# Efficient Admission Control Algorithms for Multimedia Servers [1]

Xiaoye Jiang and Prasant Mohapatra
Department of Electrical and Computer Engineering
201 Coover Hall
Iowa State University
Ames, Iowa 50011
E.mail: *prasant@iastate.edu*

**Abstract**

In this paper, we have proposed efficient admission control algorithms for multimedia storage servers that are providers of variable bit rate media streams. The proposed schemes are based on a slicing technique and use aggressive methods for admission control. We have developed two types of admission control schemes: *Future-Max* (FM) and *Interval Estimation* (IE). The FM algorithm uses the maximum bandwidth requirement of the future to estimate the bandwidth requirement. The IE algorithm defines a class of admission control schemes that use a combination of the maximum and average bandwidths within each interval to estimate the bandwidth requirement of the interval. The performance evaluations done through simulations show that the server utilization is improved by using the FM and IE algorithms. Furthermore, the quality of service is also improved by using the FM and IE algorithms. Several results depicting the trade-off between the implementation complexity, desired accuracy, the number of accepted requests, and the quality of service are presented.

**Key Words:** Admission Control, Future-Max Algorithm, Interval Estimation Algorithm, Multimedia Storage Server, Quality of Service.

[1]This research was supported in part by the National Science Foundation through the grants MIP-9628801 and CCR-9634547. A preliminary version of this paper appeared in the proceedings of the IEEE Conference on Multimedia Computing and Systems, 1997.

# 1 Introduction

Recent developments in computer systems and high speed networks have propelled the research on multimedia systems. A multimedia system requires the integration of communication, storage, retrieval, and presentation mechanisms for diverse data types including text, images, audio, and video to provide a single unified information system. The potential applications of multimedia systems span into domains such as computer-aided design, education, entertainment, information systems, and medical imaging. An efficient support mechanism for such a diverse class of application requires a suitable storage server connected to the clients through high speed networks [1]. Given a network set-up, the architecture and organization of the storage server has a significant impact on the service of multimedia clients. The design issues associated with the multimedia storage servers (MSS) differ from those associated with the services that support traditional textual and numeric data because of the difference in the characteristics of multimedia streams. A multimedia stream consists of a sequence of media quanta, such as audio samples and video frames, which convey meaning only when played continuously in time unlike the traditional textual streams [2].

An MSS should ensure that the retrieval of media streams occur at their real-time rate [3]. As the total bandwidth from the storage devices attached to the server via network to the clients is fixed, an MSS can only support a limited number of clients simultaneously [4]. Hence, before admitting a new request, an MSS must ensure that the real-time retrieval process of the existing clients (the streams that are currently being served) are not violated. The checking of this constraint and determination of the acceptance/rejection of a new request is done through the admission control algorithm employed in the MSS. The admission control algorithm checks if the available bandwidth is sufficient for the total bandwidth required by the streams currently being retrieved plus the bandwidth requirement of the new request. If it is sufficient, the server can accept the new request. Otherwise, the admission of the new request may introduce distortions or jitters in the audio or video quality [5, 6, 7, 8, 9, 10]. However, disturbances due to minor discontinuity of real-time playback may not be perceivable, and in some cases, acceptable at a lower cost by the clients. Based on the required *Quality of Service* (QoS), the admission control algorithm decides whether or not to accept the new request.

Several admission control schemes have been proposed in the literature. Detailed descriptions of some of these schemes are presented in Section 2. The goal of the admission

control schemes is to maximize the server utilization (by admitting as many requests as possible) while satisfying the QoS requirements. However, these two requirements are conflicting in nature. Most of the previously proposed schemes tend to sacrifice one in favor of the other. In this paper, we propose a set of aggressive admission control algorithms that maximize the server utilization as well as provide high QoS. Traditional admission control algorithms use statistical data of each stream, such as maximum consumption rate, average consumption rate, and distribution of consumption rate [2, 5, 6, 8, 9, 10, 11]. In the proposed approach, a complete profile of the media streams are computed while they are stored. The profile includes the consumption rate or the bandwidth requirements of the media stream. This information can be used by the server to reserve bandwidth and facilitate admission control. However, to reduce the computational overheads, the profiling is done by slicing the media streams into equal-sized time intervals. Granularity of these intervals affects the performance of the admission control schemes.

Two different type of admission control schemes, namely, *Future-Max* (FM) and *Interval Estimation* (IE) are developed in this work. In the FM algorithm, the maximum bandwidth requirement in future for a stream is used as its estimated bandwidth. For the family of IE algorithm, a combination of the maximum and average bandwidths is used for the bandwidth estimation. Different combinations of the maximum and average bandwidths result in different admission control schemes and yield different performance. The proposed admission control schemes are evaluated through simulation experiments. The performance improvement obtained using the FM algorithm increases upto a certain accuracy level and remains constant thereafter. With the IE algorithms, the performance improvement is almost linear. It is also observed that for a fixed number of clients, the QoS of the media streams improves with respect to the accuracy level. Several other performance results have been illustrated to demonstrate the validity of the proposed algorithm. The trade-off evaluation between the performance and implementation complexity is also reported. Based on the desired accuracy, QoS, and implementation simplicity, a suitable admission control scheme can be adopted from the family of algorithms proposed in this paper.

The rest of the paper is organized as follows. In Section 2, we review the requirement of admission control policy, and the advantages and disadvantages of the previously proposed admission control policies. In Section 3, we present the FM and IE algorithms and discuss the issues related to their implementation. In Section 4, we present the simulation results and discussions followed by the concluding remarks in Section 5.

# 2 Preliminaries

In this section, we classify and discuss the previously proposed admission control schemes. The characteristics of the media streams are also analyzed. The requirements of a good admission control scheme along with the limitations of the previously proposed schemes are also reported.

## 2.1 Classification of Admission Control Schemes

The admission control schemes proposed in the literature can be classified into four categories as follows [4].

- *Deterministic*: All the real time requirements are guaranteed to be met. With these algorithms, the server uses the worst case consumption rate to reserve bandwidth [2, 5, 6, 8, 9, 10]. The media streams have a perfect QoS while using the deterministic algorithm. Because of the pessimistic approach used for bandwidth reservation, the server utilization is usually low.

- *Predictive*: Server does the admission control based on the bandwidth requirement information measured during the last few time periods. The immediate past bandwidth requirements plus the average bandwidth requirement of the newly requested stream is assumed as an estimation for the future bandwidth requirement [12]. Although there is no guaranteed QoS, the server will accept a request for a media stream only if it predicts that all deadlines will be met satisfactorily in the future. It is observed through experimentation that the QoS does not degrades noticeably compared to the deterministic case.

- *Statistical*: The statistical admission control algorithms use probabilistic estimates to provide a guaranteed QoS to the accepted requests. The server considers the statistical behavior of the system in order to provide such guarantees [11]. One example is the use of average bandwidth as an estimation of the bandwidth requirement of the media streams in future.

- *Best Effort*: Normally, in real-time environment, the server provides the best effort service to non real-time requests. Thus the admission control algorithm does not guarantee any deadline to be met. The server will accept a request regardless of its

bandwidth requirement and does its best to serve. The QoS is usually low in this case, but the server utilization is high.

## 2.2 Characteristics of Multimedia Streams

The input data of an admission control algorithm includes the data associated with the server and the data associated with the stream. The data related to server refer to the total available bandwidth that can be supported by the system configuration. The media stream data refers to the stream bandwidth requirements which is determined by the consumption rate of the streams. The admission control algorithms use these two kinds of data to decide whether or not to admit a new media stream request. The total available bandwidth is a function of the hardware parameters and the disk scheduling method. The stream bandwidth requirements, although vary with respect to time, are fixed after the stream is stored.

If a media is a *Constant Bit Rate* (CBR) stream, its bandwidth requirement for the worst case and the average case will be the same. Due to the high bandwidth requirement of video and audio streams, it is not cost-effective to store and transmit them in their original formats. Usually, some kind of compression method is used to reduce their bandwidth requirements. The compressed streams are *Variable Bit Rate* (VBR) streams. Average bandwidth requirements of various CBR and VBR streams are shown in Table 1 [13]. The main characteristic of the VBR stream is that the consumption rate changes with time due to the different compression ratios of different segments of a stream. The worst case requires the maximum bandwidth which corresponds to the maximum consumption rate and the lowest compression ratio. The average case corresponds to the data related to the average consumption rate which relates to the average compression ratio. Since video/audio CBR streams require huge bandwidth, most of the media streams are stored as VBR streams. In this paper, we propose admission control scheme for MSSs that store and serve VBR media streams.

## 2.3 Requirements of Admission Control Schemes

The main function of an admission control algorithm is to reserve bandwidth corresponding to the requirements of a media stream at the admission time to guarantee the required QoS during playback. If the server can reserve the bandwidth for a request stream successfully, it accepts the request. Otherwise, it rejects the request. The total available

| Media Type (Specifications) | Data Rate |
|---|---|
| Voice quality audio (1 channel, 8 bit samples at 8 kHz) | 64 Kbits/sec |
| MPEG encoded audio, compressed VBR (equivalent to CD quality) | 384 Kbits/sec |
| CD quality audio (2 channels, 16 bit samples at 44.1 kHz) | 1.4 Mbits/sec |
| MPEG-2 encoded video, compressed VBR | 0.42 MBytes/sec |
| NTSC quality video (640 X 480 pixels, 24 bits/pixel) | 27 MBytes/sec |
| HDTV quality video (1280 X 720 pixels, 24 bits/pixel) | 81 MBytes/sec |

Table 1: Bandwidth Requirement for Typical Digital Multimedia Streams.

bandwidth is fixed. The more the bandwidth reserved for a specific stream, the less is the number of streams that a server can support simultaneously. Streams may have different bandwidth reservations due to the difference in the desired QoS or bandwidth estimation.

The server utilization of the deterministic admission control algorithms is much lower than the predictive or the statistical algorithms [14]. As a result, a server cannot support more streams in the deterministic case compared to the predictive or the statistical schemes. This is because of the fact that the total number of streams that the server can support simultaneously is proportional to the server utilization. The difference between the maximum and average consumption rate or the actual consumption rate degrades the server utilization for a given media stream. If the server uses the deterministic control policy [2, 5, 6, 8, 9, 10] to do the admission, then the maximum consumption rate is used for bandwidth reservation. However, a stream is not always in the state of its maximum consumption rate. For the non-peak periods, the bandwidth requirement of a stream is well below the bandwidth reserved for it. During these periods, the server utilization is low. The predictive or the statistical control policies use the consumption rate that is observed during a few past scheduling rounds [12], or corresponding to the distribution of the consumption rate [11]. Although these schemes have the possibility of missing deadlines of a stream as opposed to the deterministic scheme, they can support more streams than the deterministic approach. The server utilization is thus higher than the deterministic algorithm. In Figure 1, we show the typical bandwidth requirements of a VBR stream and compare it to the bandwidth reserved in deterministic, predictive and a statistical algo-

rithm (based on averaging). The graph explains the reason why the deterministic algorithm

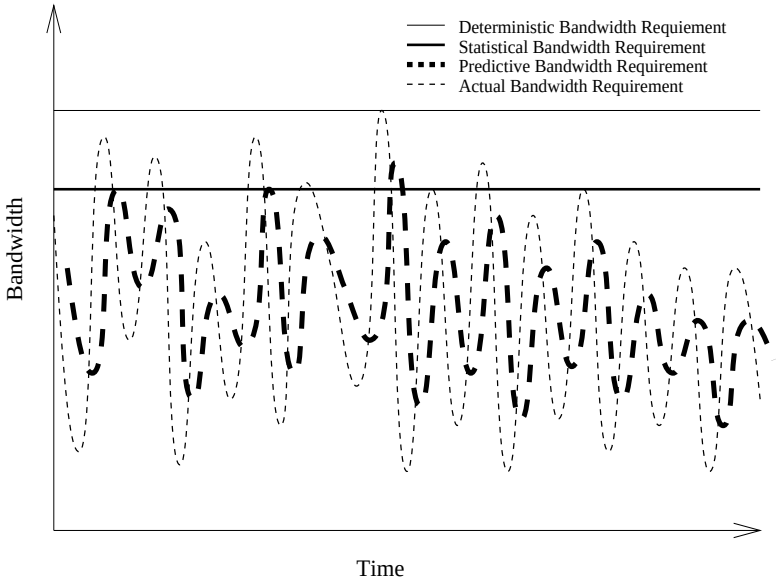


Figure 1: Reserved Bandwidth for Different Admission Policies.

has low server utilization. We introduce the notion of *Estimation Error* (EE) that defines the absolute difference between the estimated bandwidth, $BE(t)$(that is reserved), and the actual bandwidth, $A(t)$. Assuming $T$ as the total time of playback for a stream, $EE$ can be expressed as

$$EE = \int_0^T \mid BE(t) - A(t) \mid dt. \tag{1}$$

The $EE$ due to the over estimation of the bandwidth requirement is given as

$$EE^+ = \int_0^T (BE(t) - A(t)) * \theta(BE(t) - A(t))dt. \tag{2}$$

The $EE$ due to the under estimation of the bandwidth requirement is given as

$$EE^- = \int_0^T (A(t) - BE(t)) * \theta(A(t) - BE(t))dt, \tag{3}$$

where

$$\theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The $EE$ can then be computed as

$$EE = EE^+ + EE^-. \tag{5}$$

If $EE^+$ of an admission algorithm is much higher, its server utilization is much less than 1. $EE^-$ is related to the guaranteed QoS. If $EE^-$ is high, the guaranteed QoS is low. It can be observed that $EE^+_{deterministic} > EE^+_{predictive}$, and $EE^+_{deterministic} > EE^+_{statistical}$. Hence, the deterministic algorithms result in less server utilization than the predictive or the statistical algorithms. As the $EE^-_{deterministic} = 0$, the deterministic algorithms provide the highest QoS.

A good admission control algorithm should guarantee that the server meets the deadlines with respect to the specified QoS requirement. Furthermore, it should result in high server utilization which, in turn, enables the server to support more number of streams simultaneously. A good admission control algorithm should have less $EE^+$ and $EE^-$. In other words, it should have less $EE$. Thus, a good admission control algorithm should accurately model the system including the configuration, the scheduling method, and the characteristics of the stream.

The disadvantage of current admission control algorithms [2, 5, 6, 8, 9, 10, 11, 12] is that they only use a few statistical data to represent the server behavior and the media streams. This method facilitates the implementation, reduces the complexity of computation, and requires less storage space. However, the streams are not estimated accurately which may lead to poor server utilization. For example, current deterministic admission control algorithms [2, 5, 6, 8, 9, 10] use the maximum consumption rate of a whole stream to make the acceptance/rejection decision. This is a global value reflecting the behavior of the whole stream. Although the server should use the worst case of a stream and the server to do deterministic admission, it does not necessarily mean that the worst case of the server is when all the streams are at their consumption peak (worst case) because that may not happen at the same time. In fact, these peaks are more likely to be distributed uniformly. The current policies use this worst case scenario to employ admission control. In reality, there is a very small possibility that all the streams reach their consumption peak at the same time. Especially, when you have large number of streams being played, this probability is negligible. The predictive policy [12] uses the observed behavior of the system to estimate the future behavior. This method may not be an accurate model of system, although it improves the server utilization compared to the deterministic admission control scheme. The statistical policy [11] is complicated than the other two. It uses the distribution of bandwidth requirement to represent a stream. In fact, these values are also global parameters. The drawback of using the global values of a stream is that the local behavior of the streams are not captured. For example, the distribution of different

segments will be varied. So the distribution of the beginning may not be the same as the end. What we need is the local distribution of each stream at the same time points. These time points are not relative to the beginning of each stream. They refer to the time points or the snapshots at which the server plays back the media streams.

Inaccurate modeling of the server and/or the streams will degrade the performance of an admission control algorithm. The server should be modeled to obtain the total bandwidth limitation with respect to the hardware configuration and the scheduling method. For modeling the streams, we need to decrease the gap between $BE(t)$ and $A(t)$. In the next section, we present reasonably accurate modeling techniques of the media streams.

# 3 Slicing-Based Admission Control Algorithms

The responsibilities of a multimedia server include retrieval of media streams from the disks as well as recording of media streams on to the disks. The retrieval process is a real-time on-line service. Recording can be treated as an off-line service of the server, where the real-time constraints are imposed between the original recording device and the event being recorded. In this paper, we consider the on-line service issues and thus discuss only the retrieval process of media streams from the server.

In this paper, we do not consider the issues associated with disk scheduling algorithms. We assume that the server has a capacity of providing certain bandwidth. This bandwidth may be considered as the worst case bandwidth that can be provided by the disks and the server. Using efficient disk scheduling algorithms, the available bandwidth at the server may be improved. This improved bandwidth can be also used for the proposed algorithms. In other words, the proposed admission control schemes are not dependent on the disk scheduling algorithms. We just consider a fixed bandwidth that is guaranteed by the server at any time.

## 3.1 Slicing Method

A server needs to reserve an estimated bandwidth for the streams already in service before allowing a new request to be accepted. The reservation and checking of admissibility are handled by the admission control algorithm. For the actual bandwidth requirement, $A(t)$, of a stream, the bandwidth estimation, denoted as $BE(t)$, can be done in several ways. The deterministic algorithm uses a conservative estimate by considering the worst

case scenario. The bandwidth estimation, $BE_{det}$, for a stream using the deterministic admission control is determined as

$$BE_{det} = BE_{max} = \max\{A(t)\}, t \in [0, T], \tag{6}$$

where $T$ is the total time length of the stream. A statistical algorithm may use the average bandwidth requirement as its estimated bandwidth, $BE_{stat}$, which is derived from

$$BE_{stat} = BE_{ave} = \frac{\int_0^T A(t)dt}{T}. \tag{7}$$

The proposed approach relies on a closer estimation of $A(t)$. The retrieval of a media stream is done only after the completion of its storage. When a VBR media stream is stored, a complete and accurate description of the rate changes could be computed. This is the profile of bandwidth requirements of the VBR media stream. The server can use this information during playback for admission control. However, a trade-off analysis is essential for evaluating the increase in acceptance of the requests with respect to the additional overheads.

We introduce a method based on slicing to obtain the estimated bandwidth requirement, $BE(t_s)$, with respect to the slicing interval $t_s$. By using the slicing scheme, we divide $[0, T]$ into several small time intervals of the same size $t_s$(see Figure 2). The maximum value of $t_s$ can be the total time of playback of the stream ($T$). The minimum value of $t_s$ is one time unit, which is denoted as $t_{unit}$. The time unit can be the time length of a scheduling round or even smaller.

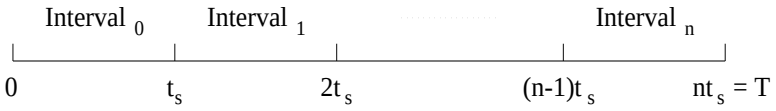


Figure 2: The Slicing Scheme.

The smaller the slicing interval, the more accurate will be the bandwidth estimation. With small slicing intervals, the $EE$ reduces. However, the implementation and computation complexity increases with the reduction of the size of the slicing intervals. Furthermore, the size reduction of intervals beyond a certain point may not have any impact on the performance improvement. These issues are addressed along with quantitative results in Section 4. We have expressed the granularity of intervals in terms of the accuracy level. A 100% accuracy level corresponds to the case when $t_s = t_{unit}$, and a 0% accuracy level corresponds to the case, $t_s = T$.

The bandwidth estimation based on the slicing method can be done in two different ways. The first method corresponds to the deterministic estimation and uses the maximum value within an interval as the estimated bandwidth requirement for the entire interval. The estimated bandwidth based on slicing for the $i$th interval, denoted as $BE_{smax}(i)$, is expressed as

$$BE_{smax}(i) = \max\{A(t)\}, t \in [it_s, (i+1)t_s], \forall i \in \{0, 1, \cdots, n-1\}, n = \frac{T}{t_s}. \quad (8)$$

The second method is a statistical scheme that uses the average of $A(t)$ within an interval as the estimated bandwidth. This bandwidth based on the slicing scheme for the $i$th interval is denoted as $BE_{save}(i)$ and is computed from

$$BE_{save}(i) = \frac{\int_{it_s}^{(i+1)t_s} A(t)dt}{t_s}, t \in [it_s, (i+1)t_s], \forall i \in \{0, 1, \cdots, n-1\}, n = \frac{T}{t_s}. \quad (9)$$

In Figure 3, we show a typical graph of $A(t), BE_{smax}(i)$ and $BE_{save}(i)$. It can be

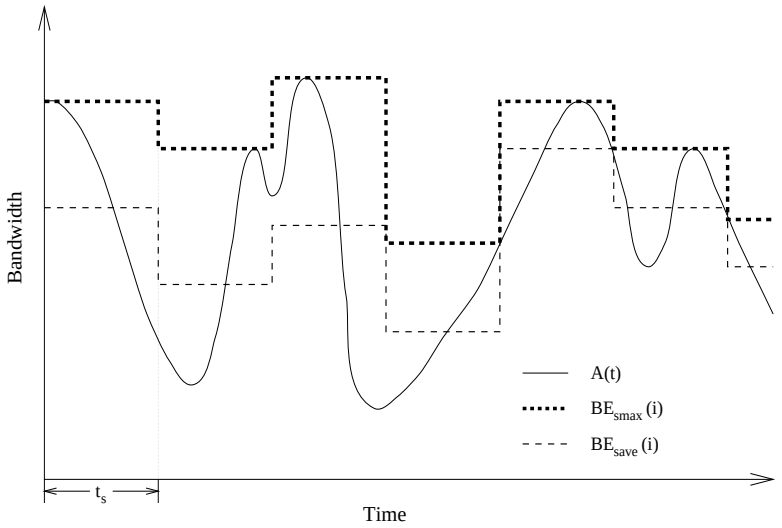


Figure 3: Difference between $A(t), BE_{smax}(i)$ and $BE_{save}(i)$.

observed from the Figure that as $t_s$ gets larger and larger, $BE_{smax}(i)$ will get close to $BE_{max}$. $BE_{save}(i)$ follows the same trend. For the extreme case, when $t_s = T$, $BE_{smax}(i)$ equals $BE_{max}$ and $BE_{save}(i)$ is $BE_{ave}$. These estimations have the lowest accuracy level. This scenario can be expressed as

$$BE_{smax}(i) = BE_{max}, \text{ if } t_s = T, \forall i \in \{0, 1, \cdots, n-1\}. \quad (10)$$

$$BE_{save}(i) = BE_{ave}, \text{ if } t_s = T, \forall i \in \{0, 1, \cdots, n-1\}. \quad (11)$$

The other extreme value of $t_s$ is $t_{unit}$. In this case, within $t_{unit}$, $A(t)$ is constant. So at this time, $BE_{smax}(i)$ and $BE_{save}(i)$ have the same value as $A(t)$. These are the best estimations and can be expressed as

$$BE_{smax}(i) = BE_{save}(i) = A(t), \text{ if } t_s = t_{unit}, \forall i \in \{0, 1, \cdots, n-1\}. \tag{12}$$

By using different combinations of $BE_{smax}$ and $BE_{save}$ and different interval size $t_s$, we can get different bandwidth estimations, $BE(i)$. Thus $BE(i)$ can be expressed as a function as follows.

$$BE(i) = f(BE_{smax}(i), BE_{save}(i), t_s), \forall i \in \{0, 1, \cdots, n-1\}. \tag{13}$$

Different expressions for $BE(i)$ will result in generating different $EE$. A good admission control algorithm can be obtained from an expression of $BE(i)$ that has a low $EE$.

In the next two subsections, we introduce new admission control algorithms based on the proposed slicing scheme.

## 3.2 Future-Max Algorithm

In this subsection, we introduce a new deterministic admission control algorithm which is based on the future maximum bandwidth requirement. Future maximum bandwidth refers to the maximum bandwidth required from the current time point to the end of the playback of the media stream. We term this algorithm as the *Future-Max* (FM) algorithm. The concept behind the FM algorithm can be explained as follows. In the deterministic admission control scheme, the reserved bandwidth for a stream corresponds to its maximum bandwidth. After the playback of the portion that requires the maximum bandwidth, it is not necessary to reserve resources corresponding to the maximum bandwidth. It is definitely beneficial to use the maximum bandwidth of the portions that is not played back instead of the whole stream. The FM algorithm scans through the future intervals in order to determine the maximum bandwidth that is required in future and uses it for admission control. The advantage of the FM algorithm can be observed from Figure 4. After the playback of the media objects that corresponds to the maximum bandwidth, there is no need to reserve bandwidth corresponding to the the maximum or the worst case. Thus beyond the maximum point, the bandwidth reservation can be reduced and performance can be gained as illustrated in Figure 4. The time of the occurrence of the maximum bandwidth affects the performance gain obtained through the use of the FM algorithm.

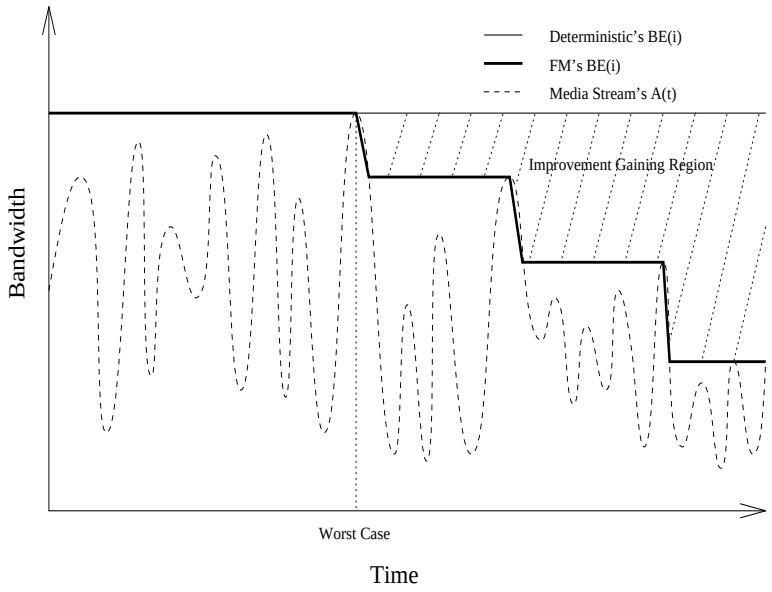


Figure 4: $A(t)$ and $BE(i)$ of Deterministic and FM Algorithms.

The slicing technique described in the previous subsection can be used to implement the FM algorithm. The bandwidth estimation of an incoming request using the FM scheme is denoted as $BE_{FM}(i)$, and is expressed as

$$BE_{FM}(i) = max\{BE_{smax}(j)\}, \forall j \in \{i, i+1, \cdots, n-1\}. \tag{14}$$

Let $l_{sac}$ be the interval at which a new request arrives and the admission control scheme examines whether it can be admitted or not. Let $K$ be the number of streams currently being served. The bandwidth requirements of the $K$ streams is estimated as $BE^k(i)$, where $k \in \{1, 2, \cdots, K\}$. The starting intervals of these streams could be different and are represented as $l^k_{start}$. Let the estimated bandwidth of the new request be $BE^{new}(i)$. A boolean function $\gamma$ can be defined as

$$\begin{gathered}\gamma(l) = (\textstyle\sum_{k=1}^{K} BE^k(l - l^k_{start}) + BE^{new}(l - (l_{sac}+1)) > Total\ Available\ Bandwidth), \\ l \in \{l_{sac}+1, l_{sac}+2, \cdots, l_{end}\}.\end{gathered} \tag{15}$$

where $l_{end}$ is the time at which the playback of the new stream is expected to end. The acceptance or rejection decision of the admission control algorithm is based on the following expression,

$$\text{decision} = \begin{cases} \text{Reject} & \exists\ l \text{ s.t. } \gamma(l) = True; \\ \text{Accept} & \forall\ l \text{ s.t. } \gamma(l) = False. \end{cases} \tag{16}$$

If the decision is "accept", the server will start service at the interval $l_{sac}+1$.

The differences between the deterministic and the FM algorithm can be elaborated as follows. For the deterministic algorithm, the estimated required bandwidth is equal to $\sum_{k=1}^{K} BE_{max}^{k} + BE_{max}^{new}$, which is a constant. For the FM algorithm, the estimated required bandwidth is a non-increasing function. Note that, the $BE_{FM}^{new}(i)$ at the time of admission control is equal to the $BE_{max}^{new}$, as in the case of the deterministic scheme. This value is used for the making the acceptance/rejection decision. Once a request is accepted, it only reserves $BE_{FM}^{new}(i)$ which is less than or equal to $BE_{max}^{new}$ used for reservation in the deterministic case. If the maximum bandwidth(the worst case) is at the interval $\delta$, then

$$\begin{aligned} BE_{FM}(i) &= BE_{max}, \quad i \leq \delta; \\ BE_{FM}(i) &< BE_{max}, \quad i > \delta. \end{aligned} \tag{17}$$

The performance improvement in terms of the $EE$ is given by

$$\Delta_{EE} = EE_{max} - EE_{FM} = \sum_{i=\delta+1}^{n-1} (BE_{max} - BE_{FM}(i)). \tag{18}$$

## 3.3 Interval Estimation Algorithms

In this subsection, we propose a family of admission control policy based on the bandwidth estimations for each of the sliced intervals. The family of *Interval Estimation* (IE) algorithm uses these estimations to decide whether or not to accept a new request. The estimations within the intervals could be deterministic, statistical, or a combination of the two. A general expression for the bandwidth estimation of the $i$th sample using the IE algorithms, $BE_{IE}(i)$ is given as

$$BE_{IE}(i) = \alpha * BE_{smax}(i) + \beta * BE_{save}(i), \text{ where } 0 \leq \alpha, \beta \leq 1 \text{ and } \alpha + \beta = 1. \tag{19}$$

The value of $\alpha$ and $\beta$ can be varied to obtained a family of admission control schemes. The extreme values of $\alpha$, $\beta$, $t_s$ and their corresponding $BE_{IE}$ are listed in Table 2. For the deterministic admission control scheme, $\alpha = 1$, $\beta = 0$, and the sampling time period equals to the whole length of the stream ($t_s = T$). The statistical admission control scheme based on only the average bandwidth requirement refers to the case when $\alpha = 0$, $\beta = 1$, and $t_s = T$. The most accurate IE algorithm can be obtained by setting $t_s = t_{unit}$.

The shaded portion in Figure 5 shows the region where the $BE_{IE}$ will lie with different values of $\alpha$ and $\beta$. This shaded portion is bounded by the curves $BE_{smax}(i)$ and $BE_{save}(i)$,

| $\alpha$ | $\beta$ | $t_s$ | $BE_{IE}$ |
|---|---|---|---|
| 1.0 | 0.0 | T | $BE_{max}$ |
| 0.0 | 1.0 | T | $BE_{ave}$ |
| $\alpha$ | $\beta$ | $t_{unit}$ | $A(t)$ |

Table 2: Typical Values of $\alpha, \beta, t_s$, and Their Corresponding $BE_{IE}$.

which corresponds to the cases of $\alpha = 1, \beta = 0$ and $\alpha = 0, \beta = 1$, respectively. So the relationship among $BE_{IE}(i)$, $BE_{smax}(i)$ and $BE_{save}(i)$ is given as

$$BE_{smax}(i) \geq BE_{IE}(i) \geq BE_{save}(i). \tag{20}$$

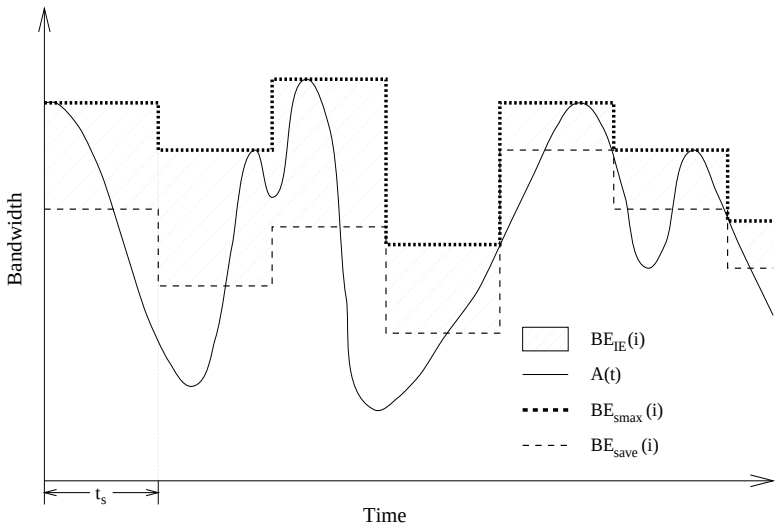


Figure 5: $BE_{IE}(i)$ and Its Relation to $BE_{smax}(i)$ and $BE_{save}(i)$.

In Figure 5, when $t_s$ gets smaller, the corresponding $BE_{smax}(i)$ and $BE_{save}(i)$ become closer to $A(t)$. Since they are the upper and lower bounds of $BE_{IE}(i)$, $BE_{IE}(i)$ will be also closer to $A(t)$. In the extreme case, when $t_s = t_{unit}$, $BE_{smax}(i)$ and $BE_{save}(i)$ are equal to $A(t)$, which also equals to $BE_{IE}(i)$. For this case, the estimation error, $EE$ of $BE_{IE}(i)$ is 0. This scenario reflects the best estimate of the bandwidth requirement. So the best bandwidth requirement estimation $BE_{IE}(i)$, where $t_s = t_{unit}$, is the optimal $BE(i)$, which results in the highest server utilization and provides the highest QoS. An intuitive explanation of this statement is that when $BE(i)$ is in its best case, we get the exact bandwidth requirement at any given time point and use it to check and reserve bandwidth of the server.

## 3.4 Implementation Complexity

In order to improve the server utilization in terms of the number of accepted requests for media streams, we need to model the streams as accurately as possible such that the $EE$ is small. However, an accurate model may need large storage space and may incur high computational complexity. In this subsection, we analyze these implementation complexities and outline the trade-off between accuracy and complexity.

To analyze the storage space requirement, let us consider a video media stream as an example. A typical 90 mins movie occupies 1GB to 3GB storage space using the MPEG-2 encoding scheme. A 100% accuracy level may require to store the bandwidth requirements of each and every video frame. A video stream has 30 frames per second. So the total size of data for a typical video stream will be:

$$30 \text{ frames/sec} * 60 \text{ sec/min} * 90 \text{ min/movie} = 162 \text{ Kframes/movie}.$$

If we use floating point value to store these sliced bandwidths, the maximum amount of extra storage required for a typical video stream will be:

$$4 \text{ Bytes/frame} * 162 \text{ Kframes/movie} = 648 \text{ KBytes/movie}.$$

Compared to the storage requirement of a single movie, even the worst case storage requirement (100% accuracy level) is not too high. So the storage space requirement is not a problem with the proposed slicing-based admission control schemes.

To reduce the computational complexity, the server can use a queue to store the sum of the bandwidth requirements of all the accepted streams for all the slicing intervals. The bandwidth requirements are stored in such a way that the head of the queue has the total bandwidth requirement for the next interval. The next element of the queue stores the bandwidth requirement of the one following the next interval, and so on. At the time of admission control, the elements from the head of the queue are removed until the bandwidth requirement at the next interval is found. For the FM algorithm, the $BE_{FM}(0)$ of the new stream that is requested is added to the head of the queue and is examined for acceptance or rejection as discussed earlier in Section 3.2. This takes O(1) time. If the new request is admitted, the $BE_{FM}(i)$'s are added to the corresponding elements of the queue for bandwidth reservation. This operation takes O(L) time, where L is the number of sliced intervals of the requested media stream. In the case of IE algorithms, admission control algorithm needs to start examining all the queue elements from the header of the queue until the end of the media stream that is being requested. The equations derived in

Section 3.3 are used to make the admission control decision. These operations require O(L) computation. If the request is accepted, the bandwidth requirement of the new stream for each of the sliced intervals are added to the corresponding elements of the queue. This operation needs an additional O(L) computations.

The storage requirement and the computational complexity are directly proportional to the accuracy level and can be reduced by lowering the accuracy level. The accuracy level can be lowered by increasing the interval size. Lowering the accuracy level may in turn lower the QoS and/or the server utilization. However, the decrease of the performance measurements may not be linear. These trade-off are analyzed quantitatively in the next section. In most cases, it may not be desirable to use a very high accuracy level. Significant improvement in the number of accepted requests and the QoS can be achieved at reasonable accuracy level for which the implementation complexity in terms of space and computation will be affordable.

# 4 Experimental Evaluation

In this section, we evaluate the performance of the proposed class of admission control schemes through simulation. The performance measures are defined followed by the description of the simulation environment. The results accompanied by discussions are reported in detail.

## 4.1 Performance Measures

The performance indicators of admission control schemes include the number of requests for media streams that can be accepted and the QoS that can be guaranteed. The number of requests for media streams that can be accepted is also dependent on the required or acceptable QoS. The QoS refers to the proportion of the media streams that are played back within their deadlines. We have defined two different types of QoS. The first type refers to the average QoS for the whole stream. We denote it as $QoS_{ave}$. The second type is the worst case for the QoS at any time point. We denote it as $QoS_{worst}$. These QoS terms can be expressed as

$$QoS_{ave} = \frac{\int_0^T QoS(t)dt}{T}, \tag{21}$$

$$QoS_{worst} = \min\{QoS(t)\}, t \in [0, T]. \tag{22}$$

$QoS_{worst}$ corresponds to the minimum QoS that can be tolerated by a client. If a client is tolerable to a degraded QoS (may be because of lower cost), we consider the required QoS as $QoS_{worst}$ in order to ensure that the QoS never falls below the acceptable level. While using the deterministic schemes, such as FM or IE (with $\alpha = 1.0$), the actual bandwidth requirement for the acceptable QoS is equal to the bandwidth requirement at $QoS = 1.0$ times the $QoS_{worst}$. Thus, it is guaranteed that the QoS will not be worse than $QoS_{worst}$. In such cases, $QoS_{ave}$ will be much higher then $QoS_{worst}$. Instead of considering the acceptable QoS as $QoS_{worst}$, if we regard that as $QoS_{ave}$, then the server utilization could be improved. However, the jitters may not be uniformly spread out and may concentrate at few time periods and the $QoS_{worst}$ be well below the acceptable range. Similar issues have been addressed recently using (m,k)-firm deadlines [15].

The server utilization is measured in terms of the total number of requests for media streams that can be supported simultaneously. The accuracy level is measured in terms of the size of the intervals, $t_s$. If $t_s$ is equal to $t_{unit}$, the accuracy level is defined as 100%. The case of $t_s = T$ corresponds to 0% accuracy.

## 4.2 Simulation Model

We have implemented a time driven simulator for the evaluation of the proposed algorithms. In the simulator, we use the real-trace data of MPEG-1 frame size from the University of Wuerzburg [16]. The frame size traces were extracted from MPEG-1 sequences which have been encoded using the Berkeley MPEG-encoder (version 1.3) which was adapted for motion-JPEG input. The frame sizes are in bits. The videos were captured in motion-JPEG format from a VCR (VHS) with a SUN SPARCstation 20 and SunVideo. The capture rate of the SunVideo video system was between 19 to 25 fps. The encoding pattern is "IBBPBBPBBPBB" and the GOP size is 12.

Since there are 20 different traces, we assume that those VBR media streams are stored in the server and the clients can request any one of them selected randomly. The simulator has the following components - a stream bandwidth requirement generator, the slicing unit, and the admission control unit.

The stream bandwidth generator is responsible for generating the bandwidth requirement of the groups from the MPEG-1 real traces. Since a reasonable transferring unit is a group in MPEG-1 format instead of frame, the generator adds up all the frame sizes in a logical group and designates that to be the stream bandwidth requirement for that

group. Then the generated stream bandwidth requirements can be computed for the slicing unit. The function of the slicing unit is to store the bandwidth requirements based on the sliced intervals $t_s$. The admission control unit decides the acceptance/rejection of the new requests using the equations derived in Section 3.

The clients are assumed to request a new media stream at each time unit. Thus, there will always be a media stream request waiting for service at the admission control unit. This is done to ensure the effectiveness of the admission control scheme without the affect of the new requests arrival pattern. We implement the FM and IE algorithms according to the equations derived in Section 3. The performance parameters were obtained by measuring the parameters several times and averaging the results.

## 4.3 Results and Observations

We present a comparative evaluation of the performance of the FM and IE algorithms. The results are obtained with respect to the the accuracy level, and the variation of $\alpha$ and $\beta$ for the class of IE algorithms.

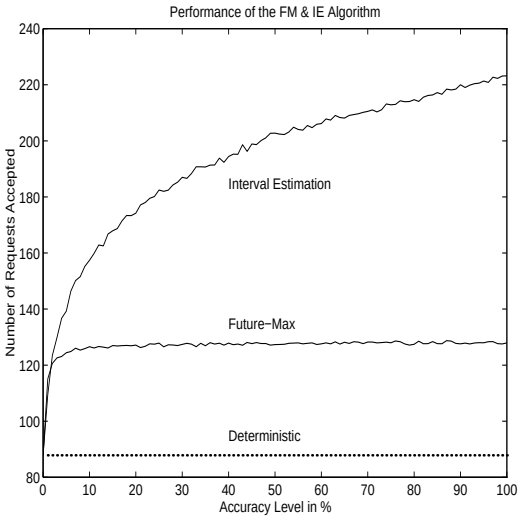


Figure 6: Performance of the FM & IE Algorithms

Figure 6 shows the performance improvement obtained using the FM algorithm and IE algorithm compared to the deterministic algorithm. It can be observed that the curve obtained using the FM algorithm increase with the increase in the accuracy level. However, the rate of increase diminishes with the increase in the accuracy level. Initially, with a small increase in the accuracy level, the number of requests accepted increases noticeably.

It may not be necessary or worth implementing higher accuracy level at the cost of the added complexity because of the small improvement in the acceptance rate. The overall performance improvement is 46%.

In Figure 6, we also show the curve corresponding to the IE algorithm with $\alpha = 1$ and $\beta = 0$. Thus the bandwidth requirement is estimated as the maximum bandwidth requirement within the interval. The 0% accuracy level corresponds to the deterministic case. The optimal case is reflected by the 100% accuracy level. With the increase in accuracy level, the rate of increase in the number of requests accepted is monotonous. The performance improvement is about 154%. It can be inferred that a significant performance improvement is obtained by using the IE admission control scheme.

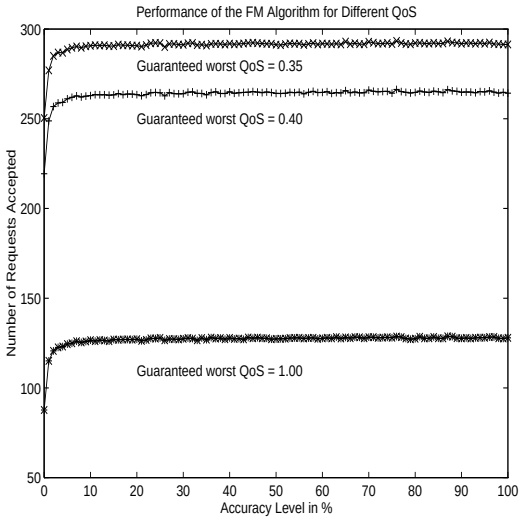


Figure 7: Performance of the FM Algorithm for Different QoS.

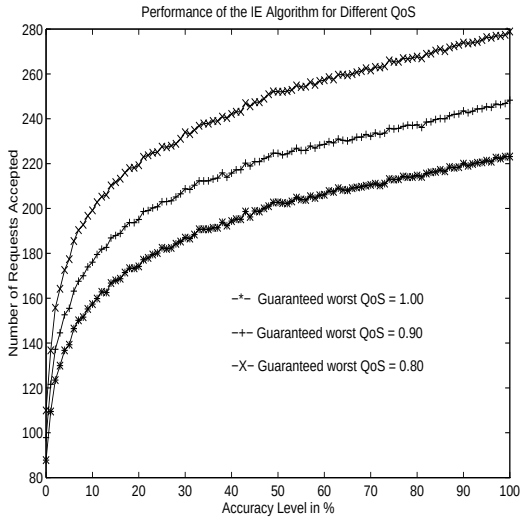


Figure 8: Performance of the IE Algorithm for Different QoS.

Figures 7 and 8 show the number of streams that can be accepted for different values of guaranteed $QoS_{worst}$ for the FM and IE ($\alpha = 1, \beta = 0$) algorithms, respectively. As expected, the number of accepted streams decreases with the increase in the guaranteed $QoS_{worst}$. For example, at 100% accuracy level, the performance improvement of the IE algorithm is about 11% for $QoS_{worst} = 0.9$, and 25% for $QoS_{worst} = 0.8$ higher than the performance of $QoS_{worst} = 1.0$. It is observed that the accepted number of streams for the FM algorithm does not vary significantly with the increase in the accuracy level beyond a certain point. However, in the case of the IE algorithm, there is almost a linear increase in the number of accepted streams with the increase in the accuracy level.

In order to guarantee $QoS_{worst}$, we scale the total available bandwidth by a factor of

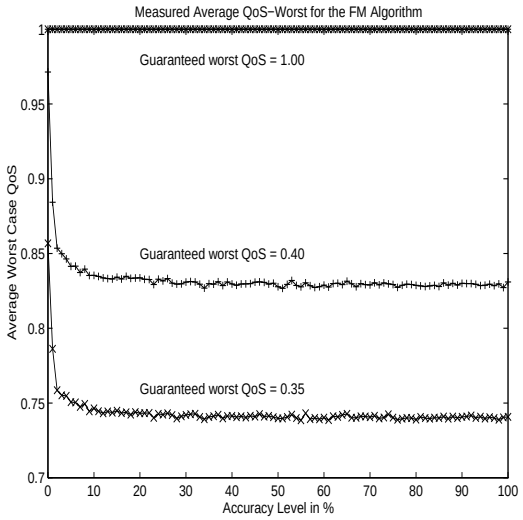


Figure 9: Measured Average $QoS_{worst}$ for the FM Algorithm.

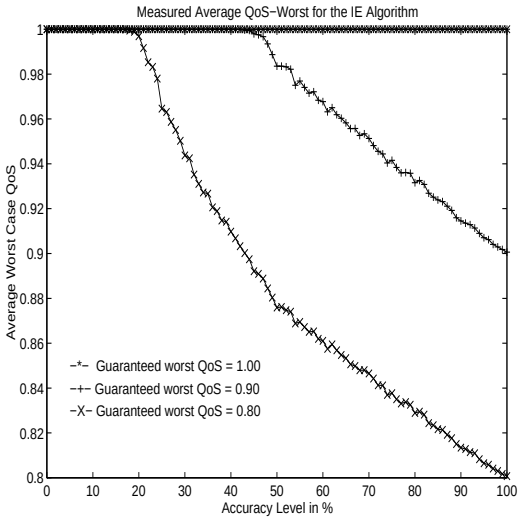


Figure 10: Measured Average $QoS_{worst}$ for the IE Algorithm.

$\frac{1}{QoS_{worst}}$. However, because of the $EE$ (as discussed in Section 2), the measured average $QoS_{worst}$ may be quite higher than the guaranteed $QoS_{worst}$. This is demonstrated in Figures 9 and 10 where the measured average $QoS_{worst}$ is obtained through simulation experiments. It is observed that the measured average $QoS_{worst}$ is significantly higher than the guaranteed $QoS_{worst}$ for the FM algorithm. This is because of the high $EE$ incurred by the FM algorithm. The difference between the measured $QoS_{worst}$ and the guaranteed $QoS_{worst}$ reduces with the increase in the accuracy level for the IE algorithm. With the increase in accuracy level, the $EE$ decreases, thus lowering the difference between the guaranteed $QoS_{worst}$ and the measured average $QoS_{worst}$ for the IE algorithm. Note that at 100% accuracy level, the measured average $QoS_{worst}$ is identical to the guaranteed $QoS_{worst}$ for the IE algorithm as the $EE$ at this level is equal to zero.

The average QoS values were measured corresponding to the system configurations of Figure 9 and 10 and are illustrated in Figures 11 and 12 for the FM and IE algorithms, respectively. As observed in the previous experiments, the $QoS_{ave}$ for the FM algorithm (Figure 11) is almost constant after an accuracy level of about 20%. However, the $QoS_{ave}$ is significantly higher than the $QoS_{worst}$ (both guaranteed and measured average) for lower values of $QoS_{worst}$. Similar observations can be observed for the $QoS_{ave}$ of the IE algorithm shown in Figure 12. The only difference is that the $QoS_{ave}$ degrades rapidly after a certain accuracy level when the guaranteed $QoS_{worst}$ is less than 1.0 while employing the IE algorithm. The rapid degradation is due to the sharp increase in the number of accepted

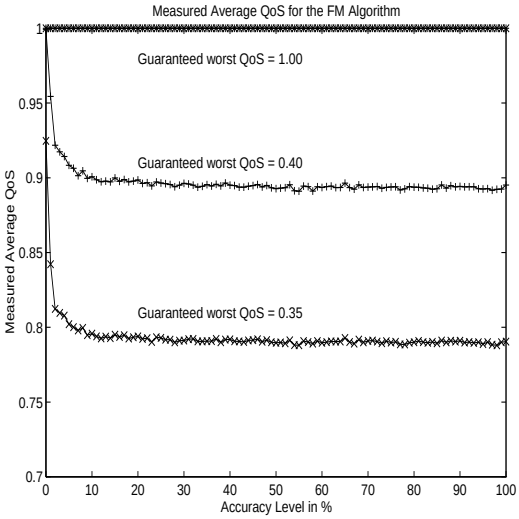


Figure 11: Measured Average QoS for the FM Algorithm.

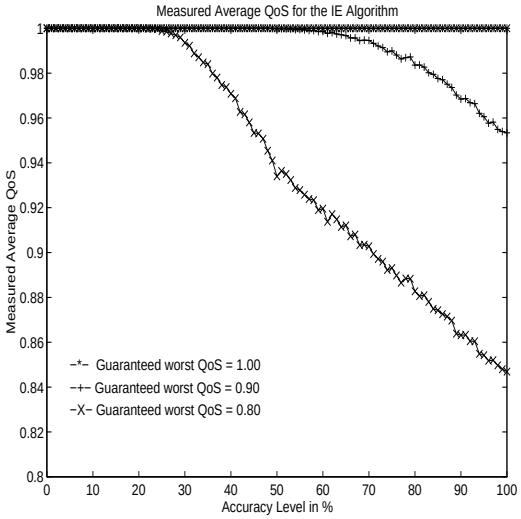


Figure 12: Measured Average QoS for the IE Algorithm.

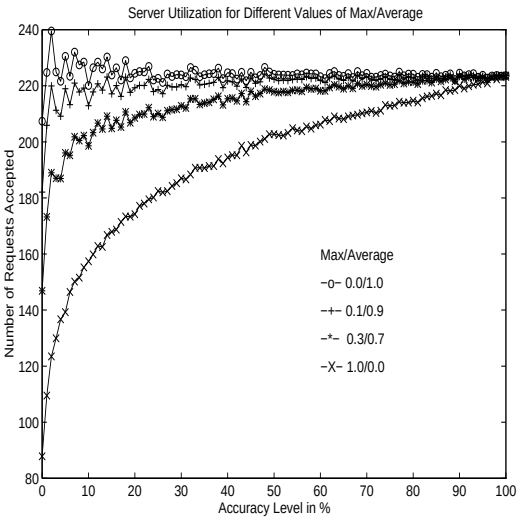


Figure 13: Server Utilization for Different Values of $\alpha$ and $\beta$.

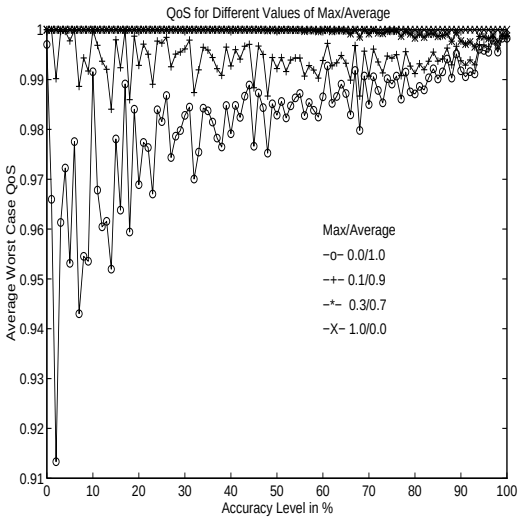


Figure 14: QoS for Different Values of $\alpha$ and $\beta$.

requests.

In Figure 13, we show the variation of the number of accepted streams with respect to different values of $\alpha$ and $\beta$. The corresponding variations in the QoS is depicted in Figure 14. The max and average values in Figures 13 and 14 denote $\alpha$ and $\beta$, respectively. For higher values of $\beta$, the number of accepted streams is high with minimal variation. Correspondingly the QoS values are lower as shown in Figure 14. All the curves with different values of $\alpha$ and $\beta$ converge at the 100% accuracy level as shown in Figures 13 and 14. This is because of the fact that at 100% accuracy level, $BE_{max}$ and $BE_{ave}$ have the same value which is equal to $A(t)$. So the corresponding $BE_{IE}(i)$ is the same at this level. With the increase in accuracy level, the EE is reduced, leading to an increase in the QoS. Thus the curves in Figure 14 mostly have a positive slope. However, it is quite interesting to observe that for certain values of $\alpha$ and $\beta$, the QoS factor degrades at higher accuracy level. At higher accuracy levels, the number of accepted requests is high. Thus there is an increase in the total bandwidth requirement, which explains the decreasing trend of QoS.

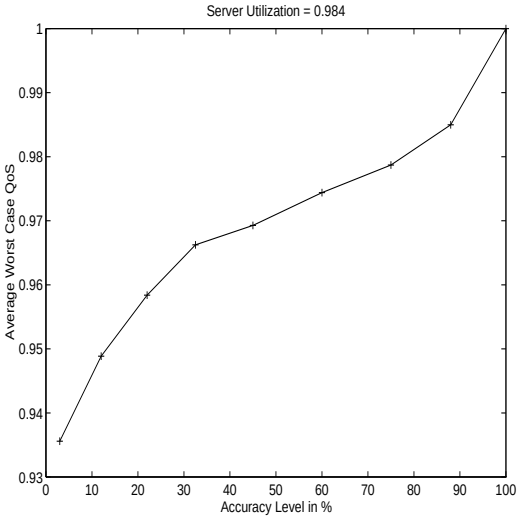


Figure 15: Measured Worst QoS for a fixed Server Utilization.

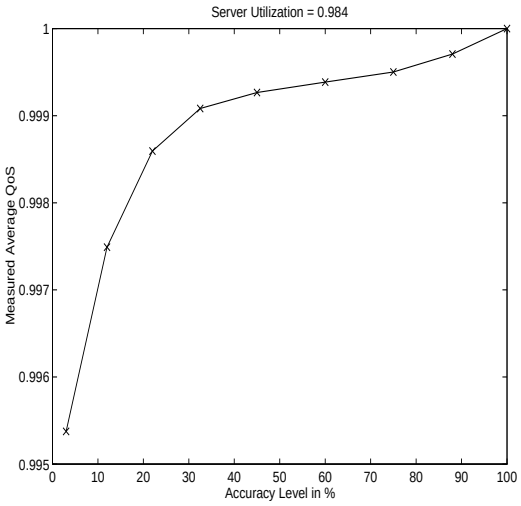


Figure 16: Measured Average QoS for a fixed Server Utilization.

Next, we compare the effect of the accuracy level on the QoS. We measure the $QoS_{worst}$ and $QoS_{ave}$ values with the same server utilization with respect to the accuracy level. The results are illustrated in Figures 15 and 16, respectively. The server utilization is held constant by keeping the number of accepted requests fixed. The results are shown for the IE algorithm with $\alpha = 1, \beta = 0$. With the increase in accuracy level, the $EE^-$ is reduced and thus there is an increase in both types of QoS. It is observed that both $QoS_{worst}$

and $QoS_{ave}$ increase sharply at the low accuracy level. This trend further advocates the inference that significant gain in performance can be achieved with a coarse-grain slicing of the bandwidth requirements of the media streams.

# 5 Concluding Remarks

In this paper, we have proposed a new family of admission control algorithms. These algorithms are based on a slicing technique and use an aggressive method to compare and reserve the bandwidth available at the server. Two types of admission control schemes are proposed. The first scheme, called FM algorithm, is based on the maximum bandwidth requirement in future for the media streams. The second algorithm, called IE algorithm, defines a class of algorithms. The IE algorithm uses a combination of the maximum and average bandwidth requirement within each interval to estimate the bandwidth. Different IE algorithms can be developed by varying the proportion of maximum and average bandwidth requirements within each of the sliced interval. The length of the slicing interval can be varied to obtain different levels of accuracy. We have discussed the trade-off between the accuracy level, the implementation complexity, and the performance of the admission control algorithm.

The performance of the proposed admission control schemes are evaluated through simulation experiments. It is observed that the performance improvement with the FM algorithm is almost negligible beyond the 20% accuracy level. However, the performance improvement in terms of the number of streams supported is almost linear in case of the IE algorithms. For a fixed server utilization, the QoS of the servers improves with the increase in accuracy level in the case of both FM and IE algorithms. Thus, an efficient algorithm for the family of algorithm proposed here can be adopted by a server on the basis of the required QoS and performance with respect to the implementation complexity.

# References


[1] G. Miller, G. Baber, and M. Gillilan, "News On-Demand for Multimedia Networks," in *Proceedings of the ACM Multimedia'93*, pp. 383–392, August 1993.

[2] A. L. N. Reddy and J. C. Wyllie, "I/O Issues in a Multimedia System," *IEEE Computer*, pp. 69–74, March 1994.

[3] D. J. Gemmell, H. M. Vin, D. D. Kandlur, and P. V. Rangan, "Multimedia Storage Servers: A Tutorial," *IEEE Computer*, pp. 40–49, May 1995.

[4] H. M. Vin, "Multimedia System Architecture," in *Proceedings of the SPIE International Symposium on Photonics for Industrial Applications, Critical Review on Defining Global Information Infrastructure: Infrastructure, Systems and Services*, November 1994.

[5] D. Anderson, Y. Osawa, and R. Govindan, "A File System for Continuous Media," *ACM Transactions on Computer Systems*, vol. 10, pp. 311–337, November 1992.

[6] J. Gemmell and S. Christodoulakis, "Principles of Delay Sensitive Multimedia Data Storage and Retrieval," *ACM Transactions on Information System*, vol. 10, no. 1, pp. 51–90, 1992.

[7] P. V. Rangan and H. M. Vin, "Designing File Systems for Digital Video and Audio," in *Proceedings of the 13th Symposium on Operating Systems Principles(SOSP'91), Operating Systems Review*, pp. 81–94, October 1991.

[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*, pp. 393–400, August 1993.

[9] H. M. Vin and P. V. Rangan, "Designing a Multi-User HDTV Storage Server," *IEEE Journal on Selected Areas in Conmmunications*, vol. 11, pp. 153–164, January 1993.

[10] P. Yu, M. S. Chen, and D. D. Kandlur, "Design and Analysis of a Grouped Sweeping Scheme for Multimedia Storage Management," in *Proceedings of 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 38–49, November 1992.

[11] H. M. Vin, P. Goyal, A. Goyal, and A. Goyal, "A Statistical Admission Control Algorithm for Multimedia Servers," in *Proceedings of the ACM Multimedia'94*, October 1994.

[12] H. M. Vin, A. Goyal, A. Goyal, and P. Goyal, "An Observation-Based Admission Control Algorithm for Multimedia Servers," in *Proceedings of the First IEEE International Conference on Multimedia Computing and Systems*, pp. 234–243, May 1994.

[13] H. M. Vin and P. V. Rangan, *Multimedia Systems and Techniques*, ch. 4, pp. 123–144. Kluwer Academic Publishers, 1996.

[14] H. M. Vin, A. Goyal, and P. Goyal, "Algorithms for Designing Large-Scale Multimedia Servers," *Computer Communications*, vol. 18, pp. 192–203, March 1995.

[15] M. Hamdaoui and P. Ramanattan, "A Dynamic Priority Assignment Technique for Streams with (m,k)-Firm Deadlines," *IEEE Trans. on Computers*, vol. 44, pp. 1443–1451, December 1995.

[16] O. Rose, "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems," Tech. Rep. 101, University of Wuerzburg, Institute of Computer Science, February 1995.