Real-Time Data Delivery for Multimedia Networks[1]

J.F. Gibbon and T.D.C. Little

Multimedia Communications Laboratory
Department of Electrical, Computer and Systems Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
{*jfg,tdcl*}*@bu.edu*

MCL Technical Report 09-15-1993

Abstract–A coherent computer-based multimedia presentation requires the use of real-time scheduling theory applied to system resources including storage devices, the bus, and network. In this paper we present a scheduling technique which is essentially static in nature during periods of network quiescence, but becomes dynamic in response to detection of network load changes or user interaction.

This mechanism, called the *limited a priori (LAP) scheduler*, is an extension of earlier work on scheduling which assumed guaranteed system loading characteristics. The LAP scheduler determines network load changes and estimates packet delay using a dynamic filtering algorithm. We show the bounds of accuracy for this technique and describe its suitability with respect to multimedia traffic across an FDDI network.

Keywords: Multimedia, real-time networks, real-time scheduling, network delay estimation, resource management.

[1] In *Proc. 18th Annual Conference on Local Computer Networks*, Minneapolis, MN, September 1993, pp. 7-16. This work is supported in part by the National Science Foundation under Grant No. IRI-9211165.

1 Introduction

Distributed multimedia technology can be used for remote medical collaboration, on-demand movie services, instructional multimedia, and other applications which utilize audio, video, graphics, and text from various sources. Coordinating the presentation of distributed multimedia data is difficult since it can involve channels with different data delivery characteristics. This problem is compounded when data delivery mechanisms are being shared by multiple processes. Figure 1 illustrates a model of such a multimedia data delivery system involving a local computer network.

Because multiple processes compete for shared resources such as the communication channel, system resources which are involved in data delivery must be managed or scheduled. The job of a real-time *scheduler* is to manage the assignment of execution resources to tasks awaiting execution within the timing constraints assigned to each task. The real-time scheduler must allocate resources to tasks in a predetermined manner based on some system of priority to ensure timing satisfaction. Allocated resources can be the CPU, storage systems, the communication channel, or other system devices.

The playout timing for a complex multimedia object can be defined as a set of temporal relationships or as a playout schedule. Once this schedule is defined, a real-time scheduler can orchestrate the various resources for the given session. However, the resource requirements can change at any time during the session due to changes in the data presentation caused by user interaction. In addition, the scheduler must also take into account resource load changes which will effect the tasks awaiting execution.

A real-time multimedia scheduler must also be able to contend with limited resource availability when establishing sessions. As sessions are added to an already loaded system, resources must be distributed to multiple sessions and ideally should provide some form of *graceful degradation* through management of quality of service.

There is much recent literature covering real-time scheduling (e.g., Cheng et al. [3]). While providing a foundation for our work, research in real-time scheduling only indirectly addresses the varied performance characteristics needed to support multimedia data delivery. In order to manage system components such as the communication channel we propose to use real-time scheduling but only over a limited period of stable resource allocation. As the performance of the pertinent resources change, our mechanism adapts to maintain a constant quality of data delivery.

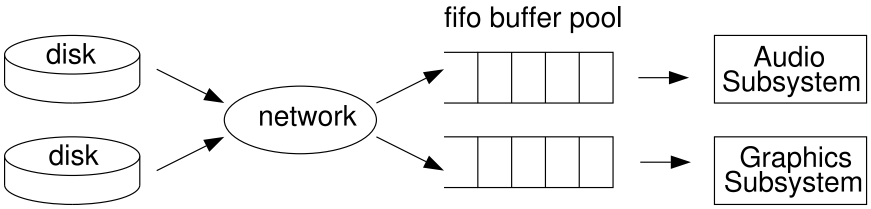

Figure 1: Delivery of Stored Multimedia Data

Scheduling the use of a communications channel can also be performed by networks dedicated to this task. A real-time network, as described by Ferrari et al. [4] and Lazar et al. [9], provides performance guarantees for data transmission. Absolute guarantees provided by the network are created through deterministic scheduling and resource allocation. Approximate guarantees are offered by using statistical methods. In a real-time network the user requests specific time-based communication requirements such as minimum, maximum, and average end-to-end delay; minimum and average arrival time between packets; packet size; packet jitter; packet loss rate; and maximum consecutive packet loss. The network then grants a connection if adequate resources exist.

Other protocols and policies bound the transmission time for time-constrained multimedia traffic for specific types of networks. The Asynchronous Transfer Mode (ATM) is considered for use with multimedia traffic due to its ability to dynamically allocate bandwidth [7, 16]. Other recent work in providing real-time guarantees for multimedia traffic includes support for FDDI networks [10] and packet switched networks [8, 6].

Research has also been conducted for managing the retrieval of time constrained data from rotating storage devices. These systems are divided into ones which advocate a specific layout on the hard disk for optimal retrieval [2, 14] and those that only use hard disk characteristics and buffering techniques for time-bounded retrievals [19, 20]. The specific topic of *synchronization* for a multimedia systems has been widely investigated (e.g., [12, 17]).

In our proposal, time-dependent data management is accomplished by a dynamic application of static *a priori* scheduling. The LAP scheduler is distinguished by the ability to accommodate system changes such as network delay variation or user interaction. Although we show results for an FDDI network, the models proposed are general in form and are not bound to a specific data delivery system.

The remainder of this paper is organized as follows. In Section 2 we formulate the basis for statistical scheduling. In Section 3 we describe our network load estimation technique. Section 4 describes the LAP scheduler and its performance when retrieving multimedia data over a simulated FDDI network. Finally, Section 5 concludes the paper.

2 Background

The use of real-time scheduling is essential in a multimedia system. We examine real-time scheduling theory by first describing some basic definitions and their application to multimedia data delivery. We then consider real-time communications.

2.1 An Overview of Real-Time Scheduling

Scheduling in a real-time system is classified as being either *hard*, *firm*, or *soft*. Hard real-time scheduling involves tasks with absolute deadlines, i.e., the system cannot recover if a task is not completed by an appointed time called its *deadline*. Systems which incur a finite penalty once a deadline is missed utilize *firm* real-time scheduling. *Soft* real-time scheduling involves deadlines which can be missed but the tasks must be eventually executed.

We consider multimedia scheduling to be firm real-time scheduling. Multimedia systems have a tolerance to missed delivery deadlines. For example, in most presentations if a graphic is displayed 0.5 s late, there is no noticeable change in the presentation as long as it is eventually displayed. In a video presentation if 10 frames are displayed 0.5 s late, their relationship to other frames would make the presentation nonsensical. Clearly, the tolerance to missed playout deadlines depends on the media and the presentation.

Another distinction of real-time systems is how the scheduling of tasks is performed, either *statically* or *dynamically*. In a static real-time system the information for all tasks that need to be scheduled is known *a priori*. A dynamic system must schedule tasks as they are requested without prior knowledge of their existence. The paradigm illustrated in this paper is of a multimedia system in which a session involving data from a stored source is requested and divided into periods. Each period is scheduled statically just before its presentation. This allows our limited *a priori* reservation system to accommodate recent user input or system load changes when creating the schedule for a period. For example, in a video-on-demand application, when new sessions are added to a system, the subsequent load changes can be distributed to all existing sessions. If a multimedia session involving non-stored data (i.e., live video from a camera) is requested, an exact bandwidth usage model cannot be created and therefore the communication channel would be scheduled based on data rate approximations.

2.2 Real-Time Communications

One technique for scheduling the communication of time-dependent data is to utilize a real-time network which provides performance guarantees for the delivery of data (see Section 1). Another way to manage time-dependent data transmission is to maintain statistics characterizing each communication channel. When the channel characteristics of the network change, the scheduler can adjust accordingly to maintain predictable service. Statistical approaches to overcoming delay and bandwidth limitations rely on choosing bounds on session delay and bandwidth usage. For example, an end-to-end control time T can be chosen that corresponds to a certain likelihood that a packet arrives on-time. This delay represents the latency caused by transmitting a data packet through a single hop local network (e.g., a ring topology) and is proportional to the number of data elements needed for buffering at the destination.

Table 1: Delay and Delay Estimator Symbols

D_v, D_p	queuing, propagation delays
D_t	transmission delay
$D^C_{v,i}$	cumulative variable delay for object i
T_i	control time for ith object
$T^C_{v,i}$	variable delay component of T
$\hat{\mu}, \hat{\sigma^2}$	current estimated mean, variance
$\hat{\mu}_s, \hat{\sigma^2}_s$	estimates used in current schedule
$\Delta\mu, \Delta\sigma^2$	max. difference between $\hat{\mu}, \hat{\mu}_s; \hat{\sigma^2}, \hat{\sigma^2}_s$

Figure 2 illustrates the selection of a control time T_1 which corresponds to the desired percentage of packets P_1 arriving on time. While such a delay function can accurately represent the delay characteristic over a given period of time, network loading can change, and a new delay function results. Our approach adapts to this change by monitoring channel delay distribution and adjusting the retrieval schedule accordingly.

The actual end-to-end delay of a single packet D_e can be described as:

$$D_e = D_p + D_t + D_v$$

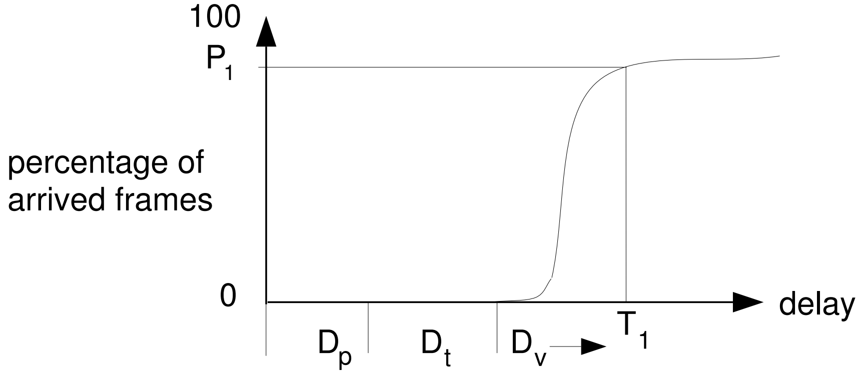

Figure 2: Probability Distribution Function for Delay

where D_p corresponds to constant overheads such as propagation delay, D_t is the transmission delay time, and D_v is the variable portion of the delay due to queuing delays in the network. If r is the number of communication packets which form an entire multimedia object, then the end-to-end transmission time of an object is described by [11]:

$$D_e = D_p + rD_t + D_v^C.$$

With the cumulative variable delay D_v^C defined by the equation $D_v^C = \sum_{j=1}^{r} D_{v_j}$ it can be modeled as the sum of r independent identically distributed random variables. To represent D_v^C, statistics are maintained for D_v of each arriving packet. These statistics are then used to create a probability distribution for the variable delay for retrieving one packet with a mean μ_{D_v} and a standard deviation $\sigma^2_{D_v}$. As shown in Figure 2, a delay time T_1 can be chosen from the delay distribution for D_v which corresponds to a desired probability of arrival P_1. A similar selection can be made from the delay distribution of D_v^C when retrieving a multipacket object.

Since D_v^C represents the sum of independent and identically distributed variable delays, then by the Central Limit Theorem the mean $\mu_{D_v^C}$ and standard deviation $\sigma^2_{D_v^C}$ can be approximated by

$$\mu_{D_v^C} = r\mu_{D_v}$$

and

$$\sigma^2_{D_v^C} = \sqrt{r}\sigma^2_{D_v}.$$

The Central Limit Theorem also states that as r increases the distribution of D_v^C becomes increasing similar to a normal distribution [18]. The difference between an exact distribution

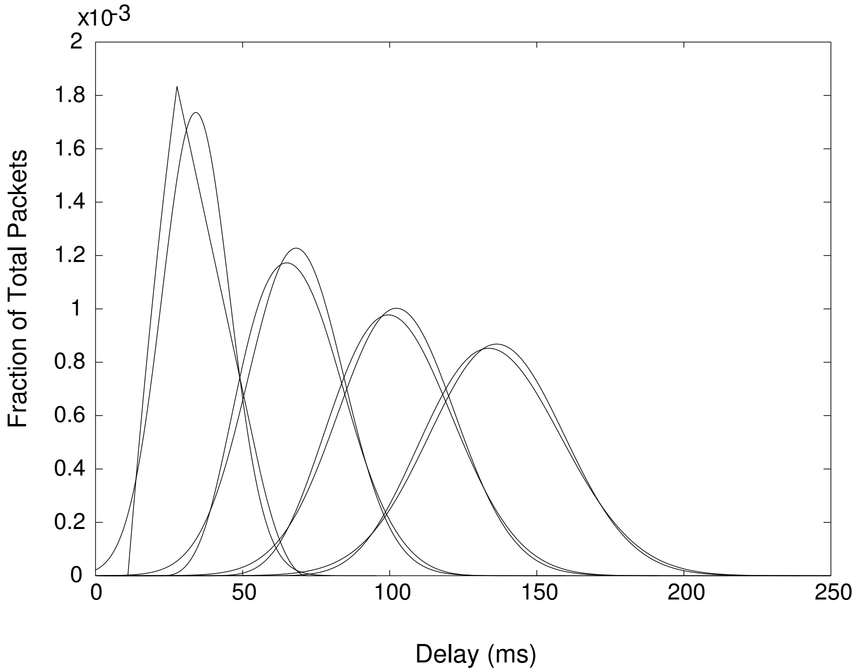

Figure 3: Comparison between the Estimated Normal Delay Distribution and Exact Delay Distribution

and its normal approximation is illustrated in Figure 3. The sets of curves as viewed from left to right represent the exact and approximate distributions of D_v^C for retrieving 1, 2, 3, and 4 consecutive packets from a local rotating storage device [2].

Because the number of communication packets r which comprise an object is generally large for multimedia data, we approximate the distribution of D_v^C as being normal. Having derived $\mu_{D_v^C}$ and $\sigma^2_{D_v^C}$ we can analyze D_v^C using the known qualities of a normal distribution. For example we can use the error function $erf(g)$, the desired likelihood of arrival (i.e., the percentage of frames arriving before their deadline), $\mu_{D_v^C}$, and $\sigma^2_{D_v^C}$ to calculate T_v^C the cumulative variable delay component of control time T (see Section 4.1). The control time T_i of object i can therefore be calculated by the equation:

$$T_i = D_p + rD_t + T_v^C.$$

From this mathematical basis, we now consider an approach to accommodating changes in system loading due to user interaction and multiple sessions.

3 Approximation of Network Load Dynamics

A limitation of static reservation approaches is their inability to adapt to system load changes or the dynamic behavior of an interactive multimedia session. Figure 4 shows a shift in retrieval delay distribution, illustrated as a density function, which is caused by a changing system load. Such changes must be considered when performing scheduling for a multimedia application which utilizes a network for data retrieval.

3.1 Delay Estimation

A change in the network loading is one of the conditions for creating a new schedule. The parameters used to estimate this change are the same parameters used to characterize the retrieval delay during the creation of a schedule. We approximate the variable delay distribution curve as normal (Section 3.1) with estimated mean and variance $\hat{\mu}$ and $\hat{\sigma^2}$, respectively. The maximum tolerable change in the parameters they are estimating are described by $\Delta\mu$ and $\Delta\sigma^2$. These values are chosen so that the schedule will not be adversely effected by the difference between the current values, as estimated by $\hat{\mu}$ and $\hat{\sigma^2}$, and their initial, exact values, over a period of maximum duration L. If the differences between the current estimated values and the values used in the current schedule are greater than $\Delta\mu$ or $\Delta\sigma^2$, then a new schedule must be created.

The accuracy of estimating these differences is contingent on the accuracy of the estimates for the mean and standard deviation themselves. The variables $\Delta\hat{\mu}$ and $\Delta\hat{\sigma^2}$ should be chosen so that they represent a change in the retrieval delay distribution and not merely an error in the approximation. The choice of $\Delta\hat{\mu}$ and $\Delta\hat{\sigma^2}$ is aided by the assumption that changes in retrieval delay distributions are discrete. These discrete changes occur when the number of user sessions being serviced by the retrieval mechanism is increased or decreased by one.

Several delay estimation techniques are possible. We review two of these approaches, including the simple window estimation method and the geometric estimation method. We use the latter method in our scheduler. As compared to the window method this technique is computationally less expensive and converges on new values geometrically instead of linearly.

3.2 Window Estimation Method

One estimation technique for the mean and variance of the variable delay distribution relies on maintaining a sliding window of the retrieval times of the last n packets that have arrived from a source. This can be implemented based on the following analysis.

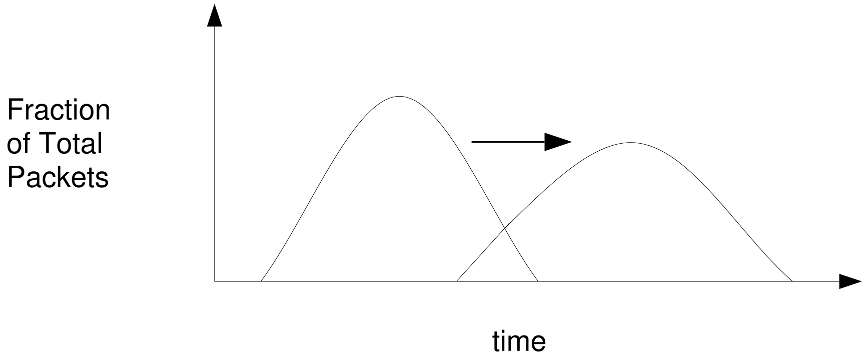

Figure 4: Change in Retrieval Delay Distribution

If $(X_1, X_2, X_3 ... X_n)$ are n independent and identically distributed samples from a variable delay distribution then $\hat{\mu}'$ is an unbiased and consistent estimator of mean μ if it is defined as [18]:

$$\hat{\mu}' \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} X_i .$$

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

$$\hat{\sigma^2}' \overset{\text{def}}{=} \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{\mu}')^2 .$$

The samples used are derived from the last n arrivals from the given source and each sample is weighted equally when the estimation is calculated.

3.3 Geometric Estimator

The system we employ to estimate the delay distribution is a another simple dynamic filter. It has the advantage of being easy to implement and having well defined performance parameters such as error estimates. Unlike the linear window estimator it estimates the mean and standard deviation using a weighted average of all packets that have arrived from a given source. This allows for geometric instead of linear convergence on the new values and a smaller amount of computation needed for calculating the new estimated parameter after each packet arrival.

If we use $\hat{\mu}$ as an estimate of the mean μ, then the equation for $\hat{\mu}$ is

$$\hat{\mu}_{i+1} = (1 - \gamma) * \hat{\mu}_i + \gamma * X_i$$

where γ is the correction factor. The larger the correction factor γ, the more responsive it is to a changing delay distribution. A large γ combined with a large variance means that the estimated parameter can vary significantly from the actual parameter and an appropriate $\Delta\mu$ or $\Delta\sigma^2$ must be chosen. The value of γ can be a function of time if packets do not arrive in regular intervals.

The estimation for σ^2 can be derived by maintaining estimates for X^2 since $\sigma^2 = E(X^2) - \mu^2$:

$$\hat{X^2}_{i+1} = (1 - \gamma) * \hat{X^2}_i + \gamma * X_i^2 \;\; ; \;\; \hat{\sigma^2}_{i+1} = \hat{X^2}_{i+1} - (\hat{\mu}_i)^2.$$

3.4 Estimator Analysis

In this section the geometric estimator is analyzed by demonstrating the cost of selecting different correction factors (γ). Figure 5 shows the convergence of the estimated mean from the initial mean of 5 ms to a new mean of 10 ms. The slowest case to converge is for the estimation algorithm with $\gamma = 0.2$. The fastest to converge uses a $\gamma = 1.0$. The remaining cases represent intermediate values of γ in increments of 0.2. Notice that there are variations in the estimated value once the average reaches 10 ms since the new arriving samples are normally distributed. The actual convergence time depends on the size of the communication packets which are assumed to be constant.

While a larger correction factor creates a quicker convergence when the estimated value changes, there is a disadvantage to using a large γ. For a stationary mean the average absolute difference between the estimated and actual mean increases linearly as the value of γ increases. This difference starts at 0 for $\gamma = 0$ and increases to 0.8 for $\gamma = 1.0$.

Both of the aforementioned estimation techniques are first order approximations of adaptive filters. It is possible to use a more sophisticated approach, but with limited return due to increased implementational complexity.

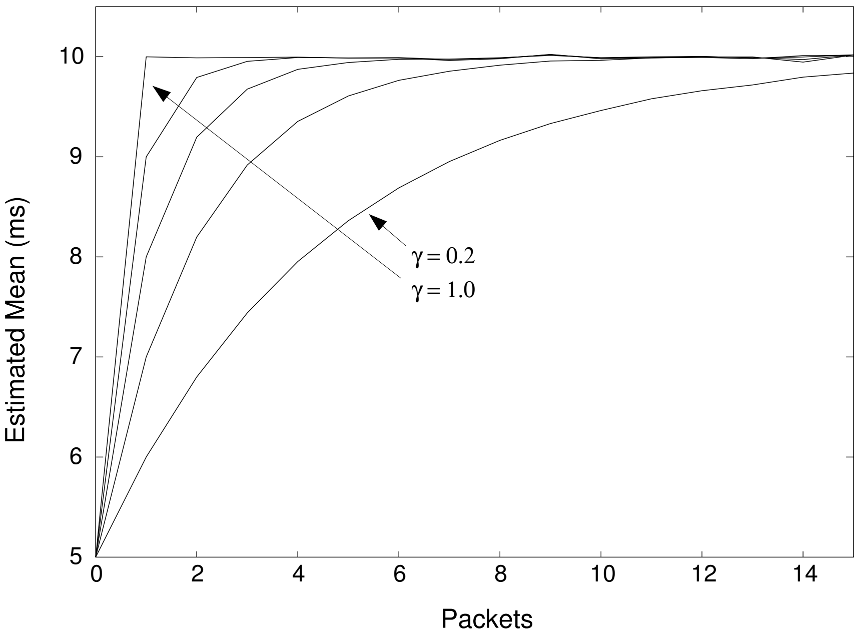

Figure 5: Convergence of the Estimated Mean for Different Values of γ When the Mean Changes from 5 to 10 ms

4 Limited A Priori Scheduler

In this section we describe our scheduling mechanism which provides a balance between static and dynamic scheduling for multimedia object retrieval, transmission, and playout. We call the approach *limited a priori* (LAP) scheduling, being based on a static, *a priori* scheduling approach [11], but supporting dynamic user input and system load changes by periodic schedule recomputation. The LAP scheduling framework is comprised of a static resource reservation mechanism, and a dynamic, run-time executor of the LAP-produced schedule. We call these components the *schedule creator* and the *schedule executor*, respectively. The essence of the LAP scheduling is as follows [13]. A multimedia session is decomposed into periods of similar resource utilization that can be scheduled independently (Figure 6). Each period is allocated resources and scheduled by the schedule creator using statistical resource reservation. The resultant schedule is then executed, along with other similar schedules, by the schedule executor.

Our multimedia scheduler creates a schedule for one portion, or period, of a session at a time. A new period is enacted either at the end of the current period or any time when required by a system change. System changes can be caused by a user request or a

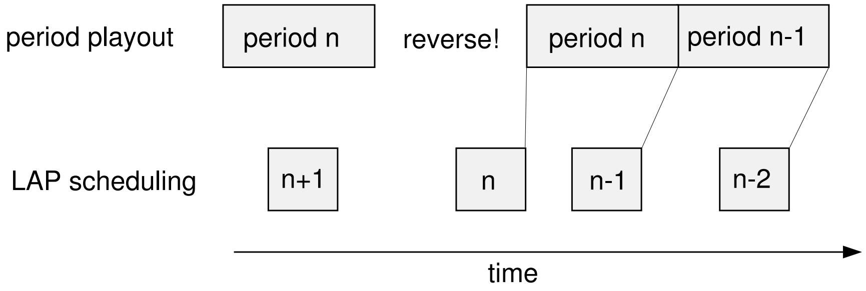

Figure 6: LAP Schedule Creation for Multiple Periods

significant variation in the communication channel load. In Figure 6 period n is recalculated and enacted due to a user request for a reverse playout and schedule $n-1$ is recalculated so that it can be enacted at the end of period n.

The length of a period L should be long enough to take advantage of bandwidth averaging, but short enough so that processor resources are not wasted calculating large portions of a schedule that go unused. A short schedule also allows for quicker schedule creation and therefore shorter response time to user interaction. The characteristics of a given period L can be found by assuming that the time between system changes occur with a normal distribution described by a mean μ_{sc} and a standard deviation σ_{sc}. The fraction of periods that are played to completion is described by $0.5 + erf(\frac{L-\mu_{sc}}{\sigma_{sc}})$. The average portion of a schedule remaining when interrupted by another schedule is $\frac{L-\mu_{sc}}{L}$. Therefore L can be chosen considering the penalty of a period being interrupted too early or being played in its entirety too frequently in accordance with a particular session and system.

Figure 7 shows the relationships between the components of the LAP scheduler. The schedules created by the schedule creator are enacted by the schedule executor.

Before a session begins, the initialization component of the LAP scheduler makes sure that the network parameter estimates have initial values. If there has been a recent session which has received packets from the same source then estimates from that session are used. If the source is new or not recently used, then sample packets are requested so that these parameter estimates can be created.

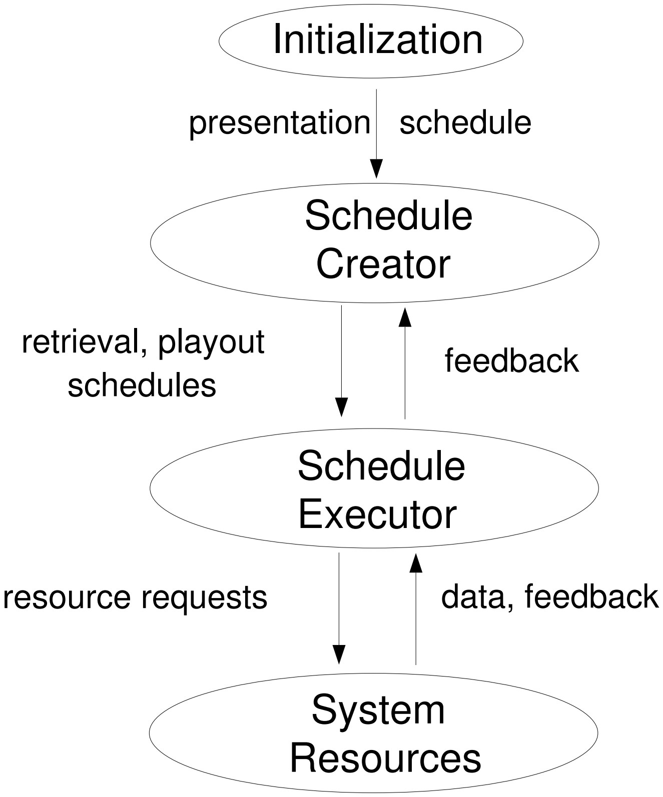

Figure 7: System Components of the LAP Scheduler

4.1 Schedule Creator

The schedule creator produces a retrieval schedule for a portion or period of a presentation schedule L seconds long. First it calculates the amount of total variable delay time T_v^C needed so that a given object i will be retrieved on time with probability F_m. This is done with the error function $erf(g)$ and estimates $\hat{\mu}$ and $\hat{\sigma^2}$ (see Section 3.1). With T_v^C established the total retrieval time T is calculated for the objects being retrieved.

Table 2: Symbols Used in LAP Scheduling Mechanism

ϕ_i	object i retrieval time
π_i	object i playout time
S_m	packet size for medium m (fixed)
$\vert x_i \vert$	size in bits of a given object i
C	channel capacity
R	destination-to-source request time
L	maximum period length
S	start time of the current schedule
H	schedule creation time
P	source request processing time
t	current time
F_m	fraction of on-time media arrival
g_m	input value for error function

The estimated values used for the new schedule being created are recorded as $\hat{\mu}_s$, $\hat{\sigma^2}_s$. They are used in the schedule executor to determine when the estimates used for this schedule are out-of-date (see Section 4.2).

If the time needed to retrieve all of the objects for a given period is longer than the period itself, then some objects must be dropped to make the presentation feasible given the current network loading. A graceful degradation algorithm would select the objects to be dropped so as to minimally impact the quality of the presentation.

Once a retrieval schedule is created at the destination the entire list of requests must be sent to the source so that the requested objects can be sent from the source at the scheduled time. In this version of the algorithm we are involving only a single medium arriving from a single source. However, the algorithm is general and can be applied to the multiple source, multiple media case

4.2 Schedule Executor

The scheduler executor enacts the schedule and maintains the retrieval channel performance estimates. When a new packet arrives the estimates for $\hat{\mu}$ and $\hat{\sigma^2}$ are updated (see Section 3.1) and the packet is routed to the correct playout subsystem. If the difference between these estimates and the ones used during the current schedule exceed a set limit ($\Delta\mu$, $\Delta\sigma^2$), or if the remaining active time for the current schedule is less than or equal to the time needed to create, request, and process a new schedule, then the schedule executor requests a new schedule from the schedule creator.

Schedule Creator Algorithm

```
1. Create schedule for next $L$ seconds
    (a) start with last object $n$, where $\Phi_n = T_c + S + L$
    (b) find $g_m$ for $F_m - 0.5$ using $erf(g)$ table
    (c) for i = 1 to n (objects in period)
        • find packets in object: $r_i = |x_i|/S_m$
        • find cumulative delay: $T^C_{v,i} = r_i\mu + g_m\sqrt{r_i\sigma^2}$
        • find total delay time: $T_i = D_p + r_iD_t + T^C_{v,i}$
    (d) record estimates used: $\hat{\mu}_s = \hat{\mu}; \hat{\sigma^2}_s = \hat{\sigma^2}$
    (e) set retrieval time of last object: $\phi_n = \pi_n - T_n$
    (f) set retrieval time for 2nd to last object to first: for i = n - 1 to 1
        • if scheduled to use channel at same time as last object, then must reschedule retrieval in relation to last object retrieval (FC Constraint)
        • else just get it there before playout (MD Constraint)
2. If for this period's schedule demand exceeds available capacity invoke graceful degradation algorithm and recreate schedule
3. Send list of requests for this period to source
```

4.3 Multimedia Traffic on an FDDI Network

The scheduler was evaluated by testing its performance for video data delivery across a simulated local computer network. An FDDI network was simulated using Network II.5 software. The network assumed a ring topology with a data rate of 100 Mb/sec. Video traffic was approximated with a model described by Maglaris for a “talking head” scene, typical in video conferencing applications [15]. The resulting compressed data stream had an average rate of 3.9 Mb/s.

Schedule Executor Algorithm

```
1. if new packet $i$ arrives
    (a) if delay is less than recorded minimum ($D_{e,i} < D_e^{min}$)
        • record new minimum: $D_e^{min} = D_e$
        • set new $D_p$: $D_p = D_e^{min} - D_t$
    (b) determine variable delay for new packet: $D_{v,i} = D_{e,i} - D_p - D_t$
    (c) calculate new mean estimate:
        $\hat{\mu} = (1 - \gamma) * \hat{\mu} + \gamma * D_{v,i}$
    (d) calculate new $D_v^2$ estimate:
        $\hat{D_v^2} = (1 - \gamma) * \hat{D_v^2} + \gamma * D_v^2$
    (e) calculate new standard deviation estimate: $\hat{\sigma^2} = \hat{D_v^2} - (\hat{\mu})^2$
2. if variance in estimates exceed limit or current schedule about to expire:
    $\{\Delta\mu < |\hat{\mu} - \hat{\mu}_s|$ or $\Delta\sigma^2 < |\hat{\sigma^2} - \hat{\sigma^2}_s|$
    or $t - S >= L - (H + P + R)\}$
    then request next schedule from Schedule Creator
```

Statistics were maintained for the retrieval of stored video data when 5, 10, and 15 stations were using the network to send live video data. In our model video frames are transmitted as blocks of contiguous packets every 1/30 of a second. We assume live video involves the transmission of packets at a steady rate since in our model data packets are sent whenever adequate data has accumulated. The length of each packet is constant at the maximum FDDI packet size of 4,506 bytes which, when discounting overhead, leaves 4,475 bytes for application data [1].

Both the stationary and transient delay distribution cases were studied. In the stationary case a larger γ or correction factors caused a greater penalty. This is because the mean and

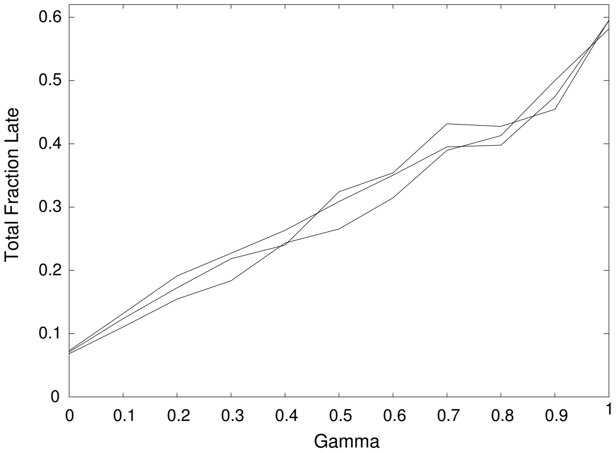

Figure 8: Fraction of Frames Which are Later Than Estimate

standard deviation estimators are more responsive to newly arriving packets and therefore are often inaccurate. As seen in Figure 8 a larger γ causes a larger percentage of late packets. When $\gamma = 0.2$ the 15 station case has the lowest fraction of late packets and the 5 station case has the highest fraction of late packets. The target percentage of late packets used in delay estimation for the LAP scheduler was 5%.

The transient case studied involved delay distribution changes caused by the number of stations transmitting live video on the network increasing from 5 to 15. In these circumstances a larger γ allows for the estimated mean to more quickly converge to the actual mean. Figure 9 shows the average mean after 1 through 15 packets have arrived from the new delay distribution. Similar to the case shown in Figure 5 the slowest converging curve is for $\gamma = 0.2$ and the quickest converging curve is for $\gamma = 1.0$. The remaining curves are for γ values increasing in increments of 0.2. Once the estimation is within range of the actual mean, 3.8 ms, the variation is due to the random nature of packet arrival times.

With these figures the penalties incurred when utilizing different correction factors for both the transient and stationary case can be considered when picking the appropriate γ for a specific multimedia application. In this manner our scheduler will adapt to changing loads and dynamic user input and can be successfully implemented for applications which retrieve multimedia data across networks such as an FDDI network.

5 Conclusion

To satisfy the data delivery requirements of a general purpose network multimedia information system, mechanisms are required to guarantee timely and predictable service. In this paper we have proposed a mechanism for this purpose that is designed to adapt to network load changes and dynamic user interaction.

Our limited *a priori* scheduler maintains synchronization by creating a schedule in advance of playout for a portion or period of a presentation, allowing a static scheduling algorithm to be implemented which takes advantage of bandwidth averaging. Changes in channel

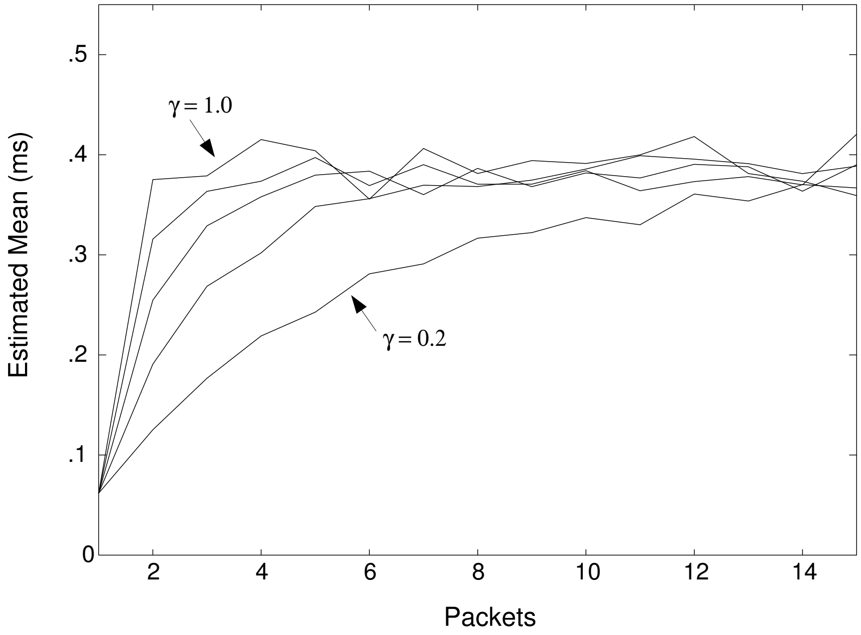

Figure 9: Average Estimated Mean After Receiving Packets from the New Delay Distribution

delay and bandwidth characteristics are accommodated by the scheduler when detected by an adaptive channel delay estimator. Results of our investigation indicate a number of potential estimation techniques with varying accuracy and complexity. Analysis indicates that a geometric aging function is adequate for our LAP mechanism and that our scheduler is applicable for multimedia data delivery across an FDDI network.

Acknowledgements

We would like to thank Azer Bestavros, David Castanon, and Mari Ostendorf for their helpful discussions.

References

[1] ANSI Std. X3.139-1987, Fiber Distributed Data Interface - Token Ring Media Access Control, Approved November. 5, 1986.

[2] H.-J. Chen and T.D.C. Little, "Physical Storage Organizations for Time-Dependent Multimedia Data," to appear at the 4th Intl. Conf. on Foundations of Data Organization and Algorithms (FODO'93), Evanston, IL, October 1993.

[3] Cheng, S., J. A. Stankovic, and K. Ramamritham, "Scheduling Algorithms for Hard-Real Time Systems – A Brief Survey," *Hard Real-Time Systems*, J. A. Stankovic and

K. Ramamritham, Ed., IEEE Computer Society Press, Washington D.C., 1988, pp. 150-173.

[4] Ferrari, D., “Design and Application of a Delay Jitter Control Scheme for Packet-Switching Internetworks,” *Proc. 2nd Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Heidelberg, Germany, November 1991.

[5] Gilge, M. and R. Gusella, “Motion Video Coding for Packet-Switching Networks - An Integrated Approach,”*SPIE Conf. on Visual Comm. and Image Processing*, Boston, November 1991.

[6] 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 Int’l Workshop on Network and OS Support For Digital Audio and Video*, San Diego, CA, November 1992.

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

[8] 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.

[9] Lazar, A.A., A. Temple, and R. Gidron, “An Architecture for Integrated Networks that Guarantees Quality of Service,”*Int’l. J. of Digital and Analog Cabled Systems*, Vol. 3, No. 2, 1990.

[10] 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.

[11] Little, T.D.C. and A. Ghafoor, “Scheduling of Bandwidth-Constrained Multimedia Traffic,” *Computer Comm.* (Special Issue: Multimedia Comm.), Vol. 15, No. 6, pp. 381-387, July/August 1992.

[12] Little, T.D.C., and F. Kao, “An Intermedia Skew Control System for Multimedia Data Presentation,” *Proc.* 3*nd International Workshop on Network and OS Support for Digital Audio and Video*, San Diego, CA, November 1992.

[13] Little, T.D.C. and J.F. Gibbon, “Real-Time Scheduling for Multimedia Services,” *Proc. SPIE Symp. OE/FIBERS ’92*, Boston, MA, September 1992.

[14] Lougher, P. and D. Shepherd, “The Design and Implementation of a Continuous Media Storage Server,” *Proc. 3rd Int’l. Workshop on Network and OS Support For Digital Audio and Video*, San Diego, CA, November 1992.

[15] Maglaris, B., D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins, “Performance Models of Statistical Multiplexing in Packet Video Comm.,” *IEEE Trans. on Comm.* Vol. 36, No. 7, pp. 834-843, July 1988.

[16] Nomura, M., T. Fujii, and N. Ohta, “Basic Characteristics of Variable Rate Video Coding in ATM Environment,” *IEEE J. on Sel. Areas in Comm.*, Vol. 7, No. 5, June 1989, pp. 752-760.

[17] Ramanathan, S. and P.V. Rangan, "Continuous Media Synchronization in Distributed Multimedia Systems," *Proc. 3rd Int'l. Workshop on Network and OS Support For Digital Audio and Video*, San Diego, CA, November 1992.

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

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

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