Journal of Computer and System Sciences **64**, 219–248 (2002)
doi:10.1006/jcss.2001.1799, available online at http://www.idealibrary.com on IDEAL®

Competitive On-line Scheduling of Continuous-Media Streams[1]

Minos Garofalakis[2]

Bell Laboratories, 600 Mountain Avenue, Murray Hill, New Jersey 07974
E-mail: minos@research.bell-labs.com

Yannis Ioannidis[3]

Department of Informatics and Telecommunications, University of Athens, 157 84 Athens, Greece
E-mail: yannis@di.uoa.gr

Banu Özden and Avi Silberschatz

Bell Laboratories, 600 Mountain Avenue, Murray Hill, New Jersey 07974
E-mail: ozden@research.bell-labs.com, avi@research.bell-labs.com

Received January 20, 2000; revised August 27, 2001

Multimedia applications require a guaranteed level of service for accessing continuous-media data. To obtain such guarantees, the database server where the data are residing must employ an admission control scheme to limit the number of clients that can be served concurrently. We investigate the problem of *on-line admission control*, where the decision of whether to accept or reject a request must be made without any knowledge about future requests. Employing *competitive analysis* techniques, we address the problem in its most general form with the following key contributions: (1) We prove a tight upper bound on the competitive ratio of the conventional Work-Conserving ($\mathscr{WC}$) policy, showing that it is within a factor $\frac{1+\Delta}{1-\rho}$ of the optimal clairvoyant strategy, where Δ is the ratio of the maximum to minimum request length (i.e., time duration), and ρ is the maximum fraction of the server's bandwidth that a request can demand; (2) we prove a lower bound of $\Omega(\frac{\log \Delta}{1-\rho})$ on the competitive ratio of *any deterministic or randomized* admission control scheme, demonstrating an exponential gap between greedy and optimal on-line solutions; (3) we propose simple deterministic schemes based on the idea of *bandwidth prepartitioning* that guarantee competitive ratios

[1] An extended abstract of this paper has appeared in the *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'98)*, Seattle, Washington, June 1998.

[2] Part of this work was done while the first author was with the University of Wisconsin-Madison.

[3] Partially supported by the Special Research Grants Account (*Ειδικος Λογαριασμος Κονδυλιων Ερευνας*) of the University of Athens under Grant 70/4/3244.

0022-0000/02 $35.00

within a small constant factor of $\log \Delta$ (i.e., they are provably *near-optimal*) if $\rho < 1/\lceil \log \Delta \rceil$; and (4) we introduce a novel admission control policy that partitions the server bandwidth based on the expected popularities of different request lengths and experimentally demonstrate its benefits compared to $\mathcal{WC}$.

Key Words: multimedia databases; resource scheduling; admission control; on-line algorithms; competitive analysis.

1. INTRODUCTION

Next generation database systems will need to provide support for various forms of multimedia data such as images, video, and audio. These new data types differ from conventional alphanumeric data in their characteristics, and hence require different techniques for their organization and management. A fundamental issue is that digital video and audio *streams* consist of a sequence of media quanta (video frames or audio samples) which convey meaning only when presented continuously in time. Hence, a multimedia database server needs to provide a guaranteed level of service for accessing such *continuous media* (CM) streams in order to satisfy their pre-specified real-time delivery rates. Given the limited amount of resources (e.g., memory, disk bandwidth, disk storage), it is a challenging problem to design effective resource management algorithms that can provide on-demand support for a large number of concurrent continuous media clients.

Typically, clients issue requests for the playback of specific CM *clips* (i.e., contiguous portions of audio or video) residing at the database server. A crucial component of CM service is the *admission control* mechanism, which is invoked whenever a new request arrives to decide whether to accept or reject the request. By accepting a request, the server commits to satisfy the resource requirements (e.g., disk bandwidth, memory) of the corresponding playback stream throughout its execution, whereas rejected requests must pursue a different course of action (depending on the application).[4] The effectiveness of the admission control component is of vital importance for the following reasons. First, the resource requirements of CM applications are high. Second, they require fractions of the server's resources to be reserved to meet their stringent performance requirements. Third, these applications tend to last for relatively long periods of time. Reserving large portions of the resources for long durations can result in drastic degradation of server utilization if the server makes wrong decisions whom to admit.

An important characteristic of admission control is the introduction of an *on-line decision making* element—the decision of whether to accept or reject a request has to be made without any knowledge of future requests, with the understanding that once a request is accepted, it is guaranteed a level of service throughout its duration (i.e., the schedule is *non-preemptive*). In this paper, we study the implications of the on-line nature of the admission control problem for CM streams. Our performance metric for admission control strategies is the *total server throughput* over a sequence

[4] Our model corresponds to the *Full-VOD* service model [1, 8, 21]. Other service models have also been explored in the literature [16].

of requests and our methodology is based on the competitive analysis framework for on-line algorithms [29]. The basic quality metric in this framework is the *competitive ratio* of an on-line algorithm, which is defined to be the maximum (over all possible request sequences) value of the ratio of the performance of the optimal off-line algorithm for a request sequence to the performance of the on-line algorithm for the same request sequence. Note that, by definition, competitive analysis is tantamount to a worst-case analysis in the off-line case. An algorithm with a low competitive ratio is one that performs close to optimal in all situations. Since no assumptions are made about the sequence of requests offered to the server, the competitive ratio provides a very *robust* measure of performance.

We assume a centralized database server where incoming playback requests require some fraction of the server's bandwidth for some period of time. For example, a request to view a half-hour MPEG-1 video clip requires 1.5 Megabits per second (Mbps) of the server's bandwidth for the 30 minutes of playback. We consider two different cases of the problem. In the first case, we assume that all requests require the same fraction of the server's bandwidth (e.g., all clips are MPEG-1 encoded videos). thus, the server can be viewed as a of collection of available playback channels. In the second, more general case, different fractions of the server's bandwidth can be reserved. We show that the conventional Work-Conserving ($\mathscr{WC}$) policy, where an incoming request is always admitted if there is sufficient bandwidth to accommodate it, can behave poorly in an on-line setting. More specifically, we show that the competitive ratio of $\mathscr{WC}$ is $1+\Delta$ for the case of identical bandwidth requests and $\frac{1+\Delta}{1-\rho}$ for the case of variable bandwidth requests, where Δ is the ratio of maximum to minimum request length and ρ is the maximum fraction of the server's bandwidth that a request can demand. We introduce novel admission control strategies based on the idea of *prepartitioning* the bandwidth capacity of the server among requests of different length and prove that, for sufficiently large server bandwidth, these strategies are $O(\log \Delta)$-competitive. We also show an $\Omega(\log \Delta)$ (resp. $\Omega(\frac{\log \Delta}{1-\rho})$) lower bound on the competitive ratio of any deterministic *or* randomized algorithm for the identical (resp. variable) bandwidth case, thereby establishing the near-optimality of our on-line algorithms. Based on the above results, we propose a bandwidth prepartitioning scheme that makes use of clip popularities to ensure good average-case as well as worst-case performance. The results of our preliminary experimental study verify the benefits of our scheme as compared to $\mathscr{WC}$. More specifically, both algorithms are shown to perform adequately well when the server is underutilized or persistently overloaded. However, we expect that a well designed system has undergone effective capacity planning and, therefore, will not be overloaded persistently but only at short time intervals. We capture such short term overloads in our experiments and demonstrate that our admission control scheme outperforms $\mathscr{WC}$ substantially under these workloads.

The remainder of this paper is organized as follows. Section 2 reviews related work in the area of multimedia databases and on-line algorithms. In Section 3, we provide the necessary definitions and formulate the on-line admission control problem. Section 4 introduces our Bandwidth Prepartitioning strategies and presents our competitive analysis results for both identical bandwidth and variable bandwidth requests. In Section 5 we discuss the findings of a preliminary experimental study with the admission control schemes described in this paper. Finally, Section 6 concludes the paper with a discussion on the implications of our results in the context of data placement in distributed CM databases.

2. RELATED WORK

Resource scheduling issues in CM databases have attracted a lot of interest from researchers in recent years [10, 11, 16, 21, 24, 26, 28, 34]. However, little attention has been paid in the multimedia literature to the on-line nature of the admission control problem for CM database servers. Long and Thakur [22] present simple adversary arguments to show that no on-line algorithm can achieve a *constant* competitive ratio in the context of the Swift distributed I/0 architecture. Aggarwal *et al.* [1, 8] present a competitiveness analysis for a different service model, termed *Shared Video-On-Demand.* Requests are notified of acceptance or rejection within a server-specified time interval (termed *notification interval*) from their arrival. Admitted requests waiting for the same clip can be batched onto a single stream. They show that allowing for sufficiently large notification intervals (linear in the length of the clips) can guarantee constant competitive ratios for simple scheduling algorithms [1, 8]. The Shared Video-On-Demand model is different from our model of CM service in the sense that it tries to capture the effects of wait tolerance and hatching on the number of clients served. Therefore, their results can be viewed as orthogonal to ours. Furthermore, the corresponding analysis assumes that (a) all CM clips have the same length (i.e., time duration) and require the same amount of bandwidth; and (b) any two requests by the same client must be separated by at least the duration of a clip. These assumptions severely limit the applicability of their results to general CM servers.

There is also a significant body of related work in the field of on-line algorithms for bandwidth allocation and circuit routing in communication networks. Lipton and Tomkins [20] study the competitiveness of randomized strategies for the non-preemptive On-line Interval Scheduling (OIS) problem, which essentially corresponds to on-line admission control in a server that can support a *single* playback stream. Under the assumption that the ratio Δ of longest to shortest interval is *not known a priori*, they present an $O((\log \Delta)^{1+\varepsilon})$-competitive randomized algorithm and show that no $O(\log \Delta)$-competitive algorithm can exist. Extensions to their randomized scheme are presented by Faigle *et al.* [13]. Awerbuch *et al.* [4] examine the more general problem of non-preemptive circuit routing on tree-structured networks and propose a general randomized technique termed "Classify and Randomly Select." The main idea is to classify on-line events in disjoint classes and then consider only the events that are assigned to a *randomly selected* class. By averaging over all possible random choices, "Classify and Randomly Select" achieves logarithmic competitive ratios (in an expected sense). However, the idea of an admission control scheme that considers only one randomly selected class of user requests and simply ignores all others is obviously not very appealing for CM database servers, since it ignores fundamental requirements such as fairness. Our proposed schemes also employ the idea of on-line classification, but they also are completely *deterministic* without compromising near-optimal competitiveness (for sufficiently large server bandwidth). Awerbuch *et al.* [3] consider non-preemptive circuit routing on general networks. They present a deterministic scheme (called ROUTE_OR_BLOCK) which, assuming that the bandwidth requested by a single circuit never exceeds an $O(1/\log nT_{max})$ fraction of edge capacity, achieves a competitive ratio of $O(\log nT_{max})$, where n is the number of nodes in the network and T_{max} is the maximum duration of a call. They also prove that their scheme is near-optimal for deterministic on-line routing. The ROUTE_OR_BLOCK algorithm is based on ideas developed for multicommodity network flow problems. Roughly speaking, the main idea is to assign each edge a "length" that is exponential in its current load and route an incoming circuit only if the length of the shortest routing path is less than the "benefit" associated with the circuit. However, as reported by Plotkin [27] and Gawlick [17], ROUTE_OR_BLOCK exhibited consistently poor performance in an actual implementation. Ad hoc changes in the algorithm's parameters were necessary to improve its behavior. Furthermore, the ROUTE_OR_BLOCK scheme itself is rather complex and unintuitive and it is not clear how it can benefit from the knowledge of statistical information, such as request popularities.

Finally, we should note that allowing *preemption* of requests can lead to better competitive ratios for on-line scheduling and admission control problems [7, 9, 14, 15, 19, 33]. However, the assumption of preemptability is unrealistic in the context of CM applications.

3. PROBLEM FORMULATION

We view a CM database server as a "black box" capable of offering a sustained bandwidth capacity of B. The input sequence consists of a collection of *requests* $\bar{\sigma} = \sigma_1, \sigma_2, ..., \sigma_N$. The ith request is represented by the tuple $\sigma_i = (t_i, l_i, r_i)$, where l_i, r_i denote the length and bandwidth requirement (respectively) of the requested CM clip and t_i is the arrival time of σ_i. Given a collection of different requests that are handled by a server (based, for example, on the clips available at the server or the server's usage patterns), we use l_{max} (l_{min}) to denote the length of the longest (shortest) request. (r_{max} and r_{min} are defined similarly.) Finally, we define $\Delta = l_{max}/l_{min}$ and $\rho = r_{max}/B$.

We use *competitive analysis* [29] to measure the performance of different admission control strategies. Our optimization metric is the *total throughput*, that is, the bandwidth-time product over a given sequence of requests. Specifically, given an on-line scheduling policy A and an input sequence $\bar{\sigma}$, we define the *benefit* of A on $\bar{\sigma}$ as $V_A(\bar{\sigma}) = \sum_{S_1} l_i \cdot r_i$, where $S_A \subseteq \bar{\sigma}$ is the set of requests scheduled by A. The *competitive ratio* of an on-line algorithm A is defined as the maximum value $\kappa(A)$ over all possible request sequences of the ratio of the throughput achieved by the optimal off-line algorithm for a request sequence to the throughput achieved by A for the same sequence. If A is a *randomized algorithm*, then the throughput achieved by A for a request sequence is averaged over all possible "coin flips" of A [23]. More formally,

$$\kappa(A)=\begin{cases}\sup\limits_{A^*,\bar{\sigma}} \dfrac{V_{A^*}(\bar{\sigma})}{V_A(\bar{\sigma})}, & \text{if } A \text{ is deterministic}\\[2ex] \sup\limits_{A^*,\bar{\sigma}} \dfrac{V_{A^*}(\bar{\sigma})}{E_A[V_A(\bar{\sigma})]}, & \text{if } A \text{ is randomized,}\end{cases}$$

where $\bar{\sigma}$ ranges over all possible request sequences, A^* ranges over all off-line (i.e., *clairvoyant*) algorithms, and the expectation $E_A[\cdot]$ is taken over the random choices of A. Thus, an algorithm with a *small* competitive ratio is guaranteed to perform close to optimal in all situations. We say that algorithm A is k-competitive if $\kappa(A) \leqslant k$.

It is conventional in the analysis of on-line methods to describe things in terms of a game between a player (the on-line algorithm) and an *adversary* (the off-line algorithm), whose goal is to produce a request sequence that would force the player to perform poorly. For randomized algorithms, different models of adversaries have been proposed depending on the adversary's knowledge of the player's random choices [23]. The lower bounds presented in this paper assume the "weakest" model of an *oblivious* adversary; that is, an adversary that is oblivious to the random choices made by the on-line algorithm.

4. COMPETITIVE ANALYSIS OF ADMISSION CONTROL

4.1. The Greedy/Work- Conserving Policy

The scheduling strategy used as a starting point in our study is the basic *Work-Conserving* ($\mathscr{WC}$) scheme traditionally used for admission control in CM servers. $\mathscr{WC}$ is based on the following greedy rule: *schedule request σ_i immediately if the server has at least r_i bandwidth available at time t_i; otherwise, reject σ_i*. As our results show, $\mathscr{WC}$ offers rather poor performance guarantees in an on-line setting.

First, consider a restricted version of the admission control problem in which all requests require a constant fraction of the server's bandwidth B. That is, $r_i = r$ for all i. Let $c = \lfloor \frac{B}{r} \rfloor \geqslant 1$ denote the number of playback *channels* available at the server.[5] The following theorem establishes the competitiveness of $\mathscr{WC}$ in this setting.

THEOREM 4.1 (Competitiveness of $\mathscr{WC}$, Identical Bandwidth Case). *$\mathscr{WC}$ is $(1+\Delta)$-competitive for scheduling requests with identical bandwidth requirements on-line; that is, $\kappa(\mathscr{WC}) \leqslant 1+\Delta$. Furthermore, this bound is tight.*

Proof. Let $V_{\mathscr{WC}}(\bar{\sigma})$ and $V_{OPT}(\bar{\sigma})$ denote the total throughput achieved over a sequence of requests $\bar{\sigma}$ by the Work-Conserving policy and the optimal off-line scheduler, respectively. Let $A_{\mathscr{WC}}$ denote the set of requests in $\bar{\sigma}$ that are accepted by $\mathscr{WC}$.

Observe that by the operation of the $\mathscr{WC}$ policy, a request $(t_i, l_i) \in A_{\mathscr{WC}}$ can cause a later request (t_j, l_j) to be rejected only if the time interval $[t_i, t_i + l_i)$ contains the starting point of l_j (i.e., t_j). (This condition is not sufficient since there

[5] Note that, in this case, maximizing throughput is equivalent to maximizing total scheduled request length [20].

may be a free channel to accommodate the later request.) This implies that the maximum possible total length that was rejected because of the selection of $(t_i, l_i) \in A_{\mathscr{WC}}$ is $l_i + l_{max}$ (that is, when scheduling (l_i, t_i) causes the rejection $(l_i - \varepsilon, t_i + \frac{\varepsilon}{2})$ and $(l_{max}, t_i + l_i - \frac{\varepsilon}{2})$, $\varepsilon > 0$). This is obviously an upper bound, since scheduling such a total length could conflict with other scheduled intervals. Thus, an upper bound on the maximum total scheduled length for any (off-line) algorithm that does not violate the server's bandwidth constraint is given by the expression

$$\sum_{(t_i, l_i) \in A_{\mathscr{WC}}} (l_i + l_{max}) = \sum_{(t_i, l_i) \in A_{\mathscr{WC}}} l_i + |A_{\mathscr{WC}}| \cdot l_{max}.$$

So, we have

$$\frac{V_{OPT}(\bar{\sigma})}{V_{\mathscr{WC}}(\bar{\sigma})} \leqslant \frac{\sum_{(t_i, l_i) \in A_{\mathscr{WC}}} l_i + |A_{\mathscr{WC}}| \cdot l_{max}}{\sum_{(t_i, l_i) \in A_{\mathscr{WC}}} l_i} \leqslant 1 + \frac{|A_{\mathscr{WC}}| \cdot l_{max}}{|A_{\mathscr{WC}}| \cdot l_{min}} = 1 + \Delta.$$

We now describe a problem instance to show that this $(1 + \Delta)$ competitive factor is tight for $\mathscr{WC}$. Consider a single-channel system (i.e., $c = 1$) and assume the sequence of requests: $(0, l_{min} + \varepsilon)$, $(\frac{\varepsilon}{2}, l_{min})$, and $(l_{min} + \frac{\varepsilon}{2}, l_{max})$. It is then easy to see that the competitive ratio of $\mathscr{WC}$ for this instance will be

$$\frac{l_{min} + l_{max}}{l_{min} + \varepsilon} \xrightarrow{\varepsilon \to 0} 1 + \Delta.$$

This completes the proof. ∎

In general, a playback request requires an arbitrary portion of the server's bandwidth B. This bandwidth requirement depends, for example, on the data encoding method used (e.g., MPEG-1, MPEG-2) or the Quality of Service (QoS) specified by the client [30]. The following theorem shows the effect of this more general model on the competitive factor of the $\mathscr{WC}$ policy.

THEOREM 4.2 (Competitiveness of $\mathscr{WC}$ Variable Bandwidth Case). *$\mathscr{WC}$ is $(B \cdot (1 + \Delta) / \max\{B - r_{max}, r_{min}\})$-competitive for scheduling requests with different bandwidth requirements on-line.*

Proof. Consider a sequence of requests $\sigma_1, ..., \sigma_N$ arriving at the server, where $\sigma_i = (t_i, l_i, r_i)$ for each i. (t_i is the time of arrival of σ_i.) We visualize the actions of the $\mathscr{WC}$ policy using a bipartite, *rejection graph* $G = (V, E)$, $V = A \cup R$ where A (resp., R) is the set of requests accepted (resp., rejected) by $\mathscr{WC}$, and the edge set E is defined by connecting each rejected request σ_i to the set of accepted requests that caused σ_i to be rejected (i.e., the requests executing at time t_i).

For each $\sigma_i \in R$ let $A(\sigma_i)$ denote the set of (accepted) neighbor nodes of σ_i. Define the *acceptance region* of $R_i \subseteq R$ as $A(R_i) = \bigcup_{\sigma_i \in R_i} A(\sigma_i)$. Our proof considers two different cases for such regions.

- *Fully overlapped Acceptance Regions.* In this case we assume that a set of arriving requests R_i is rejected by a set of running requests A (or a subset of A) *with no new request(s) accepted between rejections*. This situation is depicted in Fig. 1a.

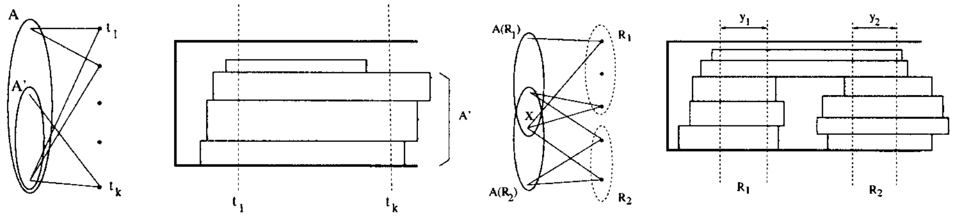

FIG. 1. (a) Fully overlapped case. (b) Partially overlapped case.

Let t_1 and t_k denote the arrival times of the (chronologically) first and last request in R_i, respectively. Also let $A' \subseteq A$ be the subset of requests in A executing at time t_k. By the operation of $\mathscr{WC}$ we know that $\sum_{A'} r_i > \max\{B - r_{max}, r_{min}\}$ (otherwise $\mathscr{WC}$ would have scheduled the request arriving at t_k. Furthermore, our assumptions imply that all requests in A' start before t_1 and complete after t_k. Thus, $t_k - t_1 < \min_{A'}\{l_i\}$. Consequently, the benefit obtained by $\mathscr{WC}$ from A is

$$V_{\mathscr{WC}} = \sum_{A} l_i r_i \geqslant \sum_{A'} l_i r_i > \min_{A'} \{l_i\} \cdot \max\{B - r_{max}, r_{min}\},$$

whereas the loss incurred because of the rejections can be upper-bounded as

$$L_{\mathscr{WC}} \leqslant \underbrace{(t_k - t_1) \cdot \min\{r_{max}, B - r_{min}\}}_{\text{max. loss in } [t_1, t_k)} + \underbrace{B \cdot l_{max}}_{\text{max. loss due to rejections at } t_k}$$
$$< \min_{A'} \{l_i\} \cdot \min\{r_{max}, B - r_{min}\} + B \cdot l_{max}.$$

Thus, if V_{OPT} is the benefit obtained by the optimal off-line scheduler, then

$$\frac{V_{OPT}}{V_{\mathscr{WC}}} \leqslant \frac{V_{\mathscr{WC}} + L_{\mathscr{WC}}}{V_{\mathscr{WC}}} < 1 + \frac{\min_{A'}\{l_i\} \cdot \min\{r_{max}, B - r_{min}\} + B \cdot l_{max}}{\min_{A'}\{l_i\} \cdot \max\{B - r_{max}, r_{min}\}}$$
$$\leqslant 1 + \frac{\min\{r_{max}, B - r_{min}\}}{\max\{B - r_{max}, r_{min}\}} + \frac{B \cdot l_{max}}{\max\{B - r_{max}, r_{min}\} \cdot l_{min}},$$

And, using the identity $\max\{a, b\} + \min\{-a, -b\} = 0$, we obtain $V_{OPT}/V_{\mathscr{WC}} < (1 + l_{max}/l_{min})(B/\max\{B - r_{max}, r_{min}\})$.

• *Partially overlapped Acceptance Regions.* In this case we assume that the acceptance region for a set of rejections can be broken into a collection of $n \geqslant 2$ consecutive acceptance sub-regions $A(R_1)$, ..., $A(R_n)$ where each sub-region rejects some requests with no intermediate arrivals, *but requests scheduled in a sub-region can also extend to future sub-region(s)*. That is, we are allowing new requests to be accepted between the last rejection in R_i and the first rejection of R_{i+1}, and requests in $A(R_i)$ can also "participate" in $A(R_{i+k})$, $k \geqslant 1$ (i.e., the acceptance regions are allowed to partially overlap). This situation is depicted in Fig. 1b. Note that if no such overlapping occurs then we would have multiple independent instances of the fully overlapped case.

We will first prove the competitiveness bound for the case $n=2$ and then extend our proof to cover larger n. Let $X \subseteq A(R_1)$ denote the requests in $A(R_1)$ that extend into $A(R_2)$. Let $l_X = \min_X\{l_i\}$ (the length of the shortest request in X) and let $r_X = \sum_X r_i$ (the total bandwidth requirement of X). Note that the length covered by *all* requests in X is at most l_X and the benefit of X is at least $l_X \cdot r_X$. Also, $l_{min} \leqslant l_X \leqslant l_{max}$ and $r_X \geqslant r_{min}$. Finally, let y_1, y_2 be the length of X's overlap with the rejection regions R_1 and R_2, respectively. (See Fig. 1b.)

Using Fig. 1b it is easy to see that the benefit obtained by $\mathscr{WC}$ is

$$V_{\mathscr{WC}} > (l_{min}-y_1)\cdot\max\{B-r_{max}-r_X, r_{min}\} + (l_{min}-y_2)\cdot\max\{B-r_{max}-r_X, r_{min}\}$$
$$+(y_1+y_2)\cdot\max\{B-r_{max}, r_{min}\} + (l_X-y_1-y_2)\cdot r_X,$$

or, after some arithmetic,

$$V_{\mathscr{WC}} > 2\cdot l_{min}\cdot\max\{B-r_{max}-r_X, r_{min}\} + l_X\cdot r_X$$
$$\geqslant 2\cdot l_{min}\cdot\max\{B-r_{max}, 2r_{min}\} + (l_X-2l_{min})\cdot r_X. \tag{1}$$

Similarly, the loss incurred by the rejections of $\mathscr{WC}$ can be upper-bounded as

$$L_{\mathscr{WC}} < \underbrace{y_1\cdot\min\{r_{max}, b-r_{min}-r_X\} + y_2\cdot\min\{r_{max}, B-r_{min}-r_X\}}_{\text{max. loss during } y_1,\, y_2}$$
$$+\underbrace{(l_X-y_1-y_2)\cdot(B-r_X)}_{\text{max. loss between } y_1 \text{ and } y_2}$$
$$+\underbrace{B\cdot l_{max}}_{\text{max. loss after } y_2} - \underbrace{(l_{min}-y_2)\cdot\max\{B-r_{max}-r_X, r_{min}\}}_{\text{min. benefit of requests in } A(R_2)-X \text{ outside } y_2}$$

which after some arithmetic manipulation gives

$$L_{\mathscr{WC}} < B\cdot(l_X+l_{max}) - l_X\cdot r_X - l_{min}\cdot\max\{B-r_{max}-r_X, r_{min}\}$$
$$\leqslant B\cdot(l_X+l_{max}) - (l_X-l_{min})\cdot r_X - l_{min}\cdot\max\{B-r_{max}, 2r_{min}\}. \tag{2}$$

We now consider the following two cases for l_X:

(1) $l_X > 2\cdot l_{min}$. In this case, Inequalities (1) and (2) give

$$V_{\mathscr{WC}} > 2\cdot l_{min}\cdot\max\{B-r_{max}, 2r_{min}\} \quad\text{and}\quad L_{\mathscr{WC}} < 2\cdot Bl_{max}.$$

Thus $V_{OPT}/V_{\mathscr{WC}} < 1+(B/\max\{B-r_{max}, 2r_{min}\})\cdot(l_{max}/l_{min})$, and the bound clearly holds.

(2) $2\cdot l_{min} \geqslant l_X \geqslant l_{min}$. In this case, let $l_X = \alpha$, where $\alpha\in[1, 2]$. Using Inequality (1) we have

$$V_{\mathscr{WC}} > \alpha\cdot l_{min}\cdot\max\{B-r_{max}-r_X, r_{min}\} + \alpha\cdot l_{min}\cdot r_X + (2-\alpha)\cdot\max\{B-r_{max}-r_X, r_{min}\}$$
$$\geqslant \alpha\cdot l_{min}\cdot\max\{B-r_{max}, 2r_{min}\}.$$

And, combining this with Inequality (2),

$$\frac{V_{OPT}}{V_{\mathscr{WC}}} < 1 + \frac{B}{\max\{B - r_{max}, 2r_{min}\}} + \underbrace{\frac{B \cdot l_{max} - (\alpha - 1)\, l_{min} \cdot r_{min} - l_{min} \cdot \max\{B - r_{max}, 2r_{min}\}}{\alpha \cdot l_{min} \cdot \max\{B - r_{max}, 2r_{min}\}}}_{f(\alpha)}$$

Differentiating $f(\alpha)$, it is easy to see that $\frac{df(\alpha)}{da} < 0$. Thus, $f(\alpha)$ is monotonically decreasing in $\alpha \in [1, 2]$, which implies that

$$\begin{aligned} \frac{V_{OPT}}{V_{\mathscr{WC}}} &< 1 + \frac{B}{\max\{B - r_{max}, 2r_{min}\}} + f(1) \\ &= 1 + \frac{B}{\max\{B - r_{max}, 2r_{min}\}} + \frac{B}{\max\{B - r_{max}, 2r_{min}\}} \cdot \frac{l_{max}}{l_{min}} - 1 \\ &\leqslant \frac{B}{\max\{B - r_{max}, r_{min}\}} \cdot \left(1 + \frac{l_{max}}{l_{min}}\right). \end{aligned}$$

This completes the proof for the case of $n = 2$ partially overlapped acceptance regions. Now consider the case $n > 2$. Let X_i denote the overlap of $A(R_i)$ and $A(R_{i+1})$, for $i = 1, \ldots, n-1$. Similar to our previous notation, let $r_{X_i} = \sum_{X_i} r_j$ and $l_{X_i} = \min_{X_i}\{l_j\} = \alpha_i \cdot l_{min}$, where $1 \leqslant \alpha_i \leqslant l_{max}/l_{min}$. It is not hard to see that Inequality (1) can be extended as

$$\begin{aligned} V_{\mathscr{WC}} > l_{min} \cdot \sum_{i=1}^{n-1} \max\{B - r_{max} - r_{X_i}, r_{min}\} + l_{min} \cdot \sum_{i=1}^{n-1} \alpha_i \cdot r_{X_i} \\ + l_{min} \cdot \max\{B - r_{max} - r_{X_{n-1}}, r_{min}\}, \end{aligned}$$

where the last term in the sum captures the (minimum possible) contribution of the last sub-region. After some manipulation the above inequality gives

$$V_{\mathscr{WC}} > n \cdot l_{min} \max\{B - r_{max}, 2r_{min}\} + l_{min} \cdot \sum_{i=1}^{n-1} (\alpha_i - 1) \cdot r_{X_i} - l_{min} \cdot r_{X_{n-1}}. \tag{3}$$

And, using a method similar to that used for the case $n = 2$, we can derive the following inequality for the loss incurred by $\mathscr{WC}$ (extension of Inequality (2)):

$$\begin{aligned} L_{\mathscr{WC}} < B \cdot l_{max} + B \cdot l_{min} \cdot \sum_{i=1}^{n-1} \alpha_i - r_{min} \cdot l_{min} \cdot \sum_{i=1}^{n-1} (\alpha_i - 1) \\ - (n-1) \cdot l_{min} \max\{B - r_{max}, 2r_{min}\}. \end{aligned} \tag{4}$$

Again, we consider two cases depending on the value of α_{n-1} (i.e., the length $l_{X_{n-1}}$).

(1) $\alpha_{n-1} > 2$. Then, Inequalities (3) and (4) give

$$V_{\mathscr{WC}} > n \cdot l_{min} \cdot \max\{B - r_{max}, 2r_{min}\} \qquad \text{and} \qquad L_{\mathscr{WC}} < n \cdot B l_{max}.$$

Thus $V_{OPT}/V_{\mathcal{WC}} < 1+(B/\max\{B-r_{max}, 2r_{min}\})\cdot(l_{max}/l_{min})$ and the bound clearly holds.

(2) $2 \geqslant \alpha_{n-1} \geqslant 1$. Then, Inequality (3) gives

$$V_{\mathcal{WC}} > (n-1)\cdot l_{min}\cdot \max\{B-r_{max}, 2r_{min}\} + l_{min}\cdot r_{min}\cdot \sum_{i=1}^{n-1}(\alpha_i - 1).$$

Combining this with Inequality (4) gives

$$\frac{V_{OPT}}{V_{\mathcal{WC}}} < \frac{B\cdot l_{max} + B\cdot l_{min}\cdot \sum_{i=1}^{n-1}\alpha_i}{(n-1)\cdot l_{min}\cdot \max\{B-r_{max}, 2r_{min}\} + l_{min}\cdot r_{min}\cdot \sum_{i=1}^{n-1}(\alpha_i-1)}$$
$$\leqslant \frac{(n-1)\cdot B\cdot l_{max} + \alpha_{n-1}\cdot B\cdot l_{min}}{(n-1)\cdot l_{min}\cdot \max\{B-r_{max}, 2r_{min}\}} \qquad \left(\text{since } \alpha \leqslant \frac{l_{max}}{l_{min}}\right)$$

or, equivalently

$$\frac{V_{OPT}}{V_{\mathcal{WC}}} < \frac{B}{\max\{B-r_{max}, 2r_{min}\}}\cdot\frac{l_{max}}{l_{min}} + \frac{B}{\max\{B-r_{max}, 2r_{min}\}}\cdot\frac{\alpha_{n-1}}{n-1}$$
$$\leqslant \frac{B}{\max\{B-r_{max}, r_{min}\}}\cdot\left(1+\frac{l_{max}}{l_{min}}\right)n$$

since $n-1 \geqslant 2 \geqslant \alpha_{n-1}$ for each $n > 2$.

This completes the proof for the case of multiple partially overlapped acceptance regions.

For the general bound, observe that the behavior of $\mathcal{WC}$ over any incoming sequence of requests can be seen as a sequence of independent (i.e., non-overlapping) execution segments, where each such segment consists of either fully overlapped or partially overlapped acceptance regions. Thus, for each such execution segment s we have shown that the ratio of the maximum possible loss to the benefit obtained by $\mathcal{WC}$ is $L^S_{\mathcal{WC}}/V^S_{\mathcal{WC}} < (B/\max\{B-r_{max}, r_{min}\})\cdot(1+l_{max}/l_{min})$. Thus, for the entire sequence of segments the ratio of loss to benefit is

$$\frac{L_{\mathcal{WC}}}{V_{\mathcal{WC}}} \leqslant \frac{\sum_s L^S_{\mathcal{WC}}}{\sum_s V^S_{\mathcal{WC}}} \leqslant \max_s\left\{\frac{L^S_{\mathcal{WC}}}{V^S_{\mathcal{WC}}}\right\} < \frac{B}{\max\{B-r_{max}, r_{min}\}}\cdot\left(1+\frac{l_{max}}{l_{min}}\right) - 1.$$

The result follows directly from this last inequality. This completes the proof. ∎

Thus, allowing variability along the second dimension (i.e., bandwidth) multiplies the competitiveness of $\mathcal{WC}$ by a factor that depends on B, r_{max}, and r_{min}. Intuitively, this term captures the effects of the worst-case bandwidth loss due to fragmentation. If $B \geqslant r_{max} + r_{min}$, as will usually be the case for CM servers and requests, then the following corollary applies.

COROLLARY 4.1. *If $B \geqslant r_{max} + r_{min}$ then $\mathcal{WC}$ is $\frac{1+\Delta}{1-\rho}$-competitive, where $\rho = r_{max}/B$.*

Note that the competitiveness bound of $\frac{1+\Delta}{1-\rho}$ is, in fact, valid regardless of the relative sizes of B and $r_{max}+r_{min}$. Theorem 4.2 just gives a tighter bound when $B<r_{max}+r_{min}$. Also, note that for typical CM numbers the fraction ρ is much smaller than unity. For example, even for the relatively high MPEG-2 rate requirements of 6–8 Mbps, a CM server with a low-end RAID can sustain 40–60 concurrent streams [25, 31]. The denominator in our competitiveness bound agrees with the bounds given by Bar-Noy *et al.* [7] for the *preemptive* version of the problem. (They avoid dependence on Δ through clever use of the preemption mechanism.)

4.2. Lower Bounds

In this section, we prove lower bounds on the competitive ratio of any *deterministic or randomized* algorithm for on-line admission control. Our results demonstrate the existence of an exponential gap between the competitive ratio of $\mathcal{WC}$ and the lower bound on the competitive ratio of any deterministic or randomized algorithm. This clearly suggests the possibility for improvement by using non-greedy schemes. We propose such schemes with near-optimal competitiveness in Section 4.3.

Once again, let us start with the identical bandwidth case (i.e., $r_i=r$ for all i). A simple adversary argument shows that for the case of a single bandwidth channel (i.e., the OIS problem [20]), there is a lower bound of $1+\Delta$ on the competitiveness of any deterministic scheduler. This argument fails when the number of channels is increased. However, as the following theorem shows, no deterministic *or* randomized admission control scheme can be better than $\Omega(\log\Delta)$ competitive.

THEOREM 4.3 (Lower Bound, Identical Bandwidth Case). *Any deterministic* or *randomized on-line admission control algorithm for CM requests with identical bandwidth requirements has a competitive ratio of* $\Omega(\log\Delta)$.

Proof. (1) *Deterministic Lower Bound.* Let A be any (deterministic) scheduling algorithm and let x denote the competitive ratio of A. We will prove that $x\geqslant O(\log\Delta)$ using an adversary argument. The basic idea in the proof is that the adversary presents A with a sequence of requests that forces A to fill up its channels with "low profit" (i.e., small duration) requests in order to maintain its competitive ratio.

More specifically, the adversary presents A with a sequence of Δ "request batches" $B_1, \ldots, B_\Delta$. (To simplify the presentation we assume that Δ is an integer.) Batch B_i consists of c (= number of channels) requests of length $i\cdot l_{min}$ arriving at time $(i-1)\cdot\varepsilon$, where ε is some arbitrarily small interval of time. Note that, since all these requests are pairwise overlapping, only c requests can be scheduled. Clearly the optimal (off-line) strategy is to schedule the requests in B_Δ, accumulating a total profit of $c\cdot\Delta\cdot l_{min}=c\cdot l_{max}$.

Consider the on-line operation of A. Let n_i denote the number of requests in B_i scheduled by A. In order to maintain its competitive ratio of x, A must schedule *at least* $\frac{c}{x}$ requests from B_1. (Otherwise, the adversary could stop the request sequence after B_1 and force A to have a competitive ratio worse than x.) Thus $n_1\geqslant\frac{c}{x}$.

Similarly, in order to maintain a competitiveness of x after B_2, the following inequality must be satisfied:

$$x \cdot (n_1 \cdot l_{min} + n_2 \cdot 2 \cdot l_{min}) \geqslant c \cdot 2 \cdot l_{min}.$$

It is easy to see that for A to satisfy the above equation (subject to the constraint $n_1 \geqslant \frac{c}{x}$) and, at the same time, minimize the total number of channels used $n_1 + n_2$ (thus allowing more profit from future larger requests), n_1 has to take its minimum value, i.e., $n_1 = \frac{c}{x}$. Substituting gives

$$c \cdot l_{min} + x \cdot n_2 \cdot 2 \cdot l_{min} \geqslant c \cdot 2 \cdot l_{min},$$

or, equivalently: $n_2 \geqslant \frac{c}{2 \cdot x}$. A simple inductive argument along the same lines can show the following claim.

CLAIM. *In order to maintain a competitive ratio of x and maximize its total profit, algorithm A must schedule* $n_i = \frac{c}{i \cdot x}$ *requests from batch* B_i *for each* $i = 1, \ldots, \Delta$.

A can then use its remaining channels to schedule requests from the final (and, most profitable) batch. Let $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$ (the nth-Harmonic number). Note that if, at any point during this "game," A exhausts its channels then $V_A < \sum_{i=1}^{\Delta} \frac{c}{i \cdot x} \cdot i \cdot l_{min} = \frac{c}{x} \cdot \Delta \cdot l_{min}$, and thus

$$\frac{V_{OPT}}{V_A} > \frac{c \cdot \Delta \cdot l_{min}}{\frac{c}{x} \cdot \Delta \cdot l_{min}} = x,$$

which is impossible since A has a competitive ratio of x. Thus, we must have

$$\frac{c}{x} \cdot \sum_{i=1}^{\Delta} \frac{1}{i} \leqslant c, \qquad \text{or} \qquad \text{equivalently, } x \geqslant H_\Delta = \ln \Delta + O(1) = O(\log \Delta).$$

This completes the proof for the deterministic lower bound.

(2) *(Oblivious) Randomized Lower Bound.* Our proof is based on the application of *Yao's result* to competitive analysis [17 23]. Briefly, this result states that the lower bound on the oblivious competitive ratio for a given problem is greater than the lower bound on the competitive ratio of *deterministic* on-line algorithms, when the request sequences for the problem are restricted to a distribution.

Our methodology is as follows. We construct a probability distribution $D_{\bar{\sigma}}$ over the request sequences $\bar{\sigma}$, and based on that randomized input sequence we provide:

1. A lower bound (L) for $E_{D_{\bar{\sigma}}}[V_{OPT}(\bar{\sigma})]$, the expected benefit accepted by the optimal off-line algorithm; and,

2. An upper bound (U) for $E_{D_{\bar{\sigma}}}[V_A(\bar{\sigma})]$, the expected benefit accrued by *any* deterministic on-line algorithm for the problem.

If κ^b denotes the oblivious competitive ratio of *any* randomized algorithm, then, by Yao's result [17, 23], $\kappa^b \geqslant \frac{L}{U}$.

First, we define the probability distribution $D_{\bar{\sigma}}$ over request sequences. Let c denote the number of playback channels at the server and, without loss of generality, assume $\log \Delta$ is integer. We assume that requests arrive in $\log \Delta + 1$ batches of c requests $B_0, \ldots, B_{\log \Delta}$ with the time separating the arrivals being a very small $\varepsilon > 0$. Furthermore, all requests in batch B_i have length equal to $2^i \cdot l_{min}$ and batches arrive according to the following probabilities:

$$P[B_0 \text{ arrives}] = 1 \qquad \text{and} \qquad P[B_i \text{ arrives} \mid B_{i-1} \text{ arrives}] = \tfrac{1}{2}.$$

Thus, for $i \in \{0, \ldots, \log \Delta\}$, the probability of the arrival sequence $B_0 \cdots B_i$ is 2^{-i}.

Next, we provide a lower bound (L) for $E_{D_{\bar{\sigma}}}[V_{OPT}(\bar{\sigma})]$. Consider an off-line strategy that always accepts the requests in the *last* arriving batch. The expected benefit of that strategy is

$$\sum_{i=0}^{\log \Delta} P[B_0 \cdots B_i] \cdot c \cdot 2^i \cdot l_{min} = c \cdot l_{min} \cdot (\log \Delta + 1).$$

Thus $E_{D_{\bar{\sigma}}}[V_{OPT}(\bar{\sigma})] \geqslant c \cdot l_{min} \cdot (\log \Delta + 1) \equiv L$.

Finally, we provide an upper bound (U) for $E_{D_{\bar{\sigma}}}[V_A(\bar{\sigma})]$ for all deterministic on-line algorithms A. Let $B(i, k)$ denote the *maximum expected benefit* from batches $B_i, \ldots, B_{\log \Delta}$, where the maximization is taken over all possible ways to accept requests from $B_0, \ldots, B_{i-1}$ so that *at most* $k \leqslant c$ channels are free. We can bound $B(i, k)$ with the following recurrence relation, where the first term in the max{ }, represents the benefit from requests in B_i (with l denoting the number of requests accepted from batch B_i) and the second term represents the maximum expected benefit from requests in batches $B_{i+1}, \ldots, B_{\log \Delta}$, given that $k - l$ channels are available,

$$B(i, k) \leqslant \max_{l \leqslant k} \{ l \cdot l_{min} \cdot 2^i, \tfrac{1}{2} B(i+1, k-l) \},$$

with the initial condition $B(\log \Delta, k) \leqslant k \cdot 2^{\log \Delta} \cdot l_{min} = k \cdot l_{max}$. Note that the factor $\frac{1}{2}$ in front of the second *max*{ } term comes from the fact that the probability of B_{i+1} arriving given that B_i has arrived is $\frac{1}{2}$.

A simple induction on $j = \log \Delta - i$ shows that for each $i \in \{0, \ldots, \log \Delta\}$, $B(i, k) \leqslant k \cdot 2^i \cdot l_{min}$. Thus, for any deterministic algorithm

$$E_{D_{\bar{\sigma}}}[V_A(\bar{\sigma})] \leqslant B(0, c) \leqslant c \cdot l_{min} \equiv U.$$

Consequently, by Yao's result, the oblivious competitive ratio of any randomized on-line algorithm is $\kappa^b \geqslant \frac{L}{U} = \log \Delta + 1$. This completes the proof for the randomized lower bound. ∎

Similar lower bounds on the competitive ratio hold for the variable bandwidth case. Again, the effect of bandwidth fragmentation introduces a multiplicative factor of $\frac{1}{1-\rho}$.

THEOREM 4.4 (Lower Bounds, Variable Bandwidth Case). *Consider a sequence of CM requests with variable bandwidth requirements. Then:*

(1) *Any* deterministic *on-line admission control algorithm has a competitive ratio of* (a) $\Omega(\frac{\rho\cdot(1+\Delta)}{1-\rho})$*, if* $\rho>\frac{1}{2}$*; and* (b) $\Omega(\frac{\log\Delta}{1-\rho})$*, otherwise.*

(2) *Any* randomized *on-line admission control algorithm has a competitive ratio of,* $\Omega(\frac{\log\Delta}{1-\rho})$ *if* $\rho\leqslant\frac{1}{2\log\Delta}$.

Proof. (1) *Deterministic Lower Bound.* Case (a) can be shown by a simple construction. For Case (b), assume that $\frac{1}{k}\geqslant\rho>\frac{1}{k+1}$ where $k\geqslant 2$ is an integer. The adversary constructs the sequence of requests in a manner that is very similar to the proof of the deterministic lower bound in Theorem 4.3, but it also exploits the bandwidth variability to ensure that the on-line algorithm will end up using only a fraction $\frac{k}{k+1+\delta}\approx 1-\rho$ of the available bandwidth B, where $\delta>0$ is arbitrarily small. Similar to the one-dimensional case, we can show that the on-line algorithm must have a competitive ratio of at least H_Δ *within that portion of the server's bandwidth*, i.e.,

$$V_{OPT}^{partial}\geqslant H_\Delta\cdot V_A.$$

But the optimal off-line scheduler can avoid this bandwidth fragmentation and schedule the entire bandwidth B, thus $V_{OPT}\geqslant\frac{k+1+\delta}{k}$, and the competitive factor must be at least $H_\Delta\cdot\frac{k+1+\delta}{k}\approx\frac{\log\Delta}{1-\rho}$.

(2) (*Oblivious*) *Randomized Lower Bound.* We follow the same methodology as in the proof of the randomized lower bound for Theorem 4.3 but taking into account the worst-case bandwidth fragmentation due to variable bandwidth requests. We assume that the bandwidth B of the server is $B=k\cdot(r_{max}-\delta)$, where $\delta>0$ is an arbitrarily small positive constant, and we consider the following request batches, which (as in the proof of Theorem 4.3) arrive with very small separation in time:

- For $i\in\{0,\dots,\log\Delta\}$, batch B_i consists of $k-1$ requests with bandwidth requirement r_{max} and length $2^i\cdot l_{min}$; and,
- Batch $B_{\log\Delta+1}$ consists of k requests with bandwidth requirement $r_{max}-\delta$ and length $2^{\log\Delta}\cdot l_{min}=l_{max}$.

Also, as in the proof of Theorem 4.3, $P[B_0 \text{ arrives}]=1$ and $P[B_i \text{ arrives}\mid B_{i-1} \text{ arrives}]=\frac{1}{2}$. Thus, for $i\in\{0,\dots,\log\Delta+1\}$, the probability of the arrival sequence $B_0\cdots B_i$ is 2^{-i}.

For the lower bound (L) on $E_{D_{\bar\sigma}}[V_{OPT}(\bar\sigma)]$, observe that the off-line strategy that always accepts the last batch will have an expected benefit:

$$\sum_{i=0}^{\log\Delta}P[B_0\cdots B_i]\cdot(k-1)\cdot r_{max}\cdot l_{min}\cdot 2^i+k\cdot(r_{max}-\delta)\cdot 2^{-\log\Delta-1}\cdot l_{min}\cdot 2^{\log\Delta}$$
$$=(k-1)\cdot l_{min}\cdot r_{max}\cdot(\log\Delta+1)+\tfrac{1}{2}\cdot k\cdot(r_{max}-\delta)\cdot l_{min}\equiv L.$$

We now provide an upper bound (U) on $E_{D_{\bar\sigma}}[V_A(\bar\sigma)]$ for all deterministic on-line algorithms A. As in the proof of Theorem 4.3, let $B(i,m)$ denote the *maximum* *expected benefit* from batches $B_i, \ldots, B_{\log \Delta+1}$ where m now denotes the number of "channels" of bandwidth r_{max} that are left in the server after batches $B_0, \ldots, B_{i-1}$. Note that $m \leqslant k-1$ by our definitions. We can then bound $B(i, m)$ using the recurrence:

$$B(i, m) \leqslant \begin{cases} \max\limits_{l \leqslant m} \{l \cdot r_{max} \cdot l_{min} \cdot 2^i + \frac{1}{2} B(i+1, m-l)\}, & \text{if } i \leqslant \log \Delta \\ m \cdot (r_{max} - \delta) \cdot l_{min} \cdot 2^{\log \Delta}, & \text{if } i = \log \Delta + 1 \text{ and } m < k-1 \\ k \cdot (r_{max} - \delta) \cdot l_{min} \cdot 2^{\log \Delta}, & \text{if } i = \log \Delta + 1 \text{ and } m = k-1. \end{cases}$$

Note that the last two clauses in the above expression follow from the observation that once a single request with bandwidth requirement r_{max} is scheduled, the total number of requests that can be scheduled is at most $k-1$ (whereas the entire last batch of k requests can be scheduled otherwise).

Using induction on $j = \log \Delta + 1 - i$, it is easy to show that $B(i, m) \leqslant m \cdot r_{max} \cdot l_{min} \cdot 2^i$ for all $i \in \{0, \ldots, \log \Delta + 1\}$. Thus, for any deterministic algorithm A, $E_{D_{\bar{\sigma}}}[V_A(\bar{\sigma})] \leqslant B(0, k-1) \leqslant (k-1) \cdot r_{max} \cdot l_{min} \equiv U$.

By Yao's result, the oblivious competitive ratio of any randomized on-line algorithm is

$$\kappa^b \geqslant \frac{L}{U} = \frac{(k-1) \cdot r_{max} \cdot (\log \Delta + 1) + \frac{1}{2} B}{(k-1) \cdot r_{max}}.$$

By our assumption $B > 2 \cdot r_{max} \cdot \log \Delta$, so the above inequality gives

$$\kappa^b \geqslant \frac{k}{k-1} \cdot \log \Delta \approx \frac{\log \Delta}{1-\rho},$$

since $k \approx \frac{1}{\rho}$ by our choice of parameters. This completes the proof. ∎

We should note that Awerbuch *et al.* [3] also proved an $\Omega(\log \Delta)$ lower bound for *deterministic* on-line circuit routing in the case of requests with identical bandwidth requirements. However, our lower bounds for the more general variable bandwidth case also demonstrate the effect of the maximum bandwidth demand (ρ) which was not factored into their results. Furthermore, we have shown that the logarithmic lower bounds cannot be improved upon through the use of randomization.

4.3. Bandwidth Prepartitioning Policies

We now propose novel deterministic admission control policies that guarantee *near-optimal* competitive ratios under the (mild) restriction that the maximum fraction of server bandwidth demanded by a request does not exceed $1/\lceil \log \Delta \rceil$. Our policies are based on *prepartitioning* the bandwidth capacity of a CM server among requests of different length. Roughly speaking, the basic idea of the bandwidth prepartitioning schemes is to isolate requests with large differences in length, thus

Simple Bandwidth Prepartitioning

1. Divide the available bandwidth B into $\lceil \log \Delta \rceil$ partitions $B_1, \ldots, B_{\lceil \log \Delta \rceil}$, where the size of the i^{th} partition is $|B_i| = \frac{B}{\lceil \log \Delta \rceil}$.
2. For each arriving request $\sigma_j = (t_j, l_j, r_j)$

   2.1 Let $i \in \{1, \ldots, \lceil \log \Delta \rceil\}$ be such that: $2^{i-1} \cdot l_{min} \leq l_j < 2^i \cdot l_{min}$ (allowing for $l_j = 2^i \cdot l_{min}$ if $i = \lceil \log \Delta \rceil$).

   2.2 If the amount of free bandwidth in partition B_i is less than r_j, then reject σ_j;
   Otherwise, schedule σ_j in partition B_i;

FIG. 2. Algorithm $\mathscr{SBP}$.

ensuring that short requests cannot "steal' the entire server bandwidth from longer (and, more profitable) requests.[6]

The first policy we introduce is termed *Simple Bandwidth Prepartitioning* ($\mathscr{SBP}$) and is depicted in Fig. 2.[7] The $\mathscr{SBP}$ algorithm exploits the server's knowledge of the Δ ratio by classifying requests to channel groups based on their length and then using a $\mathscr{WC}$ policy within each group. The idea is that by classifying the requests into different partitions according to their length range, we are ensuring that the maximum to minimum length ratio is bounded by a constant *within each partition.*

The following theorem shows that this simple prepartitioning scheme results in a significant improvement in the competitive ratio for CM servers with bandwidth B larger than $r_{max} \cdot \lceil \log \Delta \rceil$. This requirement is typically satisfied by today's servers, even for large values of r_{max} and Δ. For example, if $r_{max} = 8$ Mbps and $l_{max} = 120 \cdot l_{min}$, then $r_{max} \cdot \lceil \log \Delta \rceil = 56$ Mbps, i.e., less than the transfer rate of a *single* high-end magnetic disk [26].

THEOREM 4.5 (Competitiveness of $\mathscr{SBP}$). *Assume $\rho = r_{max}/B < 1/\lceil \log \Delta \rceil$ (or, equivalently, $c > \lceil \log \Delta \rceil$ for the identical bandwidth case). Then, the $\mathscr{SBP}$ admission control policy is:*

(1) $3 \cdot \lceil \log \Delta \rceil$*-competitive for the identical bandwidth case; and,*

(2) $\frac{3 \cdot \lceil \log \Delta \rceil}{1 - \rho \cdot \lceil \log \Delta \rceil}$*-competitive for the variable bandwidth case.*

Proof. (1) *Identical Bandwidth Case.* We first give a definition of *strong competitiveness* as defined by Bar-Noy *et al.* [7]. Given an input $\bar{\sigma} = \sigma_1, \sigma_2, \ldots, \sigma_N$ where $\sigma_i = (t_i, l_i, r_i)$, it is easy to see that the bandwidth required by $\bar{\sigma}$ at time t is $B_{\bar{\sigma}}(t) = \sum_{\sigma_i : t \in [t_i, t_i + l_i]} r_i$. We say that $\bar{\sigma}$ is *feasible* if $B_{\bar{\sigma}}(t) \leqslant B$ for all times t. We define the *cover* of a sequence $\bar{\sigma}$ as

$$V(\bar{\sigma}) = \int_t \min\{B_{\bar{\sigma}}(t), B\}\, dt.$$

[6] Note that if $\Delta = 1$ (i.e., all requests have identical lengths) then simple $\mathscr{WC}$ offers optimal competitiveness. Thus, we will assume that $\Delta > 1$ or, equivalently, $\log \Delta > 0$ in the remainder of this paper.

[7] We describe our policies in terms of the more general variable bandwidth case. The restriction to identical bandwidth requests should be straightforward.

Note that if $\bar{\sigma}$ is feasible then $V(\bar{\sigma})$ is exactly the total throughput (i.e., length-rate product) in $\bar{\sigma}$. We say that an on-line algorithm A is *strongly* k-competitive if

$$\sup_{\bar{\sigma}} \frac{V(\bar{\sigma})}{V_A(\bar{\sigma})} \leqslant k.$$

Clearly, any strongly k-competitive algorithm is also k-competitive (Section 3).

Observe that the analysis in the proof of Theorem 4.1 actually establishes that $\mathscr{WC}$ is *strongly* $(1+\Delta)$-competitive. Consider an input sequence $\bar{\sigma}$. Fix a particular group of channels C_i ($|C_i| = \frac{B}{\lceil \log \Delta \rceil}$) and let $\bar{\sigma}_i$ denote the subsequence of $\bar{\sigma}$ with lengths in the range $2^{i-1} \cdot l_{min} \leqslant l_j < 2^i \cdot l_{min}$, for any $i \in \{1, ..., \lceil \log \Delta \rceil\}$. Since the operation of $\mathscr{SBP}$ on $\bar{\sigma}_i$ is identical to $\mathscr{WC}$ using only the channels in C_i we have

$$\left(1+\left(\frac{l_{max}}{l_{min}}\right)_{\bar{\sigma}_i}\right) \cdot V_{\mathscr{SBP}}(\bar{\sigma}_i) \geqslant \int_t \min\left\{B_{\bar{\sigma}_i}(t), \frac{B}{\lceil \log \Delta \rceil}\right\} dt.$$

Observe that for all requests in $\bar{\sigma}_i$ we have $(\frac{l_{max}}{l_{min}})_{\bar{\sigma}_i} \leqslant 2$. Thus the above inequality gives

$$3 \cdot \lceil \log \Delta \rceil \cdot V_{\mathscr{SBP}}(\bar{\sigma}_i) \geqslant \int_t \min\{B_{\bar{\sigma}_i}(t), B\} \, dt.$$

It is easy to see that the left-hand side of the above inequality is an upper bound on the benefit that *any* scheduler can obtain from the requests in $\bar{\sigma}_i$ (even when using all available bandwidth). Thus, we have shown that $3 \cdot \lceil \log \Delta \rceil \cdot V_{\mathscr{SBP}}(\bar{\sigma}_i) \geqslant V_{OPT}(\bar{\sigma}_i)$, where OPT is the optimal clairvoyant scheduler using the entire server. Clearly, $V_{\mathscr{SBP}}(\bar{\sigma}) = \sum_i V_{\mathscr{SBP}}(\bar{\sigma}_i)$ and $V_{OPT}(\bar{\sigma}) \leqslant \sum_i V_{OPT}(\bar{\sigma}_i)$. Thus, $3 \cdot \lceil \log \Delta \rceil \cdot V_{\mathscr{SBP}}(\bar{\sigma}) \geqslant V_{OPT}(\bar{\sigma})$ for any sequence $\bar{\sigma}$. The result follows.

(2) *Variable Bandwidth Case*. Again, the proof for the case of variable bandwidth requests is based on the observation that the proof procedure for Theorem 4.2 actually establishes that $\mathscr{WC}$ is *strongly* $\frac{1+\Delta}{1-\rho}$-competitive, where $\rho = r_{max}/B$. The proof then proceeds along the same lines as the proof for the identical bandwidth case. ∎

Thus, by merely isolating different length ranges, the $\mathscr{SBP}$ admission control policy improves the competitiveness of $\mathscr{WC}$ from linear to logarithmic in Δ, at least for the identical bandwidth case. The main idea behind $\mathscr{SBP}$ is that in order to be competitive under a worst-case scenario, the scheduler should not allow short duration requests to monopolize the server's bandwidth. However, $\mathscr{SBP}$ can also suffer from bandwidth fragmentation in the variable bandwidth case. In the worst case, bandwidth approximately equal to r_{max} is lost *in each partition*, leading to a total bandwidth loss of $r_{max} \cdot \lceil \log \Delta \rceil$ in the server. Intuitively, we would like to be able to "combine" these bandwidth. fragments to allow for incoming requests to be scheduled across partitions, especially if these requests are long since this implies more guaranteed profit.

The *Down-shift Bandwidth Prepartitioning* ($\mathscr{DBP}$) policy depicted in Fig. 3 is based exactly on these observations. As in $\mathscr{SBP}$, the $\mathscr{DBP}$ algorithm also prohibits short requests from monopolizing the server, *but* it also allows longer (and

Down-shift Bandwidth Prepartitioning

1. Divide the available bandwidth B into $\lceil \log \Delta \rceil$ partitions $B_1, \ldots, B_{\lceil \log \Delta \rceil}$, where the size of the i^{th} partition is $|B_i| = \frac{B}{\lceil \log \Delta \rceil}$.

2. For each arriving request $\sigma_j = (t_j, l_j, r_j)$

2.1 Let $i \in \{1, \ldots, \lceil \log \Delta \rceil\}$ be such that: $2^{i-1} \cdot l_{min} \leq l_j < 2^i \cdot l_{min}$ (allowing for $l_j = 2^i \cdot l_{min}$ if $i = \lceil \log \Delta \rceil$).

2.2 If the total amount of free bandwidth in $B_1 \cup \ldots \cup B_i$ is less than r_j, then reject σ_j;
Otherwise, schedule σ_j using available bandwidth from partitions $B_i, B_{i-1}, \ldots, B_1$ *in that order.*

FIG. 3. Algorithm $\mathscr{DBP}$.

thus, more profitable requests) to be "down-shifted" to lower groups and steal unused bandwidth that would otherwise be dedicated to shorter requests. Theorem 4.6 shows that incorporating this change does not compromise logarithmic competitiveness.

THEOREM 4.6 (Competitiveness of $\mathscr{DBP}$). *Assume $\rho = r_{max}/B < 1/\lceil \log \Delta \rceil$ (or, equivalently, $c > \lceil \log \Delta \rceil$ for the identical bandwidth case). Then, the $\mathscr{DBP}$ admission control policy is:*

(1) $(1 + 8 \cdot \lceil \log \Delta \rceil)$*-competitive for the identical bandwidth case; and,*

(2) $(1 + \frac{8 \cdot \lceil \log \Delta \rceil}{1 - \rho \cdot \lceil \log \Delta \rceil})$*-competitive for the variable bandwidth case.*

Proof. (1) *Identical Bandwidth Case.* We partition the schedule derived by $\mathscr{DBP}$ along the time axis into consecutive intervals $I_1, \ldots, I_M$ of length l_{max} (or, more accurately $2^{\lceil \log \Delta \rceil} \cdot l_{min}$) each. Let A_j denote the set of requests accepted by $\mathscr{DBP}$ inside interval I_j (i.e., accepted requests whose starting point is in I_j), and let R_j be the set of requests rejected by $\mathscr{DBP}$ *and* accepted by the optimal scheduler in I_j. We use s_j to denote the *saturation level of* I_j: the largest i such that for some point in time $t \in I_j$ *all* the channels in $C_1 \cup \cdots C_i$ are busy at time t. Note that by the operation of $\mathscr{DBP}$, only requests of length less than $2^{s_j} \cdot l_{min}$, will be rejected in I_j. Finally, let $V(S)$ denote the total benefit (i.e., rate-length product) of a set of requests S.

Fix a specific interval I_j and let $k = \lceil \log \Delta \rceil$. Partition the set R_j into $R_j^1 \cup \cdots \cup R_j^{s_j}$, where R_j^i is the set of requests in R_j with lengths in the range $[2^{i-1} \cdot l_{min}, 2^i \cdot l_{min})$. Fix a specific $i \in \{1, \ldots, s_j\}$ and slice the interval I_j into 2^{k-i} subintervals of length $2^i \cdot l_{min}$ each. Consider the requests in R_j^i rejected in such a subinterval. Clearly, the maximum benefit that any scheduler could obtain from these requests is $c \cdot 2^i \cdot l_{min}$, by allowing a "batch" of c requests of maximum length $(2^i \cdot l_{min})$. But, by the operation of $\mathscr{DBP}$, since these requests were rejected $\mathscr{DBP}$ must have already accepted a benefit of

$$\frac{c}{\lceil \log \Delta \rceil} \cdot \sum_{j=1}^{i} 2^{j-1} \cdot l_{min} = \frac{c}{\lceil \log \Delta \rceil} \cdot (2^i - 1) \cdot l_{min}.$$

Furthermore, by the operation of the algorithm, this benefit will be distinct for each "batch" of rejections within different sub-intervals. Thus, the *maximum loss-to-benefit ratio* within each such sub-interval is

$$\frac{c \cdot 2^i \cdot l_{min}}{\frac{c}{\lceil \log \Delta \rceil} \cdot (2^i - 1) \cdot l_{min}} = \lceil \log \Delta \rceil \cdot \frac{2^i}{2^i - 1} \leqslant 2 \cdot \lceil \log \Delta \rceil, \qquad \textit{independent of } i.$$

Thus, if we let L_j denote the loss that the $\mathscr{DBP}$ scheme incurs within the interval I_j, then

$$V(A_j) + V(A_{j-1}) \geqslant \max \left\{ \frac{c}{\lceil \log \Delta \rceil} \cdot (2^{s_j} - 1) \cdot l_{min}, \frac{L_j}{2 \cdot \lceil \log \Delta \rceil} \right\},$$

where the first term in the max{ } follows from the definition of the saturation level of I_j. Also, it is easy to see that

$$V(R_j) \leqslant L_j + c \cdot 2^{s_j} \cdot l_{min},$$

where the second term captures the maximum possible loss due to rejections at the end of I_j. Combining the last two inequalities, we have

$$V(A_j) + V(A_{j-1}) \geqslant \max \left\{ \frac{c}{2 \cdot \lceil \log \Delta \rceil} \cdot 2^{s_j} \cdot l_{min}, \frac{L_j}{2 \cdot \lceil \log \Delta \rceil} \right\}$$

$$\geqslant \frac{1}{2 \cdot \lceil \log \Delta \rceil} \cdot \frac{c \cdot 2^{s_j} \cdot l_{min} + L_j}{2} \geqslant \frac{V(R_j)}{4 \cdot \lceil \log \Delta \rceil}.$$

Thus, summing over all intervals I_j we have $V_{\mathscr{DBP}}(\bar{\sigma}) \geqslant \sum_j V(R_j)/8 \cdot \lceil \log \Delta \rceil$.

Observe that for the optimal clairvoyant scheduler A^*, $V_{A^*}(\bar{\sigma}) \leqslant V_{\mathscr{DBP}}(\bar{\sigma}) + \sum_j V(R_j)$, or, using the above inequality, $V_{A^*}(\bar{\sigma}) \leqslant V_{\mathscr{DBP}} \cdot (1 + 8 \cdot \lceil \log \Delta \rceil)$. This completes the proof for the identical bandwidth case.

(2) *Variable Bandwidth Requests*. The proof proceeds along the same lines as for the identical bandwidth case. We now define the *saturation level* s_j of interval I_j as the largest i such that for some point in time $t \in I_j$ the *total available bandwidth* in partitions $B_1 \cup \cdots B_i$ is less than r_{max}. Note that by the operation of $\mathscr{DBP}$, only requests of length less than $2^{s_j} \cdot l_{min}$ can be rejected in I_j.

Fix a specific interval I_j and let $k = \lceil \log \Delta \rceil$. Partition the set R_j into $R_j^1 \cup \cdots \cup R_j^{s_j}$, where R_j^i is the set of requests in R_j with lengths in the range $[2^{i-1} \cdot l_{min}, 2^i \cdot l_{min})$. Fix a specific $i \in \{1, ..., s_j\}$ and slice the interval I_j into 2^{k-i} sub-intervals of length $2^i \cdot l_{min}$ each. Consider the requests in R_j^i rejected in such a sub-interval. Clearly, the maximum benefit that any scheduler could obtain from these requests is $B \cdot 2^i \cdot l_{min}$. But, by the operation of $\mathscr{DBP}$, since these requests were rejected $\mathscr{DBP}$ must have already accepted a benefit of *at least*

$$\frac{B}{\lceil \log \Delta \rceil} \cdot \sum_{j=1}^{i-1} 2^{j-1} \cdot l_{min} + \left(\frac{B}{\lceil \log \Delta \rceil} - r_{max} \right) \cdot 2^{i-1} \cdot l_{min}$$

$$= \frac{c}{\lceil \log \Delta \rceil} \cdot (2^i - 1) \cdot l_{min} - r_{max} \cdot (2^{i-1}) \cdot l_{min}.$$

Furthermore, by the operation of the algorithm, this benefit will be distinct for each "batch" of rejections within different sub-intervals. Thus, the *maximum loss-to-benefit ratio* within each such sub-interval is

$$\frac{B \cdot 2^i \cdot l_{min}}{\frac{B}{\lceil \log \Delta \rceil} \cdot (2^i - 1) \cdot l_{min} - r_{max} \cdot (2^{i-1}) \cdot l_{min}} = \lceil \log \Delta \rceil \cdot \frac{2^i}{(2^i - 1) - \rho \cdot 2^{i-1} \cdot \lceil \log \Delta \rceil}$$

$$\leqslant \lceil \log \Delta \rceil \cdot \frac{2}{1 - \rho \cdot \lceil \log \Delta \rceil}, \qquad \textit{independent of } i.$$

Thus, if we let L_j denote the loss that the $\mathscr{DBP}$ scheme incurs within the interval I_j, then

$$V(A_j) + V(A_{j-1}) \geqslant \max \left\{ \frac{B}{\lceil \log \Delta \rceil} \cdot (2^{s_j} - 1) \cdot l_{min} - r_{max} \cdot (2^{s_j - 1}) \cdot l_{min}, \frac{L_j \cdot (1 - \rho \cdot \lceil \log \Delta \rceil)}{2 \cdot \lceil \log \Delta \rceil} \right\},$$

where the first term in the max{ } follows from the definition of the saturation level of I_j. Also

$$V(R_j) \leqslant L_j + B \cdot 2^{s_j} \cdot l_{min},$$

where the second term captures the maximum possible loss due to rejections at the end of I_j. Combining the last two inequalities, we have

$$V(A_j) + V(A_{j-1}) \geqslant \max \left\{ \frac{B \cdot (1 - \rho \cdot \lceil \log \Delta \rceil)}{2 \cdot \lceil \log \Delta \rceil} \cdot 2^{s_j} \cdot l_{min}, \frac{L_j \cdot (1 - \rho \cdot \lceil \log \Delta \rceil)}{2 \cdot \lceil \log \Delta \rceil} \right\}$$

$$\geqslant \frac{1 - \rho \cdot \lceil \log \Delta \rceil}{2 \cdot \lceil \log \Delta \rceil} \cdot \frac{B \cdot 2^{s_j} \cdot l_{min} + L_j}{2} \geqslant \frac{V(R_j) \cdot (1 - \rho \cdot \lceil \log \Delta \rceil)}{4 \cdot \lceil \log \Delta \rceil}.$$

And, summing over all intervals I_j, we have $V_{\mathscr{DBP}}(\bar{\sigma}) \geqslant (1 - \rho \cdot \lceil \log \Delta \rceil) \cdot \sum_j V(R_j) / 8 \cdot \lceil \log \Delta \rceil$.

Observe that for the optimal clairvoyant scheduler OPT, $V_{OPT}(\bar{\sigma}) < V_{\mathscr{DBP}}(\bar{\sigma}) + \sum_j V(R_j)$, or, using the above inequality, $V_{OPT}(\bar{\sigma}) \leqslant V_{\mathscr{DBP}} \cdot (1 + 8 \cdot \lceil \log \Delta \rceil / (1 - \rho \times \lceil \log \Delta \rceil))$. This completes the proof for the variable bandwidth case. ∎

Corollary 4.2 follows directly from Theorems 4.5 and 4.6. Combined with the lower bounds in Section 4.2, Corollary 4.2 establishes the *near-optimality* of the $\mathscr{SBP}$ and $\mathscr{DBP}$ policies for the variable bandwidth case, assuming that the server bandwidth B is larger than $2 \cdot r_{max} \cdot \lceil \log \Delta \rceil$. Again, this is a requirement that is typically satisfied by today's CM servers and applications. (See the discussion before Theorem 4.5.) Note that even smaller competitive ratios can be obtained if $B > k \cdot r_{max} \cdot \lceil \log \Delta \rceil$, where $k > 2$.

COROLLARY 4.2 (Near-Optimality of $\mathcal{SBP}$ and $\mathcal{DBP}$, Variable Bandwidth Case). *Assume* $\rho = r_{max}/B < 1/2 \cdot \lceil \log \Delta \rceil$. *Then:*

(1) *The* $\mathcal{SBP}$ *admission control policy is* $6 \cdot \lceil \log \Delta \rceil$*-competitive for the variable bandwidth case; and,*

(2) *The* $\mathcal{DBP}$ *admission control policy is* $(1 + 16 \cdot \lceil \log \Delta \rceil)$*-competitive for the variable bandwidth case.*

Although the constants in the competitiveness bounds we have shown for $\mathcal{DBP}$ are larger than those of $\mathcal{SBP}$, we conjecture that they can be improved. To support our conjecture, note that for the identical bandwidth case when a request of length $l_j \in [2^{i-1} \cdot l_{min}, 2^i \cdot l_{min})$ is rejected in $\mathcal{SBP}$, the scheduler can guarantee that the benefit of running requests is at least $\frac{c}{\lceil \log \Delta \rceil} \cdot 2^{i-1} \cdot l_{min}$, whereas with $\mathcal{DBP}$ the corresponding guaranteed benefit is at least

$$\frac{c}{\lceil \log \Delta \rceil} \cdot \sum_{k=1}^{i} 2^{k-1} \cdot l_{min} = \frac{c}{\lceil \log \Delta \rceil} \cdot (2^i - 1) \cdot l_{min},$$

that is, nearly double the benefit for $\mathcal{SBP}$. Of course, the main advantage of $\mathcal{DBP}$ over $\mathcal{SBP}$ is that, by "down-shifting," it can significantly reduce the effects of bandwidth fragmentation in the variable bandwidth case. A formal proof of improved competitive ratios for $\mathcal{DBP}$ is left as an open problem for future research.

Even though $\mathcal{SBP}$ and $\mathcal{DBP}$ guarantee logarithmic competitiveness under a worst-case scenario, they may also severely underutilize the server in average cases. For example, when all the requests address the shortest group of clips residing on the server, both schemes will end up utilizing only $\frac{1}{\lceil \log \Delta \rceil}$ of the available bandwidth. This is clearly undesirable. We now propose a novel on-line admission control policy that employs the intuition of prepartitioning schemes (to avoid worst-case scenarios for $\mathcal{WC}$) within a framework that also allows for good average-case performance. Roughly speaking, the idea is to use the methodology of $\mathcal{DBP}$ but define the sizes of the bandwidth partitions B_i as a function of the popularities and/or the lengths of all requests in the length range $[2^{i-1} \cdot l_{min}, 2^i \cdot l_{min})$. The resulting admission control scheme, termed *Popularity-based Bandwidth Prepartitioning* ($\mathcal{PBP}$), is depicted in Fig. 4. Note that $\mathcal{PBP}$ is given in parameterized form with the parameter f being the specific function of popularities and lengths used to define the partition sizes. In Section 5, we describe two specific choices for f used in our preliminary experimental study. The $\mathcal{PBP}$ admission control scheme relies on the assumption that request (i.e., clip) popularities can be estimated with reasonable accuracy (e.g., using a "moving window" prediction method [21]). Clearly, taking popularities into account is necessary to avoid worst-case scenarios for $\mathcal{DBP}$ (i.e., when the most frequent requests are also the shortest). In fact, assuming that requests are independent and the given popularities are accurate, we can construct simple arguments based on *Chernoff bounds* [23] to show that the probability of a worst-case "loss" for $\mathcal{PBP}$ (with specific choices for f) is exponentially small.

Popularity-based Bandwidth Prepartitioning[f: *function*]

1. Let p_j denote the probability that the length of an incoming request is l_j (i.e., the popularity of l_j). Let PL_i denote the set of (popularity, length) pairs with lengths in the i^{th} range; that is,

$$PL_i = \{ (p_j, l_j) \mid l_j \in [2^{i-1} \cdot l_{min}, 2^i \cdot l_{min}) \}.$$

2. Divide the available bandwidth B into $\lceil \log \Delta \rceil$ partitions $B_1, \ldots, B_{\lceil \log \Delta \rceil}$, with

$$|B_i| = \frac{f(PL_i)}{\sum_i f(PL_i)} \cdot B.$$

3. For each arriving request $\sigma_j = (t_j, l_j, r_j)$

3.1 Let $i \in \{1, \ldots, \lceil \log \Delta \rceil\}$ be such that: $2^{i-1} \cdot l_{min} \leq l_j < 2^i \cdot l_{min}$ (allowing for $l_j = 2^i \cdot l_{min}$ if $i = \lceil \log \Delta \rceil$).

3.2 If the total amount of free bandwidth in $B_1 \cup \ldots \cup B_i$ is less than r_j, then reject σ_j;
Otherwise, schedule σ_j using available bandwidth from partitions $B_i, B_{i-1}, \ldots, B_1$ *in that order.*

FIG. 4. Algorithm $\mathscr{PBP}$.

5. EXPERIMENTAL STUDY

In this section, we describe the results of a preliminary set of experiments we have conducted with the $\mathscr{WC}$ and $\mathscr{PBP}$ strategies for on-line admission control. Since our competitiveness results clearly demonstrate the superiority of prepartitioning schemes with respect to worst-case scenarios, our goal was to ensure that the worst-case guarantees did not impair *average-case performance.* We start by presenting our experimental testbed and methodology.

5.1. Experimental Testbed

To, examine the average-case behavior of the $\mathscr{WC}$ and $\mathscr{PBP}$ schemes, we have experimented with three distinct random arrival patterns:

- *Poisson Arrivals.* Requests of different lengths arrive at the server according to a Poisson process model with an arrival rate of λ. This is a plausible probabilistic model for servers with a reasonably steady traffic flow (e.g., video servers in scientific research labs serving clips of recorded experiments to scientists around the globe).

- *Bursty Arrivals.* Requests of different lengths arrive at the server in bursts at regular intervals of time (termed *burst separation*). Each such burst itself consists of a sequence of *request batches,* where each batch consists of requests of identical length arriving during a very short period of time. The batch arrivals are again modeled as a Poisson process with an arrival rate of λ. This workload is intended to model "rush-hour traffic" situations in CM servers.

• *Poisson+Short Burst Arrivals*. Long requests arrive at the server according to a Poisson process model with an arrival rate of λ_{long} the same time, bursts of short requests arrive based on a Poisson process with an arrival rate of λ_{short}. This workload model combines some features of the previous two models. It is intended to represent situations where servers operating under a relatively steady flow of long requests (e.g., movies or sports events), occasionally have to handle bursts of short requests (e.g., the 6 o'clock news).

In most of our experiments, the request lengths were sampled from a discrete set of values between 5 and 150 minutes, with sampling probabilities (i.e., popularities) taken from a Zipfian distribution with skew parameter z [35]. We varied this skew parameter from 0.0 (uniform) to 2.0 (very skewed). Results were obtained for three different models of correlation between request lengths and popularities:

• *Positive*. Larger popularities are assigned to longer requests.

• *Negative*. Larger popularities are assigned to shorter requests.

• *Random*. No length/popularity correlation exists; that is, the values of the Zipfian probability vector are assigned to the different request lengths in a random manner.

We also experimented with two different choices for the f function parameter of the $\mathcal{PBP}$ scheme. The first choice f_1 captured the cumulative popularity of a length range, that is, $f_l(PL_i) = \sum_{(p,l) \in PL_i} p$. The second choice f_2 was the total popularity-length product of a range, that is $f_2(PL_i) = \sum_{(p,l) \in PL_i} p \cdot l$.

In our experiments for the identical bandwidth case, we assumed a server with 100 available channels. For the variable bandwidth case, we varied the server's sustained bandwidth capacity between 100 and 250 Megabits per second (Mbps) and selected the rate requirement of a request randomly between 500 Kbps and 8 Mbps. The parameter values are summarized in Table 1.

For each different combination of input parameters, we modeled the system behavior under each scheduling policy for 20,000 minutes of simulated time and 10 randomly generated request sequences. The results presented here represent the averages over these 10 runs of the system. In all cases, the comparison metric was the *fraction of the server capacity effectively utilized*; that is, the ratio $V_A(\bar{\sigma})/$(server bandwidth) $\times$ (simulation time), for each scheduler A and request sequence $\bar{\sigma}$.

TABLE 1

Experiment Parameter Settings

System, parameter	Value
Server, bandwidth capacity	100 channels/100–250 Mbps
Request, lengths	5–150 minutes
Request, rates (variable bandwidth case)	500 Kbps–8 Mbps
Zipfian popularity skew (z)	0.0–2.0

5.2. Experimental Results

We present an overview of our experimental comparison of the $\mathscr{WC}$ and $\mathscr{PBP}$ schemes for the (general) variable bandwidth case. Similar results were obtained for identical bandwidth requests. For the numbers presented here, the request lengths were sampled (based on z and the model of correlation) from the collection $\{5, 10, 15, 90, 120, 150\}$ (in minutes), the request rates were selected (uniformly) from the set $\{0.5, 1.5, 3.0, 4.5, 6.0, 8.0\}$ (in Mbps), and the server bandwidth capacity was 250 Mbps. The plots shown in this section are indicative of the results obtained for other values of request and server parameters.

We focus our discussion on $\mathscr{PBP}[f_2]$, that is, $\mathscr{PBP}$ using the "total popularity × length" partitioning criterion, since it exhibited uniformly better performance than $\mathscr{PBP}[f_1]$ in our experiments. We should stress, however, that; even with "cumulative popularity" partitioning, $\mathscr{PBP}$ outperformed $\mathscr{WC}$ by a significant margin for our "bursty" workloads.

The first set of experiments studied the relative effectiveness of the $\mathscr{WC}$ and $\mathscr{PBP}$ schemes under Poisson arrivals for different values of the Zipfian skew parameter z and different length/popularity correlations. Figure 5a shows the performance of the schemes as a function of the Poisson arrival rate λ for $z = 0.6$ and random length/popularity correlation. Our basic finding is that, by exploiting its knowledge of clip popularities, $\mathscr{PBP}$ is able to do at least as good as $\mathscr{WC}$ in all cases. We should mention that we also experimented with different models of the arrival process (e.g., using uniformly rather than exponentially distributed interarrival times) that also led to the same conclusions regarding the relative performance of the strategies under random arrivals.

The second set of experiments concentrated on the relative performance of the algorithms under the Bursty Arrival model described in the previous section. We studied the server utilization as a function of the length of the burst separation interval as well as the size of a batch of arrivals for different values of the z and λ parameters, the size (i.e., number of batches) of a burst, and different modes of correlation. Figure 5b shows the server utilization as a function of the burst separation interval for batch size equal to 40, $z = 0.6$, batch arrival rate $\lambda = 0.8$, burst size equal to 10, and random length/popularity correlation. (We show burst

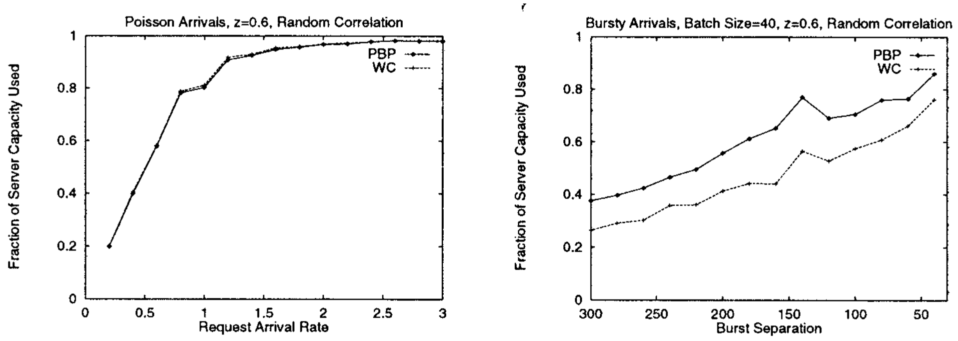

FIG. 5. (a) Server throughput under Poisson arrivals. (b) Server throughput under bursty arrivals (random correlation).

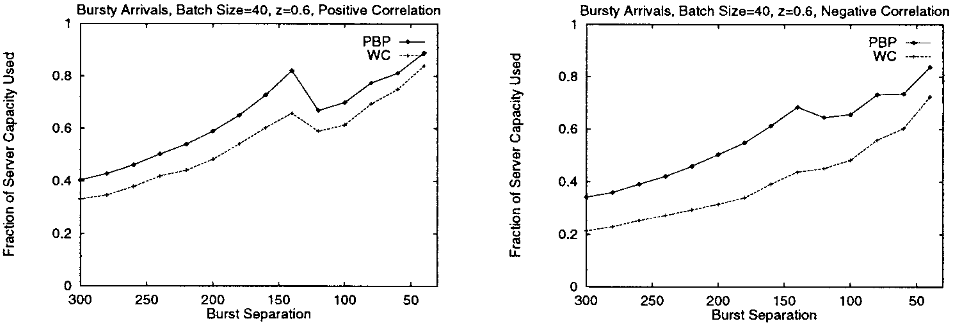

FIG. 6. (a) Server throughput under bursty arrivals (positive correlation). (b) Server throughput under bursty arrivals (negative correlation).

separations decreasing from left to right as this reflects increasing load, as in Fig. 5a.) Our results show that under such conditions, $\mathcal{PBP}$ outperforms $\mathcal{WC}$ by an average margin of 15%–40%. Note that the "jump" observed in the curves as the burst separation approaches 150 minutes is caused by our specific choice of request lengths and our model of "bursty" arrivals. The numbers from the same experiment but for *positive* length/popularity correlation (i.e., longer requests are more popular) are depicted in Fig. 6a. $\mathcal{WC}$ clearly performs better under the positive correlation assumption, since it is able to allocate more of its channels to the more popular (and, more profitable) long requests. Still, $\mathcal{PBP}$ continues to outperform $\mathcal{WC}$ by up to 25%. Figure 6b shows the results of the same experiment but for *negative* length/popularity correlation (i.e., shorter requests are more popular). Under such scenarios, our, results show that the relative improvement offered by $\mathcal{PBP}$ over $\mathcal{WC}$ can reach 50%–60%. A different perspective is depicted in Fig. 7a, where server utilization (for the same parameter values and negative correlation) is given as a function of the batch, size for a fixed burst separation of 180 minutes. Note that as the batch size increases, the Bursty Arrival model gives rise to worst-case scenarios for $\mathcal{WC}$, where a large batch of short requests can flood the server leaving no capacity for a following batch of larger requests. On the other hand, our $\mathcal{PBP}$ scheme is capable of maintaining a reasonable level of utilization under all circumstances.

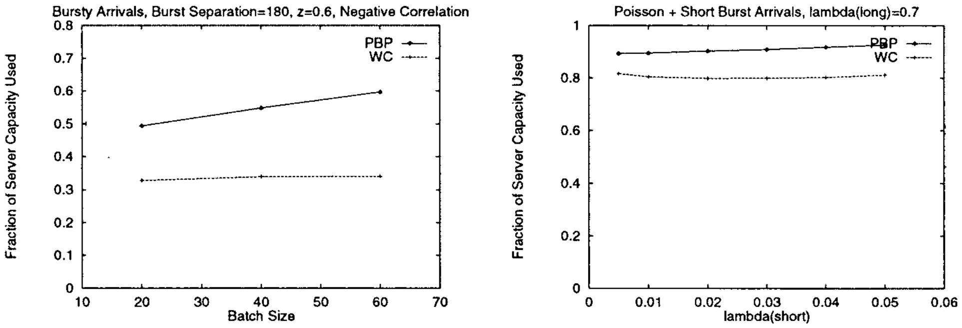

FIG. 7. (a) Server throughput under bursty arrivals as a function of batch size (negative correlation). (b) Server throughput under Poisson + Short Bursts.

The final set of experiments studied the behavior of the algorithms under the combined Poisson + Short Bursts arrival process. We concentrated on a particular scenario which, we believe, is common in Video-On-Demand environments. Specifically, we assumed that the server is working close to capacity serving requests for long (i.e., $\{90, 120, 150\}$ minutes) movies but occasionally has to handle bursts of short (i.e., $\{5, 10, 15\}$ minutes) requests. That is, λ_{long} was selected large enough to ensure high system utilization And' we studied the server utilization as a function of λ_{short}. All length popularities were assumed uniform for this experiment. The results depicted in Fig. 7b show that, under this scenario, $\mathscr{PBP}$ can offer a 10%–15% performance improvement over $\mathscr{WC}$, even at high levels of system utilization.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have addressed the Admission control problem associated with CM database servers from a novel, on-line perspective. Using server throughput as our optimization metric, we showed that the traditionally used Work-Conserving policy has a competitive ratio of $\frac{1+\Delta}{1-\rho}$, where Δ is the ratio of the maximum to minimum request length and ρ is the maximum fraction of the server's bandwidth that a request can demand. We developed novel admission control strategies based on the simple idea of preprepartitioning the bandwidth capacity of the server among requests of different length and proved that our strategies are $O(\log \Delta)$-competitive for sufficiently large server bandwidth. We also showed an $\Omega(\frac{\log \Delta}{1-\rho})$ lower bound on the competitive ratio of any deterministic *or* randomized algorithm for the problem, thus establishing that our bandwidth prepartitioning algorithms are within a multiplicative constant of the optimal on-line strategy. Based on the intuition gained from our competitiveness results, we proposed prepartitioning schemes that make use of request popularities to ensure good average-case as well as robust worst-case performance, and experimentally verified their effectiveness against the Work-Conserving policy.

We believe that the analytical and experimental results presented in this paper offer new insights to other optimization problems that arise in CM data management. For example, consider the problem of data placement and static load balancing in distributed CM servers. Briefly, the problem can be described as follows: Given a collection of continuous media clips with lengths (l_i), rates (r_j), and expected popularity (or, probability of access, p_i), determine a "good quality" mapping of these clips to a collection of servers, where each server is characterized by a bandwidth capacity (B_j) and a storage capacity (S_j) and a clip can be mapped to more than one server (i.e., replication of clips is allowed).

Traditionally, the goal of data placement schemes in this setting is to balance the expected bandwidth load (according to the popularities $\{p_i\}$) across the available servers under the given server storage constraints [11, 21, 34]. This model of "popularity-based data placement" aims at achieving good system utilization and balanced system load in an average sense. On the other hand, our competitiveness results indicate that to ensure *robust* system performance, a placement strategy should also try to achieve some secondary "goals." One such goal, for example, would be to place clips with large bandwidth requirements on servers with large bandwidth capacities to guarantee small ρ fractions for each server. As another example, the placement policy should try to replicate short clips across many servers, so that there are many possibilities of dynamically re-assigning (e.g., using a baton passing primitive [34]) streams delivering these clips to different servers. This obviously reduces the probability of a short request causing the rejection of a long request from the system, which is the worst-case scenario in all our competitiveness results. Achieving such secondary data placement goals is especially important in order to ensure good system utilization under short-term fluctuations of the load away from the averages $\{p_i\}$ or overload situations where some client requests simply must be rejected. A detailed investigation of the problem is left for future research.

The competitive analysis framework and results presented in this paper suggest several directions for future work. First, this work has only considered the bandwidth resource. Considering servers with multiple scarce resources poses an interesting challenge. For example, a request may also need a given amount of memory at the server in order to meet its performance requirements. This memory requirement can be either specified by the request itself (e.g., leaky bucket regulated traffic) or assigned by the server to meet the request's performance goals. Given the limited amount of server memory, the admission control mechanism needs to consider both the memory and the bandwidth requirements of a request. Second, we plan to use our results and methodology as a starting step towards a formal study of dynamic on-line load balancing in multimedia storage servers. Prior work [11, 34] has explored the use of techniques such as dynamic stream re-assignments and dynamic replica creation/coalescing only within ad-hoc schemes, without providing any strong performance guarantees or exploiting the wealth of theoretical results on on-line load balancing (see, for example, [2, 5, 6, 32]). Finally, incorporating the concept of *equivalent bandwidth* into our on-line analysis is another interesting problem. Equivalent bandwidth is fundamental for providing sessions with statistical performance guarantees. Since the equivalent bandwidth of a collection of sessions is typically a simple function of each session's equivalent bandwidth, this concept simplifies the admission control for applications with statistical performance guarantees [12, 18].

REFERENCES

1. S. Aggarwal, J. Garay, and A. Herzberg, Adaptive video on demand, *in* "Proceedings of the 3rd Annual European Symposium on Algorithms (ESA '95), Corfu, Greece, September 1995."

2. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, On-line load balancing with applications to machine scheduling and virtual circuit routing, *in* "Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, San Diego, California, May 1993," pp. 623–631.

3. B. Awerbuch, Y. Azar, and S. Plotkin, Throughput-competitive on-line routing, *in* "Proceeding of the 34th Annual IEEE Symposium on Foundations of Compute Science, Palo Alto, CA, November 1993," pp. 32–40.

4. B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén, Competitive non-preemptive call control, *in* "Proceedings of the Fifth Annual ACM-SIAM symposium on Discrete Algorithms, Arlington, Virginia, January 1994," pp. 312–320.

5. Y. Azar, On-line load balancing, unpublished manuscript, available from http://www.math.tau.ac.il/~ azar/, 1997.

6. Y. Azar, Z. Broder, and A. R. Karlin, On-line load balancing, *Theoret. Comput. Sci.* **130** (1994), 73–84.

7. A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, and B. Schieber, Bandwidth allocation with preemption, *in* "Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, Las Vegas, Nevada, May 1995," pp. 616–625.

8. A. Bar-Noy, J. A. Garay, A. Herzberg, and S. Aggarwal, Sharing video on demand, unpublished manuscript, available from http://www.eng.tau.ac.il/~ amotz/publications.html, 1996.

9. S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, and F. Wang, On the competitiveness of on-line real-time task scheduling, *in* "Proceedings of the 12th IEEE Real-Time Systems Symposium, San Antonio, Texas, December 1991," pp. 106–115.

10. Mon-Song Chen, D. D. Kandlur, and P. S. Yu, Optimization of the grouped sweeping scheduling (GSS) with heterogeneous multimedia streams, *in* "Proceedings of ACM Multimedia '93, Anaheim, CA, August 1993," pp. 235–242.

11. A. Dan and D. Sitaram, An online video placement policy based on bandwidth to space ration (BSR), *in* "Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, CA, May 1995."

12. A. Elwalid, D. Mitra, and R. H. Wentworth, A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node, *IEEE J. Select. Areas Comm.* **13** (August 1995), 1115–1127.

13. U. Faigle, R. Garbe, and W. Kern, Randomized online algorithms for maximizing busy time interval scheduling, *Computing* **56** (1996), 95–104.

14. U. Faigle and W. M. Nawijn, Note on scheduling intervals on-line, *Discrete Appl. Math.* **58** (1995), 13–17.

15. J. A. Garay, I. S. Gopal, S. Kutten, Y. Mansour, and M. Yung, Efficient on-line call control algorithms, *J. Algorithms* **23** (1997), 180–194.

16. M. N. Garofalakis, B. Özden, and A. Silberschatz, Resource scheduling en enhanced pay-per-view continuous media databases, *in* "Proceedings of the 23rd International Conference on Very Large Data Bases, Athens, Greece, August 1997," pp. 516–525.

17. R. Gawlick, "Admission Control and Routing: Theory and Practice," Ph.D. thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, June 1995 (Technical Report MIT/LCS/TR-679).

18. R. Guerin, H. Ahmadi, and M. Nagshineh, Equivalent capacity and its application to bandwidth in high speed networks, *IEEE J. Select. Areas Comm.* **9** (September 1991), 968–981.

19. G. Koren and D. Shasha, "An Optimal Scheduling Algorithm with a Competitive Factor for Real-Time Systems," Technical Report TR 572, Department of Computer Science, New York University, July 1991.

20. R. J. Lipton and A. Tomkins, Online interval scheduling, *in* "Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Arlington, Virginia, January 1994," pp. 302–311.

21. T. D. C. Little and D. Venkatesh, Popularity-based assignment of movies to storage devices in a video-on-demand system, *ACM Multimedia Systems* **2** (1995), 280–287.

22. D. D. E. Long and M. N. Thakur, Scheduling real-time disk transfers for continuous media applications, *in* "Proceedings of the Twelfth IEEE Symposium on Mass Storage Systems, 1993," pp. 227–232.

23. R. Motwani and P. Raghavan, "Randomized Algorithms," Cambridge Univ. Press, Cambridge, UK, 1995.

24. B. Özden, R. Rastogi, and A. Silberschatz, A framework for the storage and retrieval of continuous media data, *in* "Proceedings of the 1995 International Conference on Multimedia Computing and Systems, Washington, DC, May 1995," pp. 2–13.

25. B. Özden, R. Rastogi, and A. Silberschatz, Disk striping in video server environments, *IEEE Data Engr. Bull.* **18** (December 1995), 4–16.

26. B. Özden, R. Rastogi, and A. Silberschatz, The storage and retrieval of continuous media date, *in* "Multimedia Database Systems: Issues and Research Directions" (V. S. Subrahmanian and S. Jajodia, Eds.), pp. 237–261, Springer-Verlag, New York/Berlin, 1996.

27. S. Plotkin, Competitive routing of virtual circuits in ATM networks, *IEEE J. Select. Areas Comm.* (August 1995), 1128–1136.

28. P. Venkat Rangan and H. M. Vin, Designing file systems for digital video and audio, *in* "Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles, Monterey, CA, October 1991," pp. 81–94.

29. D. D. Sleator and R. E. Tarjan, Amortized efficiency of list update and paging rules, *Comm. Assoc. Comput. Mach.* **28** (February 1985), 202–208.

30. R. A. Staehli, "Quality of Service Specification for Resource Management in Multimedia Systems," Ph.D. thesis, Oregon Graduate Institute of Science & Technology, January 1996.

31. Y. Wang, J. C. L. Liu, D. H. C. Du, and J. Hsieh, Video file allocation over disk arrays for video-on-demand, *in* "Proceedings of the 1996 International Conference on Multimedia Computing and Systems, Hiroshima, Japan, June 1996," pp. 160–163.

32. J. Westbrook, Load balancing for response time, *in* "Proceedings of the 3rd Annual European Symposium on Algorithms (ESA '95), Corfu, Greece, September 1995," pp. 355–368.

33. G. J. Woeginger, On-line scheduling of jobs with fixed start and end times, *Theoret. Comput. Sci.* **130** (1994), 5–16.

34. L. Wolf, P. S. Yu, and H. Shachnai, DASD dancing: a disk load balancing optimization scheme for video-on-demand computer systems, *in* "Proceedings of the 1995 ACM SIGMETRICS Conference on Measurement & Modeling of Computer Systems, Ottawa, Canada, May 1995."

35. G. Kingsley Zipf, "Human Behavior and the Principle of Least Effort—An Introduction to Human Ecology," Addison–Wesley, Cambridge, MA, 1949.