for Multimedia Scheduling Schemes". IBM Research Report RC20038, April 1995.

[15] W. Sincoskie, "System Architecture for Large Scale Video on Demand", *Computer Networks ISDN System*, Vol. 22, 1991, pp. 155-162.

[16] Wolf J.L., Yu P.S., Shachnai H. "Disk Load Balancing for Video-on-Demand Systems",*ACM Multimedia Systems Journal*, to appear.

[17] W. Zohng, Y. Onozato and J. Kaniyil, "Copy Network with Shared Buffers for Large-Scale Multicast ATM Switching", *IEEE/ACM Transactions on Networking*, 1:2, April 1993, pp. 157-165.

[4] A. Dan, D. Sitaram and P. Shahabuddin, "Scheduling Policies for On-Demand Video Server with Batching", in Proceedings of the *Second ACM International Multimedia Conference (ACM MULTIMEDIA'94)*.

[5] H. Dykeman, M. Ammar and J. Wong, "Scheduling Algorithms for Videotex Systems under Broadcast Deliver", *in Proceedings of ICC'86*, pp. 1847–1851.

[6] *Electronic Engineering Times*, March 15, 1993.

[7] Fox E., "The Coming Revolution in Interactive Digital Video", *Communications of the ACM*, 7, July 1989, pp. 794-801.

[8] Garey, M.R. and Johnson, D.S. Computers and intractability: A Guide to the Theory of NP-Completeness. W.H.Freeman.

[9] Gelenbe E., "Random Neural Networks with Negative and Positive Signals and Product Form Solution", *Neural Computation*, 1(4):502-510, 1989.

[10] Gelenbe E. and Shachnai H., "A Unified Queuing Analysis for Multimedia Systems", *Manuscript*, 1996.

[11] J-Y Le Boudec, "The asynchronous Transfer Mode: A Tutorial", *Computer networks and ISDN Systems*, 24, 1992, pp. 279-309.

[12] Marchok D., Rohrs C. and M. Schafer, "Multicasting in Growable Packet (ATM) Switch", *IEEE INFOCOM*, 1991, pp. 850-858.

[13] Miller L., "Alternating Priorities in Multiclass queues", *Ph.D. Thesis*, Cornell University, Ithaca, New York, 1964.

[14] Rangan P., Vin H. and Ramanathan S., "Designing an On-Demand Multimedia Service," *IEEE Communication Magazine*, 30, July 1992, pp. 56-65.

By Theorem 3, the Rounded Ratio algorithm yields a 2-approximation to the minimal latency. Using the simulation results, as given in Figure 5, we note that the MQL provides a ratio that is a small constant to the optimum (that is, at most 3) also for the Zipf distribution, under which

$$p_{min} \approx \frac{1}{M \lg M} \ . \tag{16}$$

Theorem 2 gives the upper bound of $\lg M$ on this ratio for any distribution satisfying (16).

In Figure 6 we used a fixed arrival rate of 50 per minute, and a server capacity in the range of 500 to 2000 streams. We show the decrease in system's latency achieved by the studied schemes as the server capacity increases. We note that for a server capacity of 1000 streams or larger, the differences between the latencies under the FCFS, RR and the Rounded Ratio algorithm are very small, however, when the server capacity is very limited, the Rounded Ratio does substantially better than the other schemes.

Acknowledgment

We would like to thank Johan Håstad for his helpful comments on an earlier version of this paper.

References

[1] Aggarwal C.C., Wolf J.L., Yu P.S., *On Optimal Batching Policies for Video-On-Demand Storage Servers*, Proceedings of the Intl. Conference on multimedia Computing and Systems, Hiroshima, 1996.

[2] Aggarwal S., Garay J., Herzberg A., *Adaptive Video on Demand*, Proceedings of the 3th European Symposium on Algorithms (ESA), pp. 538-553, 1995.

[3] Anderson D., "Metascheduling for Continuous Media", *ACM Transactions on Computer Systems*, 11:3, Aug 1993, pp. 226-252.

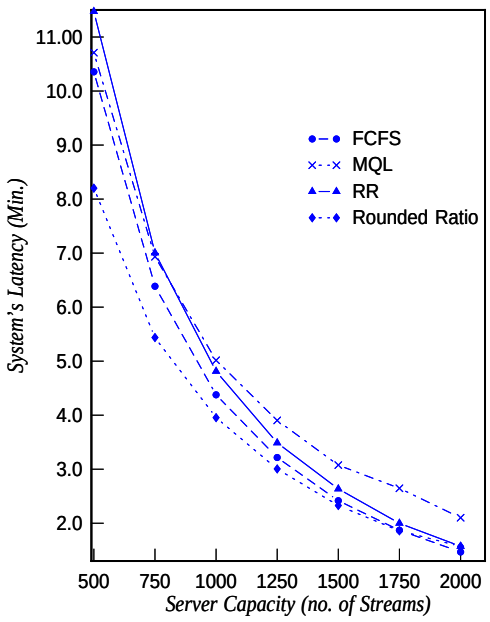

Figure 6: System's Latency vs. Server Capacity: Comparison of the Rounded Ratio with FCFS, MQL and RR

In Figure 4 we compare the system's latency under FCFS as computed in Section 3 and in the simulated system. We note, that the estimated latency is closer to the latency in the simulated system for higher arrival rates, for which it is indeed unlikely to find an idle stream (i.e., the rate at which I/O streams become available approaches d), as we assume in our analysis.

In Figures 5 and 6 we compare the system's latency under the FCFS, RR, MQL and the Rounded Ratio schemes. In Figure 5 we used a fixed server capacity and increasing loads, taking arrival rates in the range 10 to 100 per minute. We note, that the Rounded Ratio algorithm improves the latency obtained by the other schemes; This improvement becomes significant when the load is heavier. Thus, the Rounded Ratio algorithm is the best choice when the rpv $\bar{p}$ is known and fixed[3]. Among the three other schemes the FCFS provides the smallest latency. The ratio between the latencies under the RR and the FCFS did not exceed 1.2 in the simulated system.

[3]In practice, implementing the Rounded Ratio algorithm requires updating periodically $\bar{p}$ and the sizes of the batch intervals, due to changes in the relative frequencies of requests for the various programs.

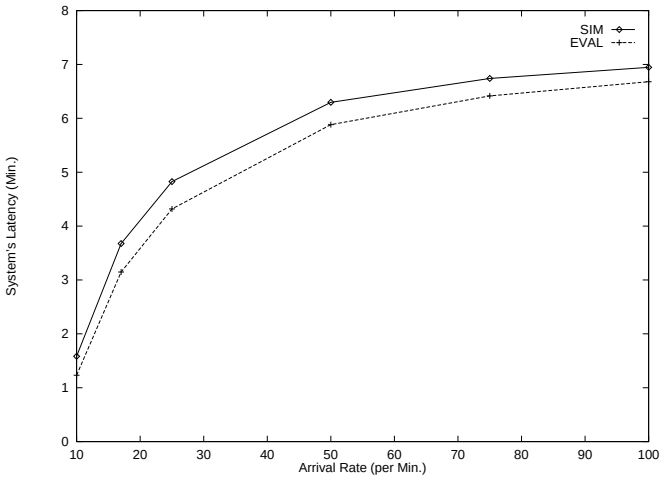

Figure 4: System's Latency vs. Arrival Rate

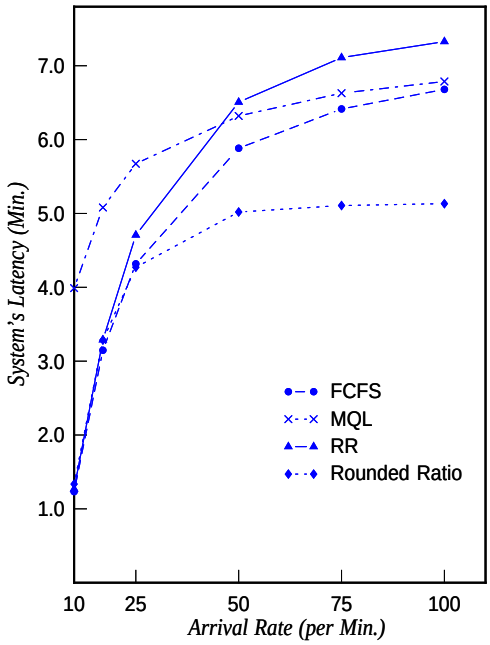

Figure 5: System's Latency: Comparison of the Rounded Ratio with FCFS, MQL and RR

Algorithm Rounded Ratio
for $i = 1$ to M
$l_i = \lceil \sqrt{\frac{p_i}{p_M}} \rceil$.
$b'_M = d \sum_{i=1}^{M} l_i$
for $j \geq 1$
Let I_j be the interval $[d(j(b'_M + 1) + 1), d(j + 1)(b'_M + 1)]$.
Schedule queue M at time $d(j + 1)(b'_M + 1)$.
for $i = M - 1, \ldots, 1$
Schedule queue i l_i times uniformly
along the interval I_j.

Figure 3: The Rounded Ratio Algorithm

and since queue i is scheduled l_i times between two successive schedules of queue M, for any $1 \leq i \leq M - 1$

$$b'_i = \frac{b'_M}{\lceil \sqrt{\frac{p_i}{p_M}} \rceil} \leq \frac{b'_M}{\sqrt{\frac{p_i}{p_M}}} \leq \frac{2b_M}{\sqrt{\frac{p_i}{p_M}}} = 2b_i \tag{15}$$

Thus, $E[\mathcal{W}(\mathcal{B}')] = \sum_{i=1}^{M} p_i b'_i \leq 2E[\mathcal{W}(\mathcal{B})]$. ■

5 Numerical Results

We give below the numerical results obtained in a simulation study of the proposed schemes. Our results validate the mathematical model used in Section 3 and compare the performance of the four batching schemes studied in this paper. We simulated a Poisson arrival process of customer requests to a multimedia database of $M = 100$ objects. The $rpv\bar{p}$ formed a Zipf's distribution, i.e., $p_k = \frac{1}{kH_M}$, where H_M, the Mth Harmonic number, is the normalization constant. The Zipf's distribution often reflects the relative frequencies of accesses to the objects, e.g., in a movie vending environment [16].

Corollary 1: *For any $c > 1$, if $p_{min} = \min_{1 \le i \le M} p_i \ge \frac{1}{cM}$, then*

$$\frac{E[\mathcal{W}(MQL)]}{E[\mathcal{W}(OPT)]} \le c \ . \tag{13}$$

4.2 The Rounded Ratio Algorithm

The Ratio Rule was shown in [1] to minimize the system's latency. However, as noted in [1], for a general rpv $\bar{p}$ this rule is *unrealizable*. Consider, e.g., the case where for some $1 \le i, j \le M$ $\sqrt{\frac{p_i}{p_j}}$ is irrational, there is no schedule that guarantees that (9) is satisfied. This holds for any server capacity.

In the following we present an approximation algorithm that we call the *Rounded Ratio*. In determining the lengths of the batch interval b'_i for queue i, $1 \le i \le M$, the algorithm uses the rounded ratio $\lceil \sqrt{\frac{p_i}{p_M}} \rceil$. The batch intervals are then defined recursively, starting from b'_M. We show that the Rounded Ratio algorithm yields a 2–approximation to the minimal latency. The algorithm is given in Figure 3.

Theorem 3: *The Rounded Ratio Algorithm achieves a ratio of 2 to the minimal latency.*

Proof: It is easy to verify that the algorithm is feasible, since l_i is an integer (Recall, that $p_i \ge p_M$ $\forall\, 1 \le i \le M-1$). Let b_i, b'_i be the batching intervals assigned to queue i by the Ratio Rule (denoted by $\mathcal{B}$) and by the Rounded Ratio algorithm (denoted by $\mathcal{B}'$) respectively. Using (11) we have

$$b'_M = d \sum_{j=1}^{M} l_i \le \frac{2d}{\sqrt{p_M}} \cdot \sum_{j=1}^{M} \sqrt{p_j} = 2b_M \ , \tag{14}$$

j and i respectively. Thus, the overall expected delay between two services of queue i is given by $d\sum_{j\neq i}\frac{p_j}{p_i} = d(\frac{1}{p_i}-1)$. As customers arrive to queue i by Poisson arrival process, for any interval I between two services of queue i, the arrival time of a customer is distributed uniformly on I, and the claim of the lemma follows. ∎

In the following we use the notation OPT for any batching scheme that minimizes the system's latency. We denote this latency by $E[\mathcal{W}(OPT)]$.

Theorem 2: *For a given rpv $\bar{p}$ the system's latency under MQL satisfies*

$$E[\mathcal{W}(MQL)] \leq \frac{M-1}{(\sum_{i=1}^{M}\sqrt{p_i})^2}\cdot E[\mathcal{W}(OPT)] \ . \tag{10}$$

Proof: Let b_i be the batching interval assigned by the Ratio Rule to queue i, $1 \leq i \leq M$. Given a delay of d time units between successive service completions, it is shown in [1] that

$$b_i = \frac{d\sum_{j=1}^{M}\sqrt{p_j}}{\sqrt{p_i}} \ . \tag{11}$$

Thus, using Lemmas 1 and 2 we have

$$\frac{E[\mathcal{W}(MQL)]}{E[\mathcal{W}(OPT)]} \leq \frac{\sum_{i=1}^{M}p_iE[\mathcal{W}_i(MQL)]}{\sum_{i=1}^{M}p_ib_i} \leq \frac{M-1}{\sum_{i=1}^{M}p_i\cdot\frac{\sum_{j=1}^{M}\sqrt{p_j}}{\sqrt{p_i}}} \tag{12}$$

and the statement of the theorem follows. ∎

Indeed, the MQL is most efficient when the distribution on accesses to the programs is moderately skewed, as stated in our next result.

Denote by E_j the event "At most one arrival to queue j in the interval I_j". From (i) and (ii) we have

$$\begin{aligned} Prob(A_j \mid B_j) &= Prob(A_j \mid E_j) \cdot Prob(E_j) + Prob(A_j \mid \bar{E}_j) \cdot Prob(\bar{E}_j) \\ &= Prob(X \leq Y) \cdot Prob(E_j) + Prob(X_1 \leq Y) \cdot Prob(\bar{E}_j) \end{aligned}$$

and the inequalities in (7) follow.

Summing up on $j \neq i$ we have the statement of the theorem. ■

4 Performance Bounds for the MQL and the Rounded-Ratio Algorithm

4.1 Analysis of the MQL

In this section we study the performance of the MQL scheme with respect to system's latency. We formulate in Theorem 2 a performance bound that depends on the distribution on requests for the programs, given by the *rpv* $\bar{p}$. Our result implies, that the MQL achieves performance ratio that is a small constant for moderately skewed distributions, as given in Corollary 1.

We first define the optimal scheme for the MOD scenario, as introduced in [1]: Let b_i denote a *fixed* length of time between successive services of queue i, $1 \leq i \leq M$. This is the *batching interval* of queue i. The *Ratio Rule* assigns to queues i, j the batching intervals b_i, b_j satisfying

$$\frac{b_i}{b_j} = \sqrt{\frac{p_j}{p_i}} \ , \text{ for all } 1 \leq i, j \leq M \ . \tag{9}$$

Lemma 1: *[1] The Ratio Rule minimizes the system's latency.*

Lemma 2: *If a stream is released every d time units then the latency of queue i under MQL is* $\frac{d}{2}(\frac{1}{p_i} - 1)$, $\forall \ 1 \leq i \leq M$.

Proof: We note, that for any $1 \leq i \leq M$, the expected number of services of queue $j \neq i$ between two successive services of queue i is $\frac{p_j}{p_i}$, which is the ratio between the rates of arrivals to queues

Theorem 1: *For any MOD system of M queues with requests generated by the rpv $\bar{p}$:*

$$\frac{E[\mathcal{W}(RR)]}{2} \leq E[\mathcal{W}(FCFS)] \leq E[\mathcal{W}(RR)] \ .$$

Proof: It is sufficient to show, that for any $1 \leq j \leq M$

$$\frac{1 - p^s_{0j}}{2} \leq 1 - p_{0j} \leq 1 - p^s_{0j} \ . \tag{7}$$

Let $I_j = [0..T]$ denote a time interval between two successive visits of the server in queue j. We note, that p_{0j} is the probability that a random sample of queue j in I_j finds it non-empty. Since the location of such a sample is distributed uniformly on $[0..T]$, while (by definition) the server always arrives at the end of the interval, we have the right inequality in (7). For showing the left inequality we denote by A_j the event "a random sample finds queue j non-empty", and by B_j the event "the server finds queue j non-empty", then

$$\frac{1 - p_{0j}}{1 - p^s_{0j}} = Prob(A_j \mid B_j) \ . \tag{8}$$

We handle below two cases separately:

(i) There is a single arrival to queue j in the interval I_j. Let X, Y denote the time of the single arrival, and the time queue j is randomly sampled respectively. Clearly, $Y \sim U(0,T)$, and by the memoryless property of the exponential distribution, we also have $X \sim U(0,T)$. It follows, that the probability that a random sample finds queue j non-empty is

$$Prob(X \leq Y) = 1/2 \ .$$

(ii) At least two customers arrive in I_j, then denote by X_1 the arrival time of the first customer and Y as in (i). Denote by C the event "A random sample of queue j occurs before the second arrival". Clearly, $Prob(X_1 \leq Y \mid \bar{C}) = 1$ and from (i), $Prob(X_1 \leq Y \mid C) = 1/2$. Hence,

$$Prob(X_1 \leq Y) \geq 1/2 \ .$$

or

$$p_{0i} = \frac{-(k_i - 1 + M - A) + \sqrt{(k_i - 1 + M - A)^2 + 4k_i}}{2} \quad 1 \leq i \leq M \ . \tag{4}$$

Summing up the equations for the p_{0i}'s we have

$$2A + \sum_{i=1}^{M} k_i + M(M-1) - MA = \sum_{i=1}^{M} \frac{\sqrt{(k_i - 1 + M - A)^2 + 4k_i}}{2}. \tag{5}$$

Since $A \in [1, M]$, a close approximation for its exact value may be found numerically. Substituting into (3) we obtain an expression for the p_{0i}'s, that gives the expected length of the vacation for queue i, $E(U_i^{\mathcal{F}})$, $1 \leq i \leq M$, as given in (1).

We note, that the Poisson arrival process of customers implies, that the arrival times of customers to the i-th queue are distributed uniformly in the interval $[0, U_i^{\mathcal{F}}]$, thus, the expected latency of queue i is

$$E[\mathcal{W}_i(FCFS)] = E(U_i^{\mathcal{F}})/2 \ , \tag{6}$$

which gives $E[\mathcal{W}(FCFS)]$.

3.2 Performance of the RR

Using the server of the walking type model for the RR scheme, we note, that the length of the vacation $U_i^{\mathcal{R}}$ taken by the server of queue i is the number of non-empty queues found by the server while circulating in the system after serving queue i. Denote by p_{0j}^s the steady state probability that upon arrival to queue j the server finds the queue empty, then

$$E(U_i^{\mathcal{R}}) = \sum_{j \neq i} (1 - p_{0j}^s) \ ,$$

In the following we establish the relation between the system's latencies under FCFS and the RR.

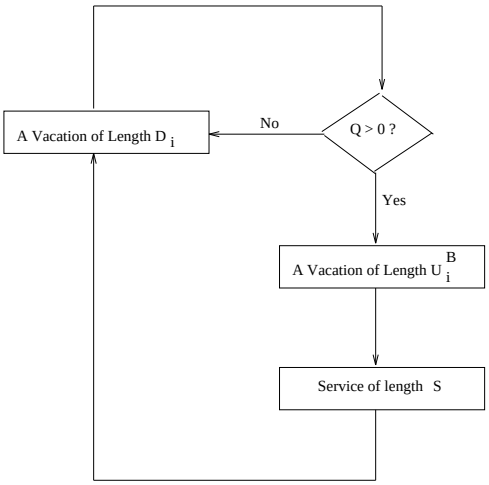

Figure 2: Server of the Walking type Model for Batching Systems

following we derive an expression for $E(U_i^{\mathcal{F}})$ that allows to compute the system's latency. Denote by p_{0j} the steady state probability that the jth queue is empty, $1 \leq j \leq M$, then

$$E(U_i^{\mathcal{F}}) = d \cdot \sum_{j \neq i}(1 - p_{0j}) = d \cdot (M - 1 - \sum_{j \neq i} p_{0j}) \ . \tag{1}$$

Observe, that the fraction of time the i-th queue is empty is the fraction of time in which the server waits for the arrival of the first customer. Let D_i' denote the random variable that gives the length of this time, then since the server repeatedly takes a vacation of length d, and the expected time till the first arrival is $1/\lambda_i$, we can write $k_i = \lceil \frac{1}{\lambda_i d} \rceil$, and $E(D_i') = k_i d$.

Therefore,

$$p_{0i} = \frac{E(D_i')}{E(D_i') + E(U_i^{\mathcal{F}})} = \frac{k_i}{k_i + \sum_{j \neq i}(1 - p_{0j})} \ . \tag{2}$$

Let $A = \sum_{j=1}^{M} p_{0j}$, then

$$p_{0i} = \frac{k_i}{k_i + M - 1 - A + p_{0i}} \ , \tag{3}$$

2.3 Batching Schemes and Performance Measure

The following schemes are studied below:

- **First Come First Served (FCFS)** – Using the arrival times of customers the scheduler chooses to serve the queue with the longest waiting customer.
- **Maximal Queue Length (MQL)** – The program with maximal queue length is scheduled next.
- **Round Robin (RR)** – The queues are scheduled in a circular manner, i.e. after serving queue i, $1 \leq i \leq M$, the scheduler inspects the queues in the order $i+1, \ldots M, 1, \ldots, i$ and serves the first non-empty queue in that sequence.

Our performance measure is *system's latency* defined as the expected wait time of a customer (i.e., a user's access request).

We denote by $\mathcal{W}_i(\mathcal{B}), \mathcal{W}(\mathcal{B})$ the random variables that gives the wait times associated with queue i and with the system respectively; For a batching scheme $\mathcal{B}$ we compute the latency of queue i and the system's latency, given by $E[\mathcal{W}_i(\mathcal{B})], E[\mathcal{W}(\mathcal{B})]$ respectively.

3 Analysis of the RR and FCFS

3.1 Average Wait Time of the FCFS

We assign to the i-th queue a single dedicated server that walks for a vacation of length d at the end of a service, and inspects the length of the queue at time instants that are multiples of d. The server gives service to the whole queue in time $S = \epsilon \ll 1$[2].

Under the FCFS scheme, the length of vacation $U_i^{\mathcal{F}}$ is determined by the number of non-empty queues that the first arriving customer finds while re-establishing the i-th queue after service. In the

[2]Note, that we do not use S in the computations, and it only accounts for the fact, that the server will inspect the i-th queue again within d time units, that is, at the next service completion in the original system.

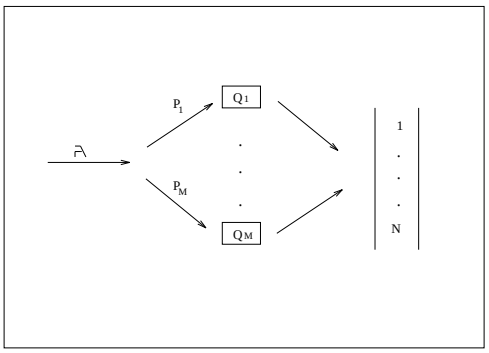

Figure 1: The MOD Queuing System

2.2 A Server of the Walking Type Model

We analyze the performance of a given batching scheme, by considering each of the queues separately: Assume that there is a single server of the walking type in the system, i.e., each service of the i-th queue, that is of length S, is followed by a vacation of length D_i, after which the server keeps checking the queue until it is found nonempty at some time t. The server then walks for a vacation of length $U_i^{\mathcal{B}}$. At the end of the vacation the server empties the i-th queue (i.e. gives service simultaneously to all of its waiting customers). In the context of a multimedia system, the service time S is the amount of time required for emptying queue i immediately after it is scheduled. This amount of time is typically very small relative to the program lengths, thus we take below $S = \epsilon \ll 1$. The random variable D_i denotes the time that elapses from a schedule of queue i till the next inspection of that queue, i.e., when an I/O stream becomes available. The random variable $U_i^{\mathcal{B}}$ denotes here the time that elapses from the arrival of the first customer (since queue i was last emptied) until the queue is scheduled again.

We first analyze the performance of a batching scheme $\mathcal{B}$ for a single queue, and then derive results for a system of M queues, where the arrival rate to queue i is λ_i, $1 \leq i \leq M$. A detailed description of our server of the walking type model for a MOD system is given in Figure 2.

1.3 Organization of the Paper

In Section 2 we introduce our server of the walking type model for Multimedia systems and give some definitions and notation. In Section 3 we use this model for analyzing the FCFS and the RR schemes. In Section 4.1 we derive performance bounds for the MQL, and in Section 4.2 we present the Rounded Ratio algorithm, that yields 2-approximation to the minimal latency of a given multimedia system.

Section 5 describes the results of a simulation study of the proposed schemes.

2 Preliminaries

2.1 The Multimedia-on-Demand System Model

In our queuing analysis of MOD batching schemes each user's request to access a specific object is represented by a customer. Assume, that customers arrive by a Poisson process with interarrival rate λ[1]. An arriving customer joins the i-th queue with probability $0 < p_i < 1$, $1 \leq i \leq M$. The vector $(p_1, \ldots, p_M)$ that gives the distribution on customer requests for the M possible types of service, is called the *request probability vector (rpv)* $\bar{p}$. Throughout the paper we assume w.l.o.g. a renumbering of the queues such that $p_1 \geq \cdots \geq p_M$. We denote by λ_i the arrival rate to queue i, given by $\lambda_i = p_i \lambda$. We assume a server capacity of N I/O streams. A description of the MOD queuing system is given in Figure 1. In the sequel we refer to multimedia objects also as *programs*. We assume in our analysis that programs are of the same length T, and that the arrival rate λ is large enough, so that the probability of finding an *idle* stream (i.e., *all* queues in the system are empty) is small. Thus, we can use the distribution on the start times of busy periods of the streams in the initial state. Assume, that program starts are initially distributed uniformly along an interval of length T, then the time between successive completions is given by $d = \frac{T}{N}$.

[1] In fact our analysis holds for any arrival process that is *memoryless* .

batching schemes, with respect to system's latency. Our mathematical model provides a method for analyzing the performance of two schemes that are natural for the problem – the *First Come First Served (FCFS)* and the *Round Robin (RR)*. Within this framework we also study the *Maximal Queue Length (MQL)* scheme, and show that in some typical scenarios it achieves a ratio that is a small constant to the minimal latency; Using the frequencies of access requests for the various objects, we give a simple algorithm that is a 2–approximation to the optimum.

1.2 Related Work

Most of the earlier related works in the area of Multimedia systems discuss *experimental* studies of various batching schemes (see, e.g., [4, 5]). In [4] a Markovian model was proposed for analyzing a few versions of the FCFS scheme: As the authors point out, this conventional queuing analysis is useful only for small systems; This is due to the size of the state space of the corresponding Markov chain, that grows *exponentially* in the number of queues in the system.

In [2] a multimedia system is viewed as an on-line batching system: The model was used for the investigation of decision rules for adaptive video-on-demand, in which the usage of batching is combined with scheduling algorithms that decide whether to accept or reject user requests in a movie vending environment.

We use in our analysis the *Server of the Walking Type* model that was introduced by Miller in [13]. An alternative model for analyzing queuing systems that possess the batching property is the G-Network, proposed by Gelenbe in [9]. In a related work [10] the G-Network model is used for generalizing the results in this paper to queuing systems with batch service mechanisms, where customers are of the *reneging* type.

scheduler selects the queue that will be served next. This selection depends on the batching scheme used by the system. The server then gives simultaneous service to all customers waiting in that queue in time T, where T is fixed, i.e., independent of the amount of customers served or the states of the other queues in the system. Indeed, our optimization problem arises in any queuing system that possesses this special batching property.

In this work we study the efficiency of batch service mechanisms in Multimedia systems: In the common application of Multimedia-On-Demand (MOD) a large database of multimedia objects is stored in a set of centralized servers; The objects are transferred through high-speed communication networks, by geographically distributed clients [7, 15, 14], where each client serves a group of users in the system. The bottleneck resource at the server could be either disk bandwidth or CPU capacity. Hence there is hard limit on the number of video streams that can be simultaneously delivered by a server. Indeed, there may be a large number of users simultaneously requesting the *same* object, as is the case in commercial environment [6]. Dedicating a stream to each request would require a very large server capacity. Such mode of operation is wasteful also due to the fact that modern communication networks, such as ATM, are equipped with multicast facility [11, 12, 17], that enables the transmission of the same data to multiple users while causing no extra overhead at the server. This feature can be exploited to reduce the number of streams required by the server to support a given number of users: If two requests for the same object are separated by a small time interval, then by delaying the service of the first request, a single stream could be used to satisfy both [3]. The multimedia server will then read data only once from the storage device, and send the data separately to the appropriate clients.

Indeed, users may withdraw access requests, when wait-times become too long, leading to a loss in potential revenue for the system. Thus, a most important aspect of this type of service is providing access instantaneously or within a small *latency* upon request.

In this work we develop an approximate model for evaluating the performance of multimedia

An Analytical Study of Multimedia Batching Schemes

Hadas Shachnai
Department of Computer Science
The Technion, Haifa 32000, Israel. *

Philip S. Yu
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA. †

Abstract

We consider the problem of minimizing the average wait times in a queuing system where a single server gives simultaneous service to a *batch* of customers. The length of this service is fixed and independent of the size of the batch. Our study is motivated by the application of this problem to Multimedia-On-Demand systems, in which user requests for multimedia objects can be serviced in batches. This can be done using the multicast facility available in the high-speed network connecting the multimedia server with the users. Indeed, users may withdraw access request, when wait-times become too long, leading to a loss in potential revenue for the system. Thus, a most important aspect of this type of service is providing access instantaneously or within a small *latency* upon request.

In this work we develop an approximate model for evaluating the performance of multimedia batching schemes, with respect to system's latency. Our mathematical model provides a method for analyzing the performance of two schemes that are natural for this problem – the *First Come First Served (FCFS)* and the *Round Robin (RR)*. Within this framework we also study the *Maximal Queue Length (MQL)* scheme and show that in some typical scenarios it achieves a ratio that is a small constant to the minimal latency; Finally, using the relative frequencies of requests for the various multimedia objects, we give a simple algorithm that is a 2–approximation to the optimum.

1 Introduction

1.1 Problem Statement and Motivation

We consider the problem of minimizing the average wait time in a queuing system where a single server gives simultaneous service to a *batch* of customers. The system consists of M queues and N servers. An arriving customer joins one of the queues, depending on the type of service he requires. Each of the servers can provide service to any of the queues. When a server becomes available the

*Contact author. Part of this work was done while the author was with IBM T.J. Watson, Yorktown Heights, NY. e-mail:hadas@cs.technion.ac.il

†e-mail:psyu@watson.ibm.com