A Statistical Admission Control Algorithm for Multimedia Servers

Harrick M. Vin, **Pawan Goyal**, **Alok Goyal**, and **Anshuman Goyal**

Department of Computer Sciences, University of Texas at Austin
Taylor Hall 2.124, Austin, Texas 78712-1188
E-mail: {vin,pawang,alok,anshu}@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

Abstract

A large-scale multimedia server, in practice, has to service a large number of clients simultaneously. Given the real-time requirements of each client and the fixed data transfer bandwidth of disks, a multimedia server must employ admission control algorithms to decide whether a new client can be admitted for service without violating the requirements of the clients already being serviced. In this paper, we present an admission control algorithm for multimedia servers which: (1) exploits the variation in access times of media blocks from disk as well as the variation in client load induced by variable rate compression schemes, and (2) provides statistical service guarantees to each client. The effectiveness of the algorithm is demonstrated through trace-driven simulations.

1 Introduction

1.1 Motivation

Recent advances in computing and communication technologies have made it feasible as well as economically viable to provide on-line access to a variety of information sources (such as reference books, journals, newspapers, images, video clips, scientific data, etc.) over high speed networks. The realization of such information management systems of the future, however, will require the development of high performance, scalable multimedia servers which can provide a wide range of services to a large number of clients [4]. The fundamental problem in developing such multimedia servers is that images, audio, video, and other similar forms of data differ from numeric data and text in their characteristics, and hence require totally different techniques for their organization and management.

The most critical of these characteristics is that digital audio and video streams consist of a sequence of media quanta, such as video frames or audio samples, which convey meaning only when presented continuously in time (unlike text in which spatial continuity is sufficient). Hence, a multimedia server must ensure that recording and retrieval of media streams to and from disks proceed at their real-time rates. The development of techniques that enable a multimedia server to provide real-time performance guarantees to a large number of clients simultaneously is the subject matter of this paper.

1.2 Relation to Previous Work

Digitization of audio yields a sequence of samples, and that of video yields a sequence of frames. We refer to a continuously recorded sequence of audio samples or video frames as a *strand*. A multimedia server can organize the storage of such media strands on disk in terms of fixed size *media blocks*. Due to the periodic nature of media playback, a multimedia server can service multiple clients simultaneously by proceeding in *rounds*, retrieving a fixed number of media blocks for each client during each round. The number of blocks of a media strand retrieved during a round is dependent on its playback rate requirement, as well as the buffer space availability at the client [10]. Ensuring continuous retrieval of each strand requires that the *service time* (i.e., the total time spent in retrieving media blocks during a round) does not exceed the minimum of the playback durations of the blocks retrieved for each strand during a round. Hence, before admitting a new client, a multimedia server must employ admission control algorithms to decide whether a new client can be admitted without violating the continuous playback requirements of the clients already being serviced.

The precise formulation of the admission control algorithm is dependent on the quality of service requirements of the clients. If the entire clientele of a multimedia server desires deterministic service guarantees (i.e., their continuous playback requirements should never be violated for the entire service duration), the corresponding admission control algorithm will be characterized by worst-case assumptions regarding the service time [1, 2, 6, 8, 10, 11]. Notice, however, that due to the human perceptual tolerances as well as the inherent redundancy in continuous media streams, most clients of a multimedia server are tolerant to brief distortions in playback continuity as well as occasional loss of media information. Therefore, providing deterministic service guarantees to all the clients is superfluous. Furthermore, the worst-case assumptions that characterize deterministic admission control algorithms needlessly constrain the number of clients that can be serviced simultaneously, and hence, lead to severe under-utilization of server resources.

As a first step towards exploiting the human perceptual tolerances as well as the differences between the average and the worst-case performance characteristics of a multimedia server, we have recently proposed an *observation-based admission control algorithm*, in which a new client is admitted for service only if the prediction from the status quo measurements of the server performance characteristics indicate that the service requirements of all the clients can be met satisfactorily [9]. It is based on the assumption that the average amount of time spent in retrieving a media block from disk does not change significantly even after a new client is admitted by the server. In fact, it uses the average access time of a media block from the server and its standard deviation observed over a finite period to predict the access times of media blocks in future rounds. A multimedia server that employs such an observation-based approach is referred to as providing *predictive* service guarantees to clients.

1.3 Research Contributions of This Paper

The deterministic algorithms (which provide strict performance guarantees) and the observation-based algorithms (which offer a fairly reliable service, but no absolute guarantees) define two ends of the spectrum. In this paper, we analyze the continuum between these two extremes and develop admission control algorithms which use *distributions* (rather than worst-case and average-case values) of: (1) access times of media blocks from disk, and (2) playback rate requirements of media strands encoded using variable bit rate compression techniques; and provide *statistical* service guarantees to each client (i.e., the continuity requirements of at least a fixed percentage of media units is ensured to be met). The proposed algorithm improves the utilization of server resources by employing an aggressive admission control criteria: new clients are admitted for service as long as the *statistical estimation* of the aggregate data rate requirement (rather than the corresponding peak data rate requirement) can be met by the server. The statistical multiplexing of the server resources resulting from the admission of such clients, however, may occasionally lead to violation of the continuous playback requirements of some of the clients. To ensure that the statistical service guarantees being provided to the clients are not violated, we propose a technique for judiciously distributing such violations among multiple clients. Finally, to demonstrate the effectiveness of our statistical admission control algorithm, we have carried out extensive simulations. We present and analyze our simulation results.

The rest of the paper is organized as follows: In Section 2, we present the statistical admission control algorithm. Techniques for meeting the service requirements of clients are outlined in Sections 3. Our simulation results are described in Section 4, and finally, Section 5 summarizes our results.

2 Statistical Admission Control Algorithm

Consider a multimedia server that is servicing n clients, each retrieving a video strand (say $S_1, S_2, ..., S_n$, respectively). Let the service requirements of client i be specified as a percentage p_i of the total number of frames that must be retrieved on time. A multimedia server can service these clients by proceeding in periodic *rounds*, retrieving a fixed number of frames for each client during each round. Let $f_1, f_2, ..., f_n$ denote the number of frames of strands $S_1, S_2, ..., S_n$ retrieved during each round. Then, assuming that $\mathcal{R}_{pl}^i$ denotes the playback rate (expressed in terms of frames/sec) of strand S_i, the duration of a round, defined as the minimum of the playback durations of the frames accessed during a round, is given by:

$$\mathcal{R} = \min_{i \in [1,n]} \left(\frac{f_i}{\mathcal{R}_{pl}^i} \right)$$

In such a scenario, ensuring continuous playback of each media strand requires that the total time spent in retrieving media blocks from disk during each round (referred to as *service time* τ) should not exceed $\mathcal{R}$. The service time, however, is dependent on the number of media blocks being accessed as well as their relative placement on disk. Since each media strand may be encoded using a variable bit rate compression technique (e.g., JPEG, MPEG, etc.), the number of media blocks that contain f_i frames of strand S_i may vary from one round to another. This difference, when coupled with the variation in the relative separation between blocks, yields different service times across rounds. In fact, while servicing a large number of clients, the service time may occasionally exceed the round duration (i.e., $\tau > \mathcal{R}$). We refer to such rounds as *overflow* rounds. Given that each client may have requested a different quality of service (i.e., different values of p_i), meeting all of their service requirements will require the server to delay the retrieval of or discard (i.e., not retrieve) media blocks of some of the more tolerant clients during overflow rounds[1]. Consequently, to ensure that the statistical quality of service requirements of clients are not violated, a multimedia server must employ admission control algorithms that restrict the occurrence of such overflow rounds by limiting the number of clients admitted for service.

To precisely derive an admission control criterion that meets the above requirement, observe that for rounds in which $\tau \leq \mathcal{R}$, none of the media blocks need to be discarded. Therefore, the total number of frames retrieved during such rounds

[1] The choice between delaying or discarding media blocks during overflow rounds is application dependent. Since both of these policies are mathematically equivalent, in this paper, we will analyze only the discarding policy.

is given by $\sum_{i=1}^{n} f_i$. During overflow rounds, however, since a few media blocks may have to be discarded or delayed to yield $\tau \le \mathcal{R}$, the total number of frames retrieved will be smaller than $\sum_{i=1}^{n} f_i$. Given that p_i denotes the percentage of frames of strand S_i that must be retrieved on time to satisfy the service requirements of client i, the *average* number of frames that must be retrieved during each round is given by $p_i * f_i$. Hence, assuming that q denotes the overflow probability (i.e., $P(\tau > \mathcal{R}) = q$), the service requirements of the clients will be satisfied if:

$$q * \mathcal{F}_o + (1-q) \sum_{i=1}^{n} f_i \ge \sum_{i=1}^{n} p_i * f_i \qquad (1)$$

where $\mathcal{F}_o$ denotes the number of frames that are guaranteed to be retrieved during overflow rounds. The left hand side of Equation (1) represents the lower bound on the expected number of frames retrieved during a round and the right hand side denotes the average number of frames that must be accessed during each round so as to meet the service requirements of all clients. Clearly, the effectiveness of this admission control criteria, measured in terms of the number of clients that can be admitted, is dependent on the values of q and $\mathcal{F}_o$. In what follows, we present techniques for accurately determining their values.

2.1 Computing the Overflow Probability

While servicing multiple clients simultaneously, an overflow is said to occur when the service time exceeds the playback duration of a round. Whereas the playback duration $\mathcal{R}$ of a round is fixed (since the server is accessing a fixed number of frames for each client), the service time varies from round to round. Let the random variable τ_k denote the service time for accessing k media blocks from disk. Then overflow probability q can be computed as:

$$\begin{aligned} q = P(\tau > \mathcal{R}) &= \sum_{k=k_{min}}^{k_{max}} P(\tau > \mathcal{R} | \mathcal{B} = k) P(\mathcal{B} = k) \\ &= \sum_{k=k_{min}}^{k_{max}} P(\tau_k > \mathcal{R}) P(\mathcal{B} = k) \qquad (2) \end{aligned}$$

where $\mathcal{B}$ is the random variable representing the number of blocks to be retrieved in a round, and k_{min} and k_{max}, respectively, denote its minimum and maximum values. Hence, computing the overflow probability q requires the determination of probability distribution functions for τ_k and $\mathcal{B}$, as well as the values of k_{min} and k_{max}, techniques for which are described below.

- **Service time characterization**:

  Given the number of blocks to be accessed during a round, since the service time is dependent only on the relative placement of media blocks on disk and the disk scheduling algorithm, and is completely independent of the client characteristics, service time distributions are required to be computed only *once* during the lifetime of a multimedia server, possibly at the time of its installation.

  The server can derive a distribution function for τ_k by empirically measuring the variation in service times yielded by different placements of k blocks on disk. The larger the number of such measurements, the greater is the accuracy of the distribution function. Starting with the minimum number of blocks that are guaranteed to be accessed during a round (i.e., the value of k_d derived in Section 2.2), the procedure for determining the distribution function for τ_k should be repeated for $k = k_d, k_d + 1, ..., k_{end}$, where k_{end} is the minimum value of k for which $P(\tau_{k_{end}} > \mathcal{R}) \simeq 1$. Using these empirically derived distribution functions, the probability $P(\tau_k > \mathcal{R})$, for various values of k, can be easily computed.

- **Client load characterization**:

  Since f_i frames of strand S_i are retrieved during each round, the total number of blocks $\mathcal{B}$ required to be accessed is dependent on the frame size distributions for each strand. Specifically, if the random variable $\mathcal{B}_i$ denotes the number of media blocks that contain f_i frames of strand S_i, then the total number of blocks to be accessed during each round is given by:

$$\mathcal{B} = \sum_{i=1}^{n} \mathcal{B}_i$$

  Since $\mathcal{B}_i$ is only dependent on the frame size variations within strand S_i, $\mathcal{B}_i$'s denote a set of n *independent* random variables. Therefore, using the *central limit theorem*, we conclude that the distribution function $\mathcal{G}_{\mathcal{B}}(b)$ of $\mathcal{B}$ approaches a normal distribution [5]. Furthermore, if $\eta_{\mathcal{B}_i}$ and $\sigma_{\mathcal{B}_i}$ denote the mean and standard deviation of random variable $\mathcal{B}_i$, respectively, then the mean and standard deviation for $\mathcal{B}$ are given by:

$$\eta_{\mathcal{B}} = \sum_{i=1}^{n} \eta_{\mathcal{B}_i}, \quad \sigma_{\mathcal{B}}^2 = \sum_{i=1}^{n} \sigma_{\mathcal{B}_i}^2 \qquad (3)$$

  Consequently,

$$\mathcal{G}_{\mathcal{B}}(b) \simeq \mathcal{N}\left(\frac{b - \eta_{\mathcal{B}}}{\sigma_{\mathcal{B}}}\right) \qquad (4)$$

  where $\mathcal{N}$ is the standard normal distribution function. Additionally, since $\mathcal{B}_i$'s denote discrete random variables that take only integral values, they can be categorized as *lattice-type* random variables [5]. Hence, using the central limit theorem, the point probabilities $P(\mathcal{B} = k)$ can be derived as:

$$P(\mathcal{B} = k) \simeq \frac{1}{\sigma_{\mathcal{B}}\sqrt{2\pi}} e^{-\frac{(k-\eta_{\mathcal{B}})^2}{2\sigma_{\mathcal{B}}^2}} \qquad (5)$$

Finally, computing the overflow probability q using Equation (2) requires the values of k_{min} and k_{max}. If b_i^{min} and b_i^{max}, respectively, denote the minimum and the maximum number of media blocks that may contain f_i frames of strand S_i, then the values of k_{min} and k_{max} can be derived as:

$$k_{min} = \sum_{i=1}^{n} b_i^{min}; \quad k_{max} = \sum_{i=1}^{n} b_i^{max} \qquad (6)$$

Thus, by substituting the values of k_{min}, k_{max}, $P(\tau_k > \mathcal{R})$, and $P(\mathcal{B} = k)$ in Equation (2), the overflow probability q can be computed.

2.2 Determination of $\mathcal{F}_o$

The maximum number of frames $\mathcal{F}_o$ that are guaranteed to be retrieved during an overflow round is dependent on: (1) the number of media blocks that are guaranteed to be accessed from disk within the round duration $\mathcal{R}$, and (2) the relationship between the media block size and the maximum frame sizes.

To compute the number of media blocks that are guaranteed to be accessed during each round, worst-case assumptions (similar to those employed by deterministic admission control algorithms) regarding the access times of media blocks from disk may need to be employed. To illustrate the procedure, consider a multimedia server that employs the SCAN disk scheduling algorithm [7]. Let k denote the number of media blocks that are to be retrieved during a round. Since, in the worst-case, each media block may be placed on a different cylinder, the disk head may have to be repositioned onto a new cylinder at most k times. Furthermore, while accessing these blocks, the disk head may have to move from the inner-most cylinder to the outer-most cylinder, or vice versa. Hence, assuming that the disk contains C cylinders and the seek time incurred while moving the disk head from cylinder c_1 to c_2 is given by $l_{seek}(c_1, c_2) = a + b * |c_1 - c_2|$ where a and b are constants, the upper bound on the total seek time incurred during each round can be computed as: $(a * k + b * C)$. Similarly, assuming that the retrieval of each media block may, in the worst case, incur maximum rotational latency (denoted by l_{rot}^{max}), the total service time for each round can be computed as:

$$\tau = b * C + (a + l_{rot}^{max}) * k \qquad (7)$$

Since $\tau \leq \mathcal{R}$, the number of media blocks, k_d, that are guaranteed to be retrieved during each round is bounded by:

$$k_d \leq \frac{\mathcal{R} - b * C}{(a + l_{rot}^{max})} \qquad (8)$$

Now, assuming that $f(S_i)$ denotes the minimum number of frames that may be contained in a block of strand S_i, the lower bound on the number of frames accessed during an overflow round is given by:

$$\mathcal{F}_o = k_d * \min_{i \in [1,n]} f(S_i) \qquad (9)$$

2.3 Admitting a New Client

Consider the scenario that a multimedia server receives a new client request for the retrieval of strand S_{n+1}. In order to validate that the admission of the new client will not violate the service requirements of the clients already being serviced, the server must first compute the overflow probability assuming that the new client has been admitted. In order to do so, the server must determine:

1. The mean and the standard deviation of the number of media blocks that may contain f_{n+1} frames of strand S_{n+1} (denoted by $\eta_{\mathcal{B}_{n+1}}$ and $\sigma_{\mathcal{B}_{n+1}}$, respectively), to be used in Equations (3) and (4);

2. The minimum and the maximum number of media blocks that may contain f_{n+1} frames of strand S_{n+1} (denoted by b_{n+1}^{min} and b_{n+1}^{max}, respectively), to be used in Equation (6); and

3. The minimum number of frames contained in a media block of strand S_{n+1} (denoted by $f(S_{n+1})$), to be used in Equation (9).

Since all of these parameters are dependent on the distribution of frame sizes in strand S_{n+1}, the server can simplify the processing requirements at the time of admission by precomputing these parameters while storing the media strand on disk.

These values, when coupled with the corresponding values for all the clients already being serviced as well as the predetermined service time distributions will yield new values for q and $\mathcal{F}_o$. The new client is then admitted for service if the newly derived values for q and $\mathcal{F}_o$ satisfy the admission control criteria:

$$q * \mathcal{F}_o + (1 - q) \sum_{i=1}^{n+1} f_i \geq \sum_{i=1}^{n+1} p_i * f_i$$

3 Enforcing Statistical Service Guarantees

Meeting the admission control criteria (i.e., Equation 1) ensures that the lower bound on the expected number of frames retrieved during a round is at least as large as the average number of frames that must be accessed during each round to meet the service requirements of all clients. Stated differently, the total number of frames discarded by the multimedia server during overflow rounds will not exceed the cumulative loss tolerance of all the clients. However, to ensure that the individual service requirements of the clients are not violated, the server must judiciously distribute the discarded frames among all the clients. The sequential nature of video playback facilitates the prediction of the set of blocks to be accessed in a round prior to its initiation, and thereby enables a server to employ various policies for discarding media blocks during overflow rounds.

To precisely formulate the selection criteria, let us assume that $\mathcal{T}_i$ denotes the entire playback duration of strand S_i. Since $\mathcal{R}_{pl}^i$ and p_i denote the playback rate of strand S_i (expressed in frames/sec) and the service requirement of the client i, respectively, the number of frames of strand S_i which can be discarded over the entire playback duration, without violating the requirements of client i, is bounded by:

$$\mathcal{F}_i = \lfloor (1 - p_i) * \mathcal{R}_{pl}^i * \mathcal{T}_i \rfloor \quad (10)$$

Since $\mathcal{R}$ denotes the duration of a round, the retrieval of strand S_i from disk will be spread across $r = \lceil \frac{\mathcal{T}_i}{\mathcal{R}} \rceil$ rounds. In order to achieve an equitable distribution of the discarded frames throughout the playback duration of a strand, and hence to minimize the perceptual loss of media information, we define *loss affordability* of client i during round j (denoted by $\chi_{i,j}$) as $\chi_{i,j} = L_{i,j} - l_{i,j}$, where:

$$L_{i,j} = \lfloor j * (1 - p_i) * f_i \rfloor ; \quad l_{i,j} = \sum_{m=1}^{j-1} \widehat{l_i^m}$$

where $\widehat{l_i^m}$ denotes the number of frames of strand S_i discarded during the m^{th} round. The server can then select a set of media blocks to be discarded during an overflow round based on the relative values of $\chi_{i,j}$. Since $L_{i,j}$ gradually increases from one round to the next, employing such a policy will enable the server to disperse the frame losses throughout the playback duration of the strand.

To minimize the number of media blocks discarded during an overflow round, we now present a two-step algorithm for selecting a set of media blocks to be discarded during a round based both on the loss affordability of the clients and their relative placement on disk. In the first step, the server can compare the loss affordability of client i with the number of frames contained in each block of strand S_i to be retrieved during the round, and label them as either *can-be-discarded* or *can-not-be-discarded*. Clearly, this labeling procedure is dependent on the video compression scheme. For instance, if all the frames in a video strand are considered equally important (e.g., intra-coded frames in a JPEG video clip), then if the loss affordability $\chi_{i,j}$ is smaller than the number of frames contained in a media block of strand S_i, the block must be labeled as *can-not-be-discarded*. Otherwise, the block is labeled as *can-be-discarded*. On the other hand, for an MPEG-encoded video strand, since discarding an I-frame effectively eliminates all the succeeding P- and B-frames until the next I-frame can be accessed, the loss affordability must be compared with the *effective* frame loss, rather than just the number of frames contained in the block.

Once all the blocks have been labeled, the server must judiciously select and discard a subset of the *can-be-discarded* blocks. To minimize the number of frames discarded during this process, the selection criteria must be governed by the reduction in service time obtained by discarding a block as well as the number of frames contained in it. Specifically, if $\psi(z)$ and $f(z)$ denote the reduction in service time yielded by discarding media block z and the number of frames contained in block z, respectively, then a gain function $g(z)$ can be defined as:

$$g(z) = \frac{\psi(z)}{f(z)}$$

The server can then minimize the cumulative frame loss by discarding *can-be-discarded* media blocks in the decreasing order of $g(z)$.

Regardless of the labeling algorithm as well as the selection policy, once a media block to be discarded is determined, the multimedia server can recompute the service time τ_{new}. If $\tau_{new} > \mathcal{R}$, then the number of media blocks discarded need to be progressively increased until the new retrieval sequence yields $\tau_{new} \leq \mathcal{R}$. If, even after discarding all the *can-be-discarded* media blocks, the service time continues to be greater than the duration of the round, then $L_{i,j}$ can be set to $\mathcal{F}_i$, and the above process can be repeated. Since the clients are admitted by the server only when the admission control criteria (Equation (1)) is satisfied, it is guaranteed that the total number of frames discarded will never exceed $\sum_{i=1}^{n} \mathcal{F}_i$. Hence, the server is guaranteed to find a subset of media blocks which when discarded yield sufficient reduction in service time, while ensuring that the service requirements of the clients are not violated.

4 Experimental Evaluation

In this section, we demonstrate the viability of the statistical admission control algorithm through trace-driven simulations. The simulations were carried out in an environment consisting of a synchronous disk array with 16 disks. The characteristics of each disk are shown in Table 1. For the purposes of the simulations, each video strand is assumed to be encoded using a Variable Bit Rate (VBR) compression technique, and striped across the entire disk array. Successive blocks of a strand are assumed to be stored on the disk using the random placement model [3]. A *greedy* disk scheduling algorithm, which derives a retrieval sequence of media blocks from disk so as to simultaneously minimize both seek time and rotational latency, was employed for the simulations [9]. Furthermore, the playback rate of each video strand is assumed to be 30 frames/sec. The trace data for frame size variation yielded by VBR encoding techniques was obtained from Bellcore, University of California at Berkeley, and Columbia University.

4.1 Computing the Overflow Probability

For a multimedia server servicing n clients, an overflow is said to occur when the service time τ exceeds the playback duration $\mathcal{R}$ of a round. For various values of $\mathcal{R}$, we have computed the overflow probability q by determining distribution functions for: (1) service time τ_k for various values of k, and (2) the number of blocks $\mathcal{B}$ accessed during a round.

To illustrate, consider the scenario in which a multimedia server is retrieving 30 frames for each client during each

Disk capacity	4 GBytes
Number of disks in the array	16
Number tracks per disk	1024
Disk block size	32 KBytes
Rate of disk rotation	3600 RPM
$l_{seek}(t_1, t_2)$	$4 + 0.02 * \|t_1 - t_2\|$ ms
Maximum seek time	24.48 ms
Maximum rotational latency	16.66 ms

Table 1 : Disk parameters assumed in the simulation

round. Since the playback rate for all the clients is assumed to be 30 frames/sec, we get $\mathcal{R} = 1$ sec. Using Equation (8), the minimum number of blocks that are guaranteed to be retrieved within time $\mathcal{R}$ can be derived as $k_d = 48$. Starting with $k = k_d$, the process of computing service time distributions was repeated, by incrementing the value of k by 1 for each iteration, until for a particular value of $k = k_{end}$, $P(\tau_{k_{end}} > \mathcal{R}) \simeq 1$. For our environment, this termination condition was satisfied for $k_{end} = 170$. Notice that since the service time distributions are required to be computed only once during the lifetime of a multimedia server, the complexity of the above procedure is not very critical. Furthermore, since the variation in service times for different values of k follow the same pattern, instead of generating service time distribution functions for all values of $k \in [k_d, k_{end}]$, standard statistical modeling techniques can be employed to characterize the observed values of τ_k as a well-known distribution function. For instance, assuming that media blocks of each strand have been placed on disk using the random placement model, it can be shown that the service time yielded by SCAN disk scheduling algorithm is characterized by a normal distribution function. Once such a distribution function is determined, it can be parameterized appropriately to obtain the service time distribution functions for various values of k.

In order to validate our hypothesis that the total number of media blocks accessed during a round is normally distributed, we experimentally measured the variation in the number of blocks accessed during a round for different numbers of clients. Figure 1 shows one such distribution function obtained for 60 clients. As is evident from the figure, the empirically derived distribution function is closely approximated by the normal distribution.

Finally, using the distribution functions for service time and the number of blocks accessed during a round, we derived the overflow probability q (using Equation (2)), and studied its variation with increase in the number of clients n and the playback duration $\mathcal{R}$ (see Figure 2). As depicted in Figure 2, the value of q rises very slowly for small values of n, grows very rapidly over a small range of values of n, and finally,

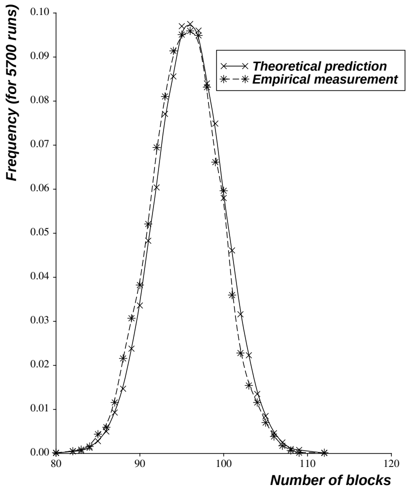


Figure 1 : Comparison of the empirically derived variation in the number of blocks accessed during each round for sixty clients and the theoretically derived normal distribution function

saturates at large values of n. This is because, for small n, the value of k_{max} derived using Equation (6) is sufficiently small such that the distribution function for $\tau_{k_{max}}$ is completely to the left of $\mathcal{R}$, and hence, $q \approx 0$. Similarly, for large values of n, the value of k_{min} derived using Equation (6) is sufficiently large such that the distribution function for $\tau_{k_{min}}$ is completely to the right of $\mathcal{R}$, yielding $q \approx 1$. For all the values of n between these extremes, the sudden rise in the value of q can be attributed to the bell-shaped curve for the distribution functions. Figure 2 also demonstrates that as the number of frames accessed for each client during a round increases (i.e. $\mathcal{R}$ increases), the server can accommodate a larger number of clients for the same value of q, albeit at the cost of increased latency. The rate of increase in the number of clients, however, decreases with increase in $\mathcal{R}$.

4.2 Evaluating the Statistical Admission Control Algorithm

We have compared the performance of our statistical admission control algorithm with conventional deterministic admission control algorithms [1, 2, 10, 11] (see Figure 3). As is depicted in Figure 3, the statistical admission control algorithm achieves a 200% increase in the number of clients that can be serviced simultaneously by the server.

Furthermore, to evaluate the effectiveness of the statistical admission control algorithm, we have compared its performance with theoretically derived as well as empirically observed bounds on the maximum number of clients that can be serviced simultaneously by a multimedia server. The theoretical upper bound, denoted by n_{max}, is defined as the ratio of the average data transfer bandwidth from the disk to the average data rate requirement of the clients. The empirically observed bound, denoted by n_{obs}, is derived by continuously

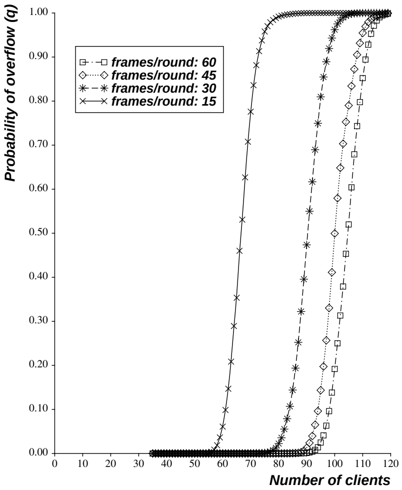


Figure 2 : Variation of the overflow probability with number of clients

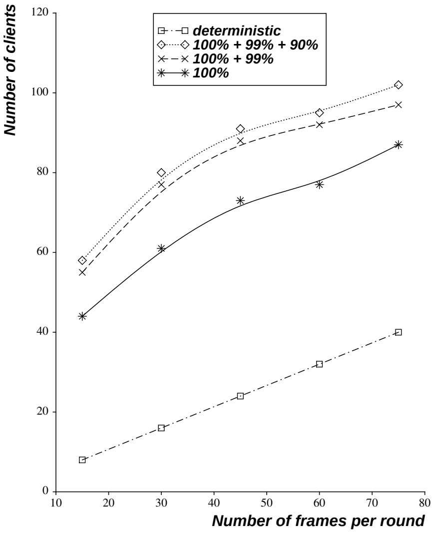


Figure 3 : Comparison of the statistical admission control algorithm with its deterministic counterpart

increasing the number of clients being serviced by the server until the service requirements of at least one of the clients are violated. Finally, n_{stat} denotes the maximum number of clients that can be admitted for service by the statistical admission control algorithm proposed in this paper. For various values of the desired quality of service (i.e., p_i), Figure 4 depicts the variation in n_{max}, n_{obs}, and n_{stat} with increase in round duration $\mathcal{R}$.

Figure 4 demonstrates that the number of clients admitted by the statistical admission control algorithm increases with decrease in p_i. However, the rate of increase in n_{stat} decreases with decrease in p_i. To explain this behavior, consider the statistical admission control criteria (i.e., Equation (1)). Since the value of $q * \mathcal{F}_o$ is likely to be very small as compared to the term $(1 - q) * \sum_{i=1}^{n} f_i$, the maximum permissible value of the overflow probability can be approximated as: $q \approx (1 - p_i)$. Thus, for the region of interest depicted in Figure 4 (namely, $0.9 \leq p_i \leq 1$), the overflow probability q must be bounded within $[0.0, 0.1]$. Within this region, relaxing the requirements of the clients from $p_i = 1.0$ to $p_i = 0.99$, and thereby increasing the maximum tolerable overflow probability from $q = 0.0$ to $q = 0.01$, enables the server to exploit the statistical variation in service times, and admit a much larger number of clients. However, as depicted in Figure 2, once the overflow probability is about 0.01, the server operates in a state where each increase in the number of clients results in a sharp increase in the value of q. Hence, relaxing the service requirements of the clients further, and thereby increasing the maximum tolerable value of overflow probability, does not yield the same rate of increase in n_{stat}. Figure 4 also demonstrates that, in the region of interest, the value of n_{max} increases linearly with the reduction in p_i. In comparison, the rate of increase in n_{stat} is higher when $(1 - p_i) \leq 0.3$, and lower for all higher values of $(1 - p_i)$. Hence, the difference between the values of n_{max} and n_{stat} decreases when p_i reduces from 1.0 to 0.97, (see Figures 4(a) and 4(b)), but starts to increase once $p_i < 0.97$ (see Figures 4(b) and 4(c)).

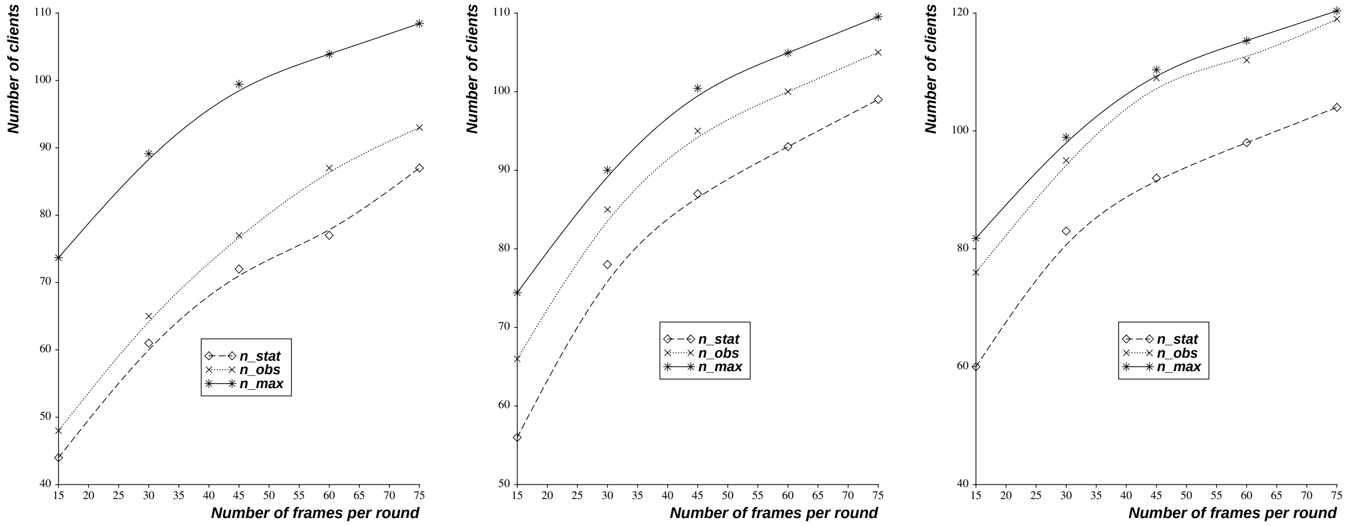
Figure 4 also demonstrates the values of n_{stat} supported by the statistical admission control algorithm are inherently more conservative than n_{obs}. This is because, for all n, $n_{stat} \leq n \leq n_{max}$, although the statistical analysis predicts that there exists a sequence of access patterns which may result in the violation of service requirements of some of the clients, it may not be encountered during the finite playback times of each media strand. Furthermore, in order to ensure that satisfying the admission control criteria (i.e., Equation (1)) is sufficient for meeting the statistical service requirements of all the clients, the value of $\mathcal{F}_o$ was derived by making worst-case assumptions (Section 2.2). In practice, however, employing the techniques for judiciously selecting a set of blocks to be discarded during overflow rounds (outlined in Section 3) may successfully retrieve a much larger number of frames as compared to $\mathcal{F}_o$. Hence, in practice, n_{obs} is ensured to be at least as large as n_{stat}. The difference between n_{obs} and n_{stat}, however, can be bridged by utilizing the available service time distributions and deriving $\mathcal{F}_o$ as:

$$\mathcal{F}_o = k_{stat} * \min_{i \in [1,n]} f(S_i)$$

where k_{stat} is the maximum value of k for which $P(\tau_k \leq \mathcal{R}) \simeq 1$.

5 Concluding Remarks

In this paper, we have presented a statistical admission control algorithm which improves the utilization of server resources by exploiting the variation in the access times of media blocks from disk, as well as the variation in playback rate requirement induced by variable rate compression techniques. The



(a) p_i = 1.0 (b) p_i = 0.99 (c) p_i = 0.90

Figure 4 : Variation in n_{max}, n_{obs}, and n_{stat} with increase in round time for different service requirements

main goals of our admission control algorithm are to: (1) accept enough traffic to efficiently utilize the server resources, while not accepting clients whose admission may lead to the violations of the service requirements of clients, and (2) provide statistical service guarantees to each client. We have demonstrated the effectiveness of the statistical admission control algorithm through extensive simulations. Our simulation results reveal that, as compared to its deterministic counterpart, the statistical admission algorithm achieves a 200% increase in the number of clients serviced simultaneously. A prototype multimedia server, based on the algorithms presented in this paper, is being implemented at the UT Austin Distributed Multimedia Computing Laboratory.

REFERENCES

[1] D. Anderson, Y. Osawa, and R. Govindan. A File System for Continuous Media. *ACM Transactions on Computer Systems*, 10(4):311–337, November 1992.

[2] J. Gemmell and S. Christodoulakis. Principles of Delay Sensitive Multimedia Data Storage and Retrieval. *ACM Transactions on Information Systems*, 10(1):51–90, 1992.

[3] M. K. McKusick, W. N. Joy, S. J. Leffler, and R. S. Fabry. A Fast File System for UNIX. *ACM Transactions on Computer Systems*, 2(3):181–197, August 1984.

[4] G. Miller, G. Baber, and M. Gilliland. News On-Demand for Multimedia Networks. In *Proceedings of ACM Multimedia'93, Anaheim, CA*, pages 383–392, August 1993.

[5] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 1991.

[6] A.L. Narasimha Reddy and J. Wyllie. Disk Scheduling in Multimedia I/O System. In *Proceedings of ACM Multimedia'93, Anaheim, CA*, pages 225–234, August 1993.

[7] T. Teorey and T. B. Pinkerton. A Comparative Analysis of Disk Scheduling Policies. *Communications of the ACM*, 15(3):177–184, March 1972.

[8] F.A. Tobagi, J. Pang, R. Baird, and M. Gang. Streaming RAID: A Disk Storage System for Video and Audio Files. In *Proceedings of ACM Multimedia'93, Anaheim, CA*, pages 393–400, August 1993.

[9] Harrick M. Vin, Alok Goyal, Anshuman Goyal, and Pawan Goyal. An Observation-Based Approach For Designing Multimedia Servers. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Boston, MA*, pages 234–243, May 1994.

[10] Harrick M. Vin and P. Venkat Rangan. Designing a Multi-User HDTV Storage Server. *IEEE Journal on Selected Areas in Communications*, 11(1):153–164, January 1993.

[11] P. Yu, M.S. Chen, and D.D. Kandlur. Design and Analysis of a Grouped Sweeping Scheme for Multimedia Storage Management. *Proceedings of Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego*, pages 38–49, November 1992.