On G-Networks and Resource Allocation in Multimedia Systems

Erol Gelenbe *

Department of Electrical
and Computer Engineering
Duke University
Durham, NC 27709-0291, USA

Hadas Shachnai †

Department of Computer Science
The Technion
Haifa 32000, Israel

Abstract

Multimedia systems are gaining importance as novel computer and communication system architectures, which are specialized to the storage and transfer of video documents. Consider a multimedia-on-demand server who transmits video documents through a high-speed network, to geographically distributed clients. The server accumulates requests for specific documents in separate queues. The queues need to share the transmission medium in some fashion, typically in Round-Robin mode. We describe the resulting performance modeling problem, and develop an approximate representation using queuing networks. Our analytic model enables the efficient implementation of a new scheduling scheme, that we call the Local Round-Robin (LRR). We show that LRR yields significant improvement in system performance, compared to the original Round-Robin.

1 Introduction

1.1 The Multimedia Server System

Multimedia servers are quite different from those of conventional computer file systems, due to their strict timing requirements: In the common application of *Multimedia-On-Demand (MOD)* service subscribers can choose both the program they wish to view and the time they wish to view it [18, 22]. A most challenging aspect in the management of such systems is providing service within a small latency and guaranteeing a sustained and almost constant transfer rate of the multimedia information. Indeed, the high data rates for motion video [15] also imply the need of large storage capacity in the multimedia server, however, in this work we focus on the main bottleneck of *disk bandwidth*, measured as the number of video streams that can be simultaneously delivered by the server.

In multimedia-on-demand systems a large database of video documents is stored in a centralized server. These documents are transferred through high-speed communication network to geographically distributed clients [5, 18, 17]; Modern communication networks, such as ATM, are equipped with multicast facility [13, 14, 23], i.e., the same data can be sent to multiple clients without causing any extra overhead at the server. This feature is exploited to reduce the number of streams required by the server to support a given number of clients: If two requests for the same documents are separated by a small time interval, then by delaying the service of the first request, a single stream can be used to satisfy both [1] (see Figure 1). Thus, time-sharing of a given server capacity is implemented by *batching* requests for the same video documents in one queue, and by serving all of the waiting requests in the queue simultaneously. This mode of operation distinguishes multimedia systems from the traditional computer/communication systems, that can sometimes be modeled as standard queuing networks [6, 11, 7].

In the present paper we develop a queuing model that enables to analyze and compare the performance of multimedia servers under various batching schemes. The formal framework that we use is based on the *G-network* model that was introduced in [8]. The G-network captures some important features of multimedia systems and provides tools for analyzing their performance. Our analysis addresses the general multimedia-on-demand scenario, where waiting customers may lose patience and leave the system without being served, leading to a loss in potential revenue for the system. In the sequel we also explore some optimization problems that arise in the management of a multimedia server with a given stream capacity. In

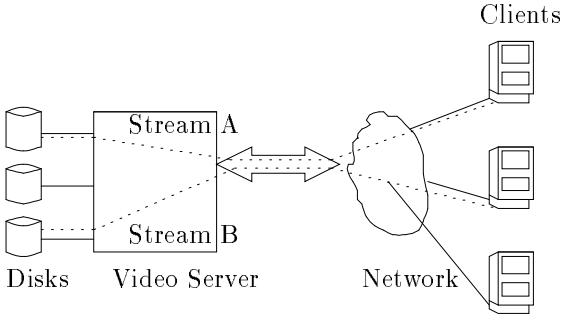

Figure 1: Multimedia-on-demand server environment

*e-mail: erol@ee.duke.edu
†e-mail: hadas@cs.technion.ac.il

particular, we show that using our queuing model, the problem of allocating a fixed server capacity to individual queues so as to maximize the system throughput (or minimize the average wait time) can be solved efficiently, in time that is polynomial in the server capacity and the number of documents.

1.2 Related Work

We mention briefly some of the past work in this area: In [2] a multimedia system is viewed as a special case of an on-line batching system, where no assumptions are made on the arrival times of requests or the frequency of viewing requests for the various documents. The batching property implies that a single server gives simultaneous service to a *batch* of clients, with the length of this service independent of the size of the batch. The model was used for the investigation of various decision rules for *adaptive video-on-demand*, in which a batching scheme is combined with scheduling algorithms, that decide whether to accept or reject client requests in a movie vending environment.

In [19] the problem of finding an efficient schedule for a MOD system was formulated as a stochastic optimization problem, and functional equations were derived, for defining the optimal scheme. It was shown, that the computational complexity as well as the space required to solve these equations can be sizable, and the authors proceed with experimental study of heuristic solutions.

In [4] a Markovian model was used for representing a multimedia system. The authors show that even when simplifying assumptions are used, this conventional method is prohibitive, due to the large state space of the corresponding Markov chain. In [20] a server of the walking type model was proposed for studying the performance of multimedia applications; however, this model is valid only in the special case where the users patience interval is of *infinite* length.

1.3 Outline of the Paper

The rest of the paper is organized as follows:
In Section 2.1 we present the system model and our performance measures. In Section 2.2 we introduce the G-Network formalism. Section 2.3 gives some definitions and the notation used in our study of the *discrete resource allocation problem*, that arises in multimedia systems.
In Section 3 we analyze the multimedia server performance under the Round-Robin (RR) scheme.

In Section 4 we introduce the *Local Round-Robin (LRR)* scheme that assigns a fixed server capacity to each of the queues based on the relative frequency of requests for the corresponding document. We show that efficient allocation of the server capacity to the queues can be found using our analytic model and the relation of the LRR scheme to the discrete resource allocation problem.
Section 5 describes the results of a simulation study of the Round-Robin and the LRR, and a comparison to the results obtained using our analytic model. We summarize in Section 6 with a discussion of possible directions for future work.

2 Modeling and Mathematical Preliminaries

In this section we first describe our system model and the performance measures that are of interest. Then we introduce the G-network formalism and the resource allocation problem, that will be discussed in the following sections.

Consider a database of M video documents. User requests (to view specific documents) arrive as Poisson process with rate Λ; a user chooses to view document i with probability p_i, $0 < p_i < 1$, where $\sum_{i=1}^{M} p_i = 1$. The probabilities $p_1, \ldots, p_M$ reflect the relative popularity of each document. A user request for a document must provoke a transmission of that document through a multimedia server, whose output is then sent through a high-speed network to the user destination; a document transmission is associated with a delivery of a *data stream* from the I/O subsystem at the server to the user. Since the server can multicast the document over the network, a single data stream can be used to service viewing requests of multiple users, that are all queued for that document. Thus, all users waiting to view document i, $1 \leq i \leq M$, will be serviced simultaneously. The system overhead for such service is determined by the length of the ith document, denoted by L_i.

We assume a server capacity of N data streams, that is, at any given time at most N data streams can be delivered by the server. In addition, users may lose patience and withdraw their requests if the server does not respond within a certain amount of time.

The type of service provided to multimedia users can be modeled as a queuing system that consists of M queues; Customers arrive to the system by Poisson process with rate Λ; a customer joins queue i with probability p_i, $0 < p_i < 1$. Thus, the arrival rate to queue i is $\Lambda_i = p_i\Lambda$. There are N identical servers in the system, that provide batch service, that is, an available server can choose any non-empty queue and service simultaneously all the customers in this queue. The service time of queue i is some constant L_i, that does no depend on the length of queue i. We model the *impatience property* of the users in queue i by a Poisson departure process with rate λ_i, $1 \leq i \leq M$. Our queuing model is depicted in Figure 2.

2.1 Performance Measures and Notation

We use in our analysis of a given batching scheme $\mathcal{S}$ the following performance measures:

- The *System Throughput* is the average number of requests the system completes handling per time unit. We define the *turn-away probability* as the ratio between the number of reneging viewers and the total number of arrivals in a given time interval. Thus, a scheduling strategy that maximizes the system throughput minimizes the turn-away probability.

- The *wait time* of a user request that is serviced by the system, is the time that elapses from the arrival of that request until the server starts the transmission of the corresponding document to

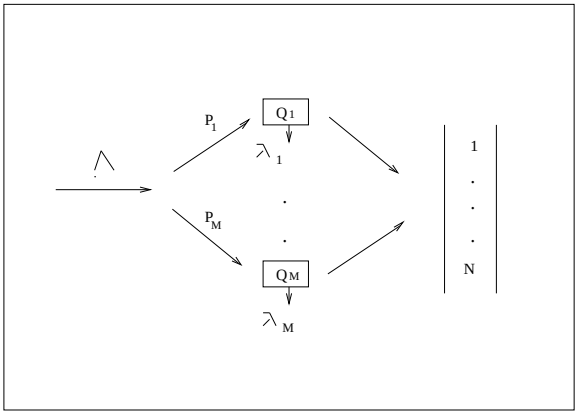

Figure 2: Queuing System Model for the Multimedia Server

the user. Our objective is to minimize the average wait time of the system.

We now define the random variables used in our analysis:

$Q_{\mathcal{S}}(i)$ – length of queue i.
$W_{\mathcal{S}}(i)$ – wait time of queue i, i.e., the time that a user request waits for service.
$T_{\mathcal{S}}(i)$ – throughput of queue i.
$T_{\mathcal{S}}$ – total throughput of the system.
We are interested in the mean values of these random variables, denoted by $\overline{Q}_{\mathcal{S}}(i)$, $\overline{W}_{\mathcal{S}}(i)$, $\overline{T}_{\mathcal{S}}(i)$ and $\overline{T}_{\mathcal{S}}$ respectively.

2.2 The G-Network Model

The G-Network model turns out to be a convenient tool for the analysis we need to carry out. We outline below the principal properties of these networks that will be used for deriving our results.

When viewed as a queueing network, the G-network has essentially two types of customers: positive and negative. Positive customers have the same behavior as ordinary queueing network customers. If a positive customer joins a queue it waits until it receives service or it can be destroyed or displaced by a negative customer arriving to the queue. A negative customer joining a non-empty queue has the power to *destroy* one positive customer or a batch of positive customers [10], or it will vanish immediately if the queue is empty. Thus, negative customers do not receive service and their actions are assumed to be taken instantaneously.

External customer arrivals to queue i constitute independent Poisson processes with rate Λ_i for positive customers and rate λ_i for negative customers. Positive customers have *iid* exponential service distribution times with rate μ_i at queue i. A positive customer leaving a queue after the completion of service may join another queue either as a negative or as a positive customer. The movement of customers between queues is represented by a Markov chain. A positive customer leaving queue i (upon service completion) joins queue j as a positive customer with probability $P^+_{i,j}$, or as a negative customer with probability $P^-_{i,j}$. Then $P_{i,j} = P^+_{i,j} + P^-_{i,j}$ represents the transition probability of a Markov chain modeling the movement of customers between queues. The customer may leave the network with probability d_i. Thus, in a system of M queues we have the following relation:

$$\sum_{j=1}^{M} P^+_{i,j} + \sum_{j=1}^{M} P^-_{i,j} + d_i = 1 \qquad 1 \le i \le M$$

In our multimedia system model, each queue is a special case of a G-network queue with positive and negative customers and batch removals. As each single service of queue i empties the queue (i.e. all waiting requests are serviced simultaneously), a service may be viewed as
(i) service of the first customer in the queue, followed by (ii) an instantaneous return of that customer to the queue as a negative customer (with $P^-_{i,i} = 1$), that removes all the remaining customers (i.e. a batch of customers of unlimited size) from the queue.
The withdrawal of a user request from queue i can be modeled as arrival of a negative customer, that removes a single positive customer from queue i.

2.3 The Discrete Resource Allocation Problem

In Section 4 we discuss a scheduling approach that is based on a partition of a given server capacity to M subsets of data streams. Each subset is allocated to a single queue. For finding the desired partition, we use an optimal solution for the discrete resource allocation problem, defined as follows:

DRA: Given N resources, M objects and a non-decreasing function $f_i(N_i)$, find an allocation vector $\vec{N} = (N_1, \ldots, N_M)$

$$\text{that maximizes } \sum_{i=1}^{M} f_i(N_i)$$

$$\text{Subject to } \sum_{i=1}^{M} N_i = N$$

$$\text{such that } N_i \in \{0, 1, \ldots, N\},$$
$$i = 1, 2, \ldots, M.$$

Ibaraki and Katoh discuss in [12] several algorithms for solving the DRA, and conclude with the following result:

Theorem 1: *Given the values of $f_i(N_i)$, $\forall\ 1 \le i \le M$, $1 \le N_i \le N$, an optimal solution for the DRA can be found in $O(MN^2)$ steps.*

The solution is obtained by an algorithm called DP, that employs standard dynamic programming techniques (see Ch. 3.3 in [12]).

3 Round-Robin Scheduling of the Multimedia Server

Consider the implementation of the Round-Robin scheme in the queuing system of Figure 2. Let $L_i = L$ denote the length of a service of queue i, $1 \leq i \leq M$ [1], and denote by $D = L/N$ the *inspection frequency parameter* of the system. The queues are inspected at time instances that are integral multiples of D. (By tuning D, the exhaustive service property of the system can be better utilized. That is, D can be increased to guarantee, that the scheduled batches of requests will not be too small).

Assume that initially all servers are idle. Since our N servers represent N data streams that are operated by a single multimedia server, in the corresponding queuing system we assume a sequential schedule of the servers. Thus, at time rD, $r \geq 0$, a single server can start giving service to one of the queues.

The Round-Robin scheme maintains a list of available servers, and a pointer to queue i, that was scheduled latest for service. At any time instance $t = rD$,

- The next available server inspects the queues in the order $i+1, \ldots, M, 1, \ldots, i-1, i$, until it finds the first non-empty queue[2].
- If all queues are empty, then the server remains idle and starts another circular inspection of the queues at time $t = (r+1)D$.

A pseudocode of Round-Robin is given in Figure 3.

Let q_j denote the stationary probability that the j-th queue is non-empty. Then the expected delay between visits to queue i is given by $D \sum_{j \neq i} q_j$, and the total average service time perceived at the i-th queue is given by

$$\tau_i = D(1 + \sum_{j \neq i} q_j) \ , \tag{1}$$

and the service rate is $\mu_i = [\tau_i]^{-1}$.

We now turn to the calculation of the queue length distribution for each document. Our queuing system is a special case of a product form G-network (see in [8, 9, 10]). In particular, when there are n_i requests in queue i, the queue length can either

- increase (with rate Λ_i) to $(n_i + 1)$, due to arrival of a positive customer.
- decrease (with rate λ_i) to $(n_i - 1)$, due to arrival of a negative customers that 'kills' a single positive customer.
- decrease to zero (with rate μ_i), due to arrival of a negative customer that 'kills' all the positive customers in the queue.

[1] We simplify the calculations by assuming that all documents are of the same length. In Section 4 we consider the general case, where each document has distinct length.

[2] The inspection time of a single queue is some small constant $B \ll D$, therefore services will not be delayed due to inspections.

```
r := 1;
D := L/N;
pointer := 0;
t :=Current_Time();
Insert the Servers {1, ..., N} to idle_list.
repeat
   while (Current_Time − t) < rD do;
   if (idle_list non-empty) then
      k := header(idle_list);
      i := (pointer mod M) + 1;
      while ((queue(i) is empty) and (i ≠ pointer))
            i := (i mod M) + 1;
      if (queue(i) is non-empty) then
         pointer := i;
         Mark server(k) as busy;
         Delete server(k) from idle_list;
   r := r + 1;
until false;
```

Figure 3: Algorithm *Round-Robin*:

Let $p(n_i)$ denote the steady state probability for n_i requests in queue i. For $n_i > 0$ the global balance equation for $p(n_i)$ is

$$p(n_i)[\Lambda_i + \lambda_i + \mu_i] = \Lambda_i p(n_i - 1) + \lambda_i p(n_i + 1). \tag{2}$$

For $n_i = 0$ we have

$$p(0)\Lambda_i = \lambda_i p(1) + \mu_i \sum_{k \geq 1} p(k). \tag{3}$$

The solution to this system of equations can be written as

$$p(n_i) = q_i^{n_i}(1 - q_i), \tag{4}$$

where q_i has the form

$$q_i = \frac{\Lambda_i}{\lambda_i + \frac{\mu_i}{1-q_i}}. \tag{5}$$

From (1) we have

$$q_i = \frac{\Lambda_i}{\lambda_i + 1/((1-q_i)(D(1+\sum_{j \neq i} q_j)))} \ , \tag{6}$$

which yields a system of M non-linear equations for the q_i's. For a given M a close approximation of the q_i's can be obtained using numerical iterative methods for solving such system of equations (e.g., Newton-Raphson [16]). This gives the steady state distribution, from which we can calculate the average wait time of queue i under Round-Robin:

$$\overline{W}_{\mathcal{RR}}(i) = \frac{\overline{Q}_{\mathcal{RR}}(i)}{\Lambda_i} = \frac{\sum_{n_i \geq 0} n_i p(n_i)}{\Lambda_i} \ . \tag{7}$$

The system average wait time is given by

$$\overline{W}_{\mathcal{RR}} = \sum_{i=1}^{M} p_i \overline{W}_{\mathcal{RR}}(i) \ . \tag{8}$$

The throughput of queue i is

$$\overline{T}_{\mathcal{RR}}(i) = \overline{Q}_{\mathcal{RR}} \mu_i \ , \tag{9}$$

and, the throughput of the system is

$$\overline{T}_{\mathcal{RR}} = \sum_{i=1}^{M} \overline{T}_{\mathcal{RR}}(i) \ . \tag{10}$$

4 Multimedia Scheduling and the Resource Allocation Problem

In this section we study a scheduling approach that is based on the assignment of a fixed portion of server capacity to each of the queues. Thus, the ith queue has a set of I/O streams that are used only for serving viewing requests for the ith document. The resulting queuing system is depicted in Figure 4.

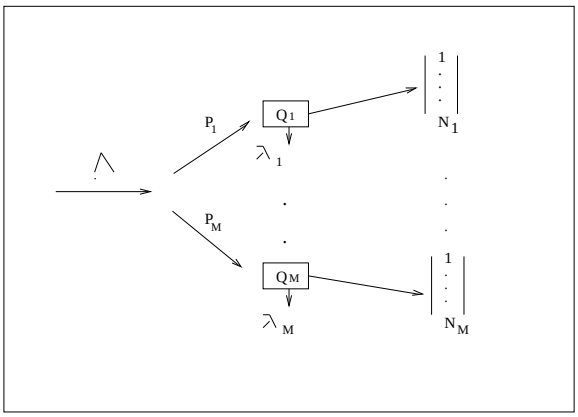

Figure 4: Queuing System Model of the LRR Scheme

For a server capacity of N streams, the *Local Round-Robin (LRR)* scheme allocates N_i streams to document i, $1 \leq i \leq M$. The LRR schedules each of the queues by the Round-Robin algorithm, as given in Figure 3, with $M = 1$, and $N = N_i$ for queue i, $1 \leq i \leq M$.

Thus, the implementation of LRR consists of two phases:

(i) *Preprocessing phase:* Find an efficient allocation of server capacity to the M queues (i.e., define the allocation vector $(N_1, \ldots, N_M)$).

(ii) *Scheduling phase:* For $i = 1, \ldots, M$ let L_i denote the length of document i: Schedule queue i at time instances rD_i, $r \geq 0$, where $D_i = L_i/N_i$, by the algorithm Round-Robin.

For the first phase, we can find efficient partition of the server capacity to the queues, e.g., by using the algorithm DP [12]. In order to find such a partition, we need to compute $f_i(N_i)$ $\forall$ $1 \leq i \leq M$ and for any $1 \leq N_i \leq N$. Typically, the functions $f_1, \ldots, f_M$ are *unknown*. Indeed, $f_i(N_i)$ can be estimated by running a simulation of M sub-systems, each consisting of a single queue, with N_i servers. Then we can measure the system performance. However, since we need to calculate for sub-system i $f_i(N_i)$, $\forall$ $1 \leq N_i \leq N$, running a simulation becomes impractical for larger values of M and N.

In the following we propose to use our analytic model for the estimation of the functions $f_1, \ldots, f_M$. Since $f_i(N_i)$ depends on the performance measure of interest, we exemplify by taking one of the measures discussed in Section 3. In particular, suppose the queues are scheduled by the LRR, and we look for an allocation vector $(N_1, \ldots, N_M)$ that maximizes the system throughput. Let $\overline{T}_{\mathcal{LRR}}(i)$ denote the expected throughput of queue i. We define

$$f_i(N_i) = \overline{T}_{\mathcal{LRR}}(i) \ .$$

The results in Section 3 can be applied as follows. Taking $M = 1$ and $N = N_i$, we have $D_i = L_i/N_i$. From (5) we can write

$$f_i(N_i) = \frac{q_i}{(1 - q_i)\Lambda_i} \tag{11}$$

where

$$q_i = \frac{B - \sqrt{B^2 - 4\Lambda_i/\lambda_i}}{2} \tag{12}$$

and

$$B = 1 + \frac{1}{\lambda_i D_i} + \frac{\Lambda_i}{\lambda_i} \ . \tag{13}$$

5 Numerical Results

We give below the numerical results obtained in a simulation study of the multimedia server system. Our results validate the mathematical model used in Section 3 and evaluate its relative efficiency in the implementation of the LRR scheme. We simulated a multimedia system with a database of $M = 100$ documents; user requests arrived by Poisson process. The *rpv* $\bar{p}$ formed a Zipf's distribution, i.e., $p_i = 1/(iH_M)$, where H_M, the Mth Harmonic number, is the normalization constant. The Zipf's distribution often reflects the relative frequencies of accesses to multimedia objects, e.g., in a movie vending environment [21]. We used a server capacity of $N = 800$ streams.

We assumed the following reneging pattern of user requests: After waiting an amount of time given by the wait threshold T, the user's remaining time until departure is given by the random variable R, which is determined by an Exponential distribution. That is, $R \sim exp(\nu)$, where $1/\nu$ is the mean time that elapses after T until the viewer leaves the system. Thus the viewer's overall wait time is given by $W = T + R$. We used $T = 5, 10$ minutes and $\nu = 0.15, 0.33$ respectively. In our analytic model we assumed arrival of negative

customers to queue i, with rate $\lambda_i = \Lambda_i(1 - (T + 1/\nu)/\tau_i)$, where τ_i is given in (1).

In Figures 5 and 6 we compare the performance of the RR as predicted by the G-Network model with the results obtained in our simulation. Figure 5 shows

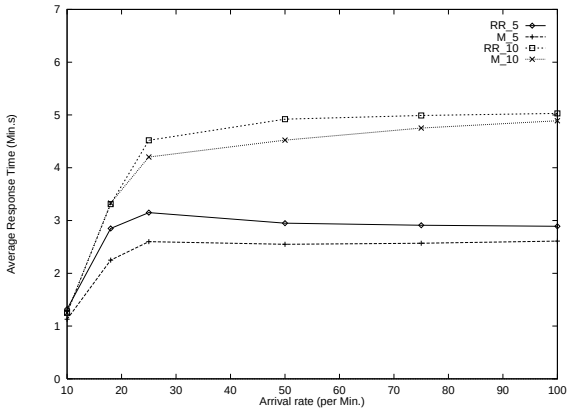

Figure 5: Average Wait Time - Comparison of Simulation and Analytical Model

system's average wait time under RR. We used fixed server capacity and increasing loads, taking request arrival rates in the range 10 to 100 per minute. We note, that the estimated wait time is closer to the average wait time measured in the simulated system as T grows larger, and for both values of T the discrepancy between the two is bounded by 15%. In Figure 6 we present similar results in the comparison of the estimated throughput with the throughput of our simulation.

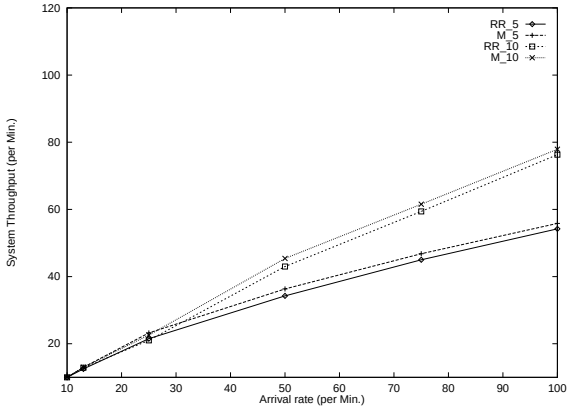

Figure 6: System Throughput - Comparison of Simulation and Analytical Model

In Figure 7 we compare the system throughput obtained by the LRR scheme, when the allocation of streams to queues is done by our analytic model or by running a simulation for every queue and each allocation vector $(N_1, \ldots, N_M)$. We first note, that the LRR scheme improves significantly the performance

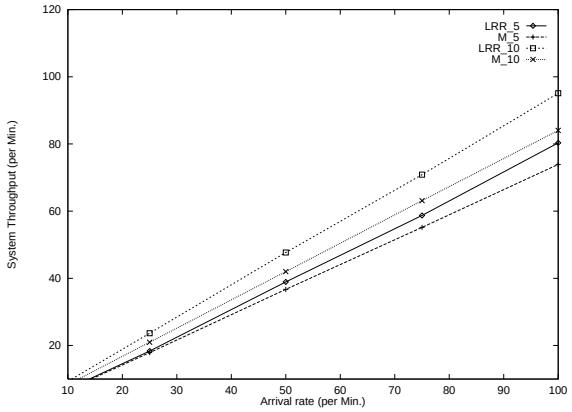

Figure 7: LRR - Throughput vs. Arrival Rate

of Round-Robin, especially for higher loads (e.g., for $\Lambda = 100$ and $T = 5$ the ratio $\overline{T}_{\mathcal{LRR}}/\overline{T}_{\mathcal{RR}}$ approaches 1.5). Figure 7 shows that our estimated implementation of the LRR, which reduces significantly the runtime of the allocation algorithm, provides a 1.14 approximation to the optimal throughput (computed via simulation).

6 Discussion

We have studied the performance of two service schemes for multimedia systems, based on a Round-Robin schedule, namely, the original RR and a new scheme called LRR. The usage of the G-Network formalism enabled to derive analytic results for the steady state distribution of such systems under these schemes, and to obtain efficient partition of the server capacity to queues, for the implementation of the LRR. Our numerical study showed that LRR can improve significantly the performance of the system, compared to the commonly used RR.

Indeed, our analytic model can be used also in configuration planning for multimedia systems, e.g., for finding the minimal server capacity that guarantees a desired average wait time or throughput under a given batching scheme. Alternatively, for a fixed server capacity, our results provide a bound on the maximal amount of documents that enable to achieve a desired quality of service.

While the RR requires no extra knowledge of *system* parameters, the LRR uses the relative frequencies of requests for documents. Therefore, a given allocation vector $\vec{N}$ is optimal as long as the request probability vector $(p_1, \ldots, p_M)$ is fixed. It would be of interest to devise a dynamic version of the LRR scheme, that adapts the partition of server capacity to the queues to the changes in the statistical behavior of the system.

Finally, we discussed allocation vectors that were optimal with respect to a single criterion. In the context of multimedia systems, it is sometimes desirable to combine several performance measures; e.g., for finding an allocation vector $\vec{N}$ that minimizes the system average wait time, subject to the constraint that

the expected loss of users (due to reneging) is bounded by a given $0 \leq \alpha < 1$.

References

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

[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] Baskett F., Chandy K.M., R.R. Muntz, Palacios F.G., *Open, Closed and Mixed Networks of Queues with Different Classes of Customers*, Journal of ACM, 22(2):248-260 (Apr. 1975).

[4] A. Dan, D. Sitaram and P. Shahabuddin, "Scheduling Policies for an On-Demand Video Sever with Batching", *Proc. ACM Multimedia '94*, SF, CA, Oct. 1994, pp. 391-398.

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

[6] Ferrari D., Computer Systems Performance Evaluation, Prentice-Hall, INC, 1978.

[7] Ferrari D., Serazzi G. Zeigner A., Measurement and Tuning of Computer Systems,Prentice-Hall, INC., 1983.

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

[9] Gelenbe E., *G-Networks with Triggered Customer Movement*, Journal of Applied Probability, Vol. 30, pp. 742-748 (1993).

[10] Gelenbe E., *G-Networks with Signals and Batch Removals*, Probability in Engineering and Informational Sciences, Cambridge University Press, England, Vol. 7, pp. 335-342 (1993).

[11] Gelenbe, E., Mitrani I., Analysis and Synthesis of Computer Systems, Academic Press, 1980.

[12] T. Ibaraki and N. Katoh, *Resource Allocation Problems - Algorithmic Approaches*, The MIT Press, 1988.

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

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

[15] International Organization for Standardization, Coding of Motion Pictures and Associated Audios - for digital storage media at up to 1.5 Mbits/s. IS 11172, Nov. 1992.

[16] Ralston A., Rabinowitz P., "A First Course in Numerical Analysis", *McGraw-Hill Kagakusha, Ltd.* 2nd Edition, 1978 (Ch. 8.8).

[17] P. V. Rangan and H. M. Vin, "Designing File Systems for Digital Video and Audio", *Proc. of 12th ACM Symposium on Operating Systems*, 1991.

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

[19] Shachnai H., Yu P. S., "The Role of Wait Tolerance in Effective Batching: A Paradigm for Multimedia Scheduling Schemes". To appear in Multimedia Systems Journal.

[20] Shachnai H., Yu P. S., "On Analytic Modeling of Multimedia Batching Schemes", *Proceedings of the 3rd International Workshop on Multimedia Information Systems (MIS'97)*, Como, September 1997.

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

[22] H. M. Vin and P. V. Rangan, "Designing a Multi-User HDTV Storage Server", *UCSD Technical Report CS92–225*, Jan. 1992.

[23] 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.