a high-volume user of the network which is shared by other processes with undetermined bandwidth utilizations (e.g., other users and NFS).

Figures 10 and 16 illustrate the efficacy of the LAP scheduler in retrieving objects for timely playout. In Fig. 16, the actual versus target percentages of on-time arrivals are illustrated when retrieving approximately 1,500 video frames during several sessions that operated under various network performance conditions. Each session requested 10 f/s and was scheduled using 50% of the available network capacity (i.e., Z = 0.5). Still there are limiting factors in the accuracy of the retrieval times due to the burstiness of traffic, the difficulty in predicting future channel behavior from past behavior, and the limitations of the normal approximation.

As mentioned previously, an *lrf* value can indicate a change in estimated network performance. However, it is often the case that network load changes are detected near the creation of a new retrieval schedule. The increase in the recorded interarrival delays may in some cases be caused, not by a change in network performance, but rather due to the additional time it takes for packets to be received by the client machine while it is busy creating a retrieval schedule.

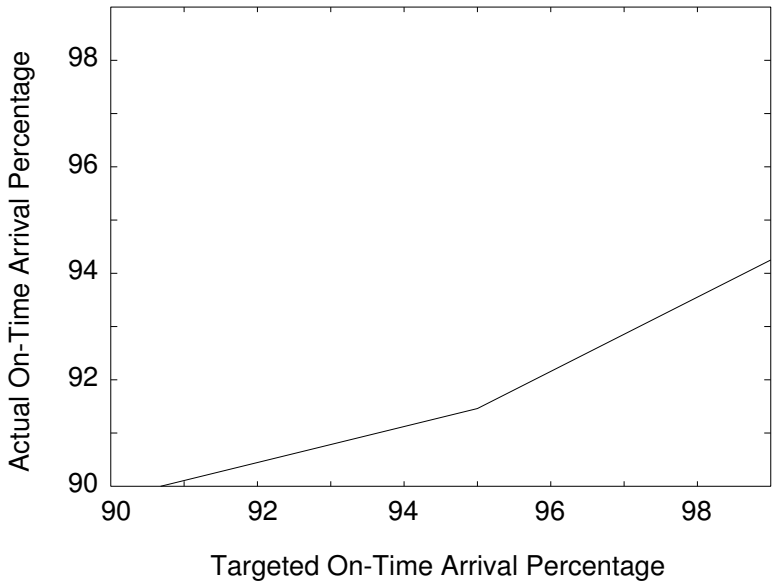

Figure 16: Targeted On-Time Arrival Percentage Versus Actual On-Time Arrival Percentage During Retrieval across a Multihop Ethernet

Results from Sessions B and C indicate that the LAP scheduler can effectively schedule a multimedia presentation in accordance with estimated network capacity. Specifically, Figs. 13 and 14 show how the LAP scheduler can degrade a digital media session when there are less than adequate channel resources. Similarly, results from Session D demonstrate the ability of the LAP scheduler to schedule considering buffer space limitations at the client station. While the LAP scheduler is generally successful at maintaining a specific buffer capacity limit, the spikes in buffer capacity that exceed the stated limit are a good illustration of the difficulties associated with predicting future network performance with past network performance. Often the network performance will vary significantly over a short duration. For example, if statistics are maintained for the last 10 seconds of network performance and then a large number of new users suddenly begin using the network, the statistics are no longer useful. We use the likelihood ratio test to indicate such a change but there remains a latency between the change in the network performance and when its effects are detected, new estimates are calculated, and a new schedule is created.

In the implementation discussed, the amount of processing required by the LAP scheduler did not affect its ability to perform effectively; however, there are be systems with slower CPUs, smaller buffers, and stricter limits on response time to user input in which the LAP scheduler will not perform as well due to its overhead of execution. Furthermore, our

implementation was affected by the variable performance offered to the algorithm by the multitasking operating system. It would be very interesting to execute the LAP algorithms on an operating system that provides deterministic or statistical real-time guarantees.

5 Conclusion

The LAP scheduler uses retrieval delay modeling and retrieval delay estimation for the timely delivery of multimedia data from across a generalized channel. The performance of the LAP scheduler is limited by factors such as the difficulty in predicting future network behavior from past performance, the use of normal approximations in the estimation process, and its implementation on a multitasking operating system. However, compared to other multimedia data delivery techniques, it is quite effective. In contrast, to achieve the same delivery characteristics, a simplistic control system for sending multimedia objects from a source to a destination would requires either a very large constant delay time and/or a large buffer.

The LAP scheduler is further distinguished from other multimedia data delivery techniques by its ability to degrade under bandwidth and buffer constraints, and its ability to accommodate the presentation of multiple media with different temporal presentation requirements from several sources. Moreover, it can be used with arbitrary network protocols because it uses a delay modeling and adaptation technique. This same property allows the scheme to respond to unpredictability due to dynamic user interaction.

References

[1] Ades, S., R. Want, and R. Calnan, "Protocols for Real Time Voice Communication on a Packet Local Network," *Proc. IEEE INFOCOM '87*, San Francisco, CA, March, 1987, pp. 525-530.

[2] Barberis, G. and D. Pazzaglia, "Analysis and Optimal Design of a Packet-Voice Receiver," *IEEE Transactions on Communications*, Vol. 28, No. 2, February 1980, pp. 217-227.

[3] Bisdikian, C.C., B. Patel, F. Schaffa, and M. Willebeek-LeMair, "On the Effectiveness of Priorities in Token Ring for Multimedia Traffic," *Proc. 18th Annual Conference on Local Computer Networks*, Minneapolis, MN, September 1993. pp. 25-31.

[4] Bolot, J.-C., T. Turletti, and I. Wakeman "Scalable Feedback Control for Multicast Video Distribution in the Internet", *Proc. ACM SIGCOMM '94*, London, England, August/September 1994, pp. 58-67.

[5] Braden, R, L. Zhang, S. Berson, S. Herzog, and J. Wroclaswki, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," *Internet Draft*, November 1995.

[6] Chen, H.-J. and T.D.C. Little, "Physical Storage Organizations for Time-Dependent Multimedia Data," *Proc. 4th Intl. Conf. on Foundations of Data Organization and Algorithms (FODO'93)*, Evanston, IL, October 1993, pp. 19-34.

[7] Chou, S.T.-C and H. Tokuda, "System Support for Dynamic QOS of Continuous Media Communication" *Proc. 3rd Intl. Workshop on Network and Operating System Support For Digital Audio and Video*, San Diego, California, November 1992, pp. 322-327.

[8] Decina, M. and T. Toniatti, "On Bandwidth Allocation to Bursty Virtual Connections in ATM Networks," *Proc. ICC '90* (IEEE Intl. Conf. on Communications), 1990, pp. 844-851.

[9] De Prycker, M., M. Ryckebusch, and P. Barri, "Terminal Synchronization in Asynchronous Networks," *Proc. ICC '87* (IEEE Intl. Conf. on Communications '87), Seattle, WA, June 1987, pp. 800-807.

[10] Elliott, C., "High-Quality Multimedia Conferencing through a Long-Haul Packet Network," *Proc. ACM Multimedia'93*, Anaheim, CA, August 1993, pp. 91-98.

[11] Escobar, J., D. Deutsch, and C. Partridge, "Flow Synchronization Protocol," *Proc. Conf. on Global Communications (GLOBECOM)*, Orlando, Florida, Dec. 1992. pp. 1381-1387.

[12] Ferrari, D., "Client Requirements for Real-Time Communication Services," *IEEE Communications Magazine*, Vol. 28, No. 11, November 1990, pp. 65-72.

[13] Ferrari, D. and D.C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE J. on Sel. Areas in Comm.*, Vol. 8, No. 3, pp. 368-379, April 1990.

[14] Gibbon, J.F. and T.D.C. Little, "Multimedia Data Delivery Using Network Delay Modeling," to appear in *Telecommunication Systems*, Special Issue on Performance Modeling and Analysis of Local Computer Networks, 1996.

[15] Gibbon, J.F. and T.D.C. Little "Real-Time Data Delivery for Multimedia Networks," *Proc. 18th Annual Conference on Local Computer Networks*, Minneapolis, MN, September 1993, pp. 7-16.

[16] Gibbon, J.F., "Real-Time Scheduling for Multimedia Services Using Network Delay Estimation," *Ph.D. Dissertation*, Multimedia Communications Lab TR 05-20-1994, Boston University, May 1994.

[17] Gilge, M. and R. Gusella, "Motion Video Coding for Packet-Switched Networks - an Integrated Approach," Tech. Rept. TR-91-065, International Computer Science Institute, Berkeley, CA, December 1991.

[18] Jeffay, K., D.L. Stone, T. Talley, and F.D. Smith, "Adaptive Best-Effort Delivery of Digital Audio and Video Across Packet-Switched Networks," *Proc. 3rd Intl. Workshop on Network and Operating System Support For Digital Audio and Video*, San Diego, CA, November 1992, pp. 1-11.

[19] Keshav, S.,"The Packet Pair Flow Control Protocol," ICSI Tech. Rept. TR-91-028, Computer Science Division, Department of EECS, University of California, Berkeley and International Computer Science Institute, Berkeley, CA, May 1991.

[20] Kishimoto, R., Y. Ogata, and F. Inumaru, "Generation Interval Distribution Characteristics of Packetized Variable Rate Video Coding Data Streams in an ATM Network," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, June 1989, pp. 833-841.

[21] Kobza, J. and S. Liu, "A Head-Of-Line Approximation to Delay-Dependent Scheduling in Integrated Packet-Switched Networks," *Proc. IEEE Infocom '89*, Ottawa, Ontario, Canada, April 1989, pp. 1106-1113.

[22] Lazar, A.A., A. Temple, and R. Gidron, "An Architecture for Integrated Networks that Guarantees Quality of Service," *Intl. J. of Digital and Analog Cabled Systems*, Vol. 3, No. 2, 1990, pp. 229-238.

[23] Lim, C.C., L. Yao, and W. Zhao, "Transmitting Time-Dependent Multimedia Data in FDDI Networks," *Proc. SPIE Symposium OE/FIBERS'92*, Boston, MA, September 1992.

[24] Little, T.D.C. and A. Ghafoor, "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Computer Comm.* Vol. 15, No. 6, July/August 1992, pp. 381-387.

[25] Lougher, P. and D. Shepherd, "The Design and Implementation of a Continuous Media Storage Server," *Proc. 3rd Intl. Workshop on Network and Operating System Support For Digital Audio and Video*, San Diego, CA, November 1992.

[26] Montgomery, W.A., "Techniques for Packet Voice Synchronization," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, No. 6, December 1983, pp. 1022-1028.

[27] Nomura, M., T. Fujii, and N. Ohta, "Basic Characteristics of Variable Rate Video Coding in ATM Environment," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, June 1989, pp. 752-760.

[28] Ohnishi, H., T. Okada, and K. Noguchi, "Flow Control Schemes and Delay/Loss Tradeoff in ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December 1988, pp. 1606-1616.

[29] Pasquale, J.C., G.C. Polyzos, E.W. Anderson, V.P. Komvella, "The Multimedia Multicast Channel," *Journal of Internetworking: Research and Experience*, Vol. 5, No. 4, December 1994, pp. 151-162.

[30] Stark, H., and J.W. Woods, *Probability, Random Processes, and Estimation Theory For Engineers*, Prentice-Hall, Englewood Cliffs NJ, 1986.

[31] Steinmetz, R. and C. Engler, "Human Perception of Media Synchronization," Tech. Rept. 43.9310, IBM European Networking Center, Heidelberg, Germany.

[32] Tokuda, H., Y. Tobe, S.T.-C.Chou, and J.M.F. Moura, "Continuous Media Communication with Dynamic QOS Control Using ARTS with an FDDI Network," *Proc. ACM SIGCOMM 92*, October 92, pp. 88-98.

[33] Yee, J. and P. Varaiya, " An Analytical Model for Real-Time Multimedia Disk Scheduling," *Proc. 3rd Intl. Workshop on Network and Operating System Support For Digital Audio and Video*, San Diego, CA, November 1992.

[34] Yu, P.S., M.-S. Chen, and D.D. Kandlur, "Design and Analysis of a Grouped Sweeping Scheme for Multimedia Storage Management," *Proc. 3rd Intl. Workshop on Network and Operating System Support For Digital Audio and Video*, San Diego, CA, November 1992.