Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Discrete Applied Mathematics 129 (2003) 49–61
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Competitive video on demand schedulers for popular movies ☆

Christos Bouras[a,b,*], Vaggelis Kapoulas[a,b], Grammati Pantziou[c], Paul Spirakis[a,b]

[a] *Research Academic Computer Technology Institute, Riga Feraiou 61, 262 21 Patras, Greece*
[b] *Department of Computer Engineering and Informatics, University of Patras, 265 00 Rion-Patras, Greece*
[c] *Department of Informatics, Technological Education Institute of Athens, Ag. Spyridonos Str., 122 10 Egaleo-Athens, Greece*

Abstract

In this paper we investigate the online video on demand problem, namely having to accept or reject a request for a movie without knowing the future requests. We present online movie-scheduling schemes that implement the principles of refusal by choice and delayed notification. A novel way to schedule movies that exploits the knowledge of the distribution of the preference of requests for movies, is shown to have a competitive ratio that outperforms all the previously known schemes in practical situations. In fact, our scheduler has a competitive ratio bounded above by a constant, independent of the number of the users, channels, or movies, in the case that a large fraction of the requests tends to concentrate in a small number of movies. We extend our approach by presenting an "adaptive" randomized scheduler which initially is not aware of the movie popularities but it adapts to it, and achieves the same asymptotic competitive ratio.

Keywords: Video on demand; Online scheduling algorithm; Competitive ratio; Probability distribution

1. Introduction

The area of interactive home video entertainment is actively developing. For example, hybrid networks that enable multimedia connections are being studied as a step towards

☆ This research was partially supported by the European ESPRIT Long Term Research Projects GEPPCOM (contract No. 9072) and ALCOM-IT (contract No. 20244), and the Greek Ministry of Education.
* Corresponding author. Computer Technology Institute, P.O. Box 1122 GR-26110 Patras, Greece.
E-mail addresses: bouras@cti.gr (C. Bouras), kapoulas@cti.gr (Vaggelis Kapoulas), pantziou@teiath.gr (Grammati Pantziou), spirakis@cti.gr (Paul Spirakis).

0166-218X/03/$ - see front matter
PII: S0166-218X(02)00233-0

all digital video networks [14], as is the problem of bandwidth allocation strategies for combined analog/digital transmission of data over a CATV system [12]. Experiences and service trials are being performed, mostly for near video on demand, which consists of broadcasting movies at fixed time intervals (e.g. every 10 min) [19,22]. Also, special hardware and switches are being built for this purpose.

Recent advances in computing and communication technology have made feasible video on demand systems. Usually, the movies are stored in a central video server (which may be connected to other servers by a high-bandwidth WAN). The video server is connected via a high-capacity fiber line to local distribution centers (hubs) from which coax cables are used to broadcast to the households.

In this architecture, there appear to be two major bottlenecks:

- The limited number of broadcast channels available on the coax cable (shared by many households).
- The number of movies which the server is able to transmit concurrently (see [18]).

Previous works and current implementations attempt to achieve under these constraints:

- *Full video on demand*: Whenever a user's request for a movie arrives, it is immediately served, provided there is capacity (a channel) available; otherwise the request is rejected.
- *Near video on demand*: A fixed set of movies is played regularly, at fixed time intervals.

Much attention has been given to the hardware requirements for such designs, and the quality of service, due to the bottlenecks mentioned above. It is immediate that either policy may poorly utilize the available resources. A considerable number of feasible distributions of requests may cause the first approach to tie up the distribution resources, while the latter provides (by definition) a limited service, and a bad decision (of movies broadcasted) translates directly into resource waste.

The intermediate terrain that lies between the two extreme policies was investigated by Aggarwal et al. [1], where the concept of adaptive video on demand was introduced:

- *Adaptive video on demand*: Upon arrival, a movie request is accepted and served (possibly with some delay), or rejected. The decision as to whether accept or reject (and the amount of delay) is made by a scheduling algorithm.

Note that the scheduling algorithm might specify that the request be rejected, even though there might be channels that are currently not being used.

Different issues involved in the design of a video on demand system have been studied by different researchers the last few years. Architectural issues have been studied in [17,24], physical storage organizations necessary for supporting video on demand systems in [8,9,21], and probabilistic models for the assignment of video data onto a storage hierarchy in [20]. The first attempt to tackle video on demand from an optimization perspective was done by Aggarwal et al. [1]. In that important work, the video on demand problem was studied in an online setting, where an online algorithm receives a sequence of requests for service. The performance of the online algorithm on a sequence of requests is compared to the performance of an optimal offline algorithm that services the *same* sequence of requests. Such an analysis of an online algorithm is referred to as *competitive* analysis. Aggarwal, Garay and Herzberg showed upper and lower bounds on the competitive ratio of online scheduling algorithms for certain

scenarios, and also introduced the concept of refusal by choice with delayed notification and presented algorithms that exhibit under certain conditions, an asymptotically optimal behaviour.

Refusal by choice (and, in fact, by *random choice*) was used previously in the problem of admission control in fast networks (see [2–4,13]). The admission control problem, first defined by Garay and Gopal [10], is the problem of deciding online whether or not a network should accommodate a request for a large amount of data. The online video on demand research was complementary to the research for the admission control problem. While the research on admission control was mostly concerned with online allocation of network paths in a way that would minimize overlap with paths of future requests, the adaptive video on demand research focuses mostly on the issue of "revenue" of movie schedulers (over very simple networks). The revenue has to do with grouping requests so that a single transmission may serve many users requesting the same movie (popular movies). Also, the problem of online video on demand is related to the call control problem (see [3,11]).

One interesting observation is that most requests tend to concentrate on (a few) popular movies. This property has been observed and used in the area of scheduling and storage management in VoD systems [16,23,25]. In this paper (of which shorter versions are [6,7]) we present a novel movie-scheduling scheme which exploits the knowledge of the underlying distribution of movie requests, and achieves constant competitive ratio in the case that the number of popular movies is small (which is a realistic assumption). Our method follows the principle of refusal by choice with delayed notification. We also present a randomized movie-scheduling algorithm which does not need to know the distribution of the movie requests in advance but it is able to follow slow changes in this distribution, in an adaptive way that has a small transient behaviour. Our scheduler will adapt to such an unknown distribution quickly (statistically learns). This method also achieves constant competitive ratio in the case that the number of popular movies is small. This is due to the assumption on a distribution of the input requests (which restricts the oracles that would create worst-case behaviours in the lower bounds of [1]).

The rest of the paper is organized as follows. In Section 2 we present the video on demand architectural model used, and some basic definitions. In Section 3 we present the online movie scheduling algorithm S that knows the distribution $p()$ of the movie requests, and in Section 4 we analyze its performance against an optimal offline algorithm. In Section 5 we extend our approach by presenting an online scheduling algorithm R which is not aware of $p()$ but it adapts to it, and we discuss its performance.

2. The model and definitions

The video on demand model considered here follows [1], as far as the architecture is concerned. It consists of a video server which acts as a database of movies, and supports a fixed number of movie-streams (sessions). The users connect to the server via dedicated links and make movie-requests to it. The communication network used is equipped with a multicast facility. Thus, the same movie-stream can be sent to more

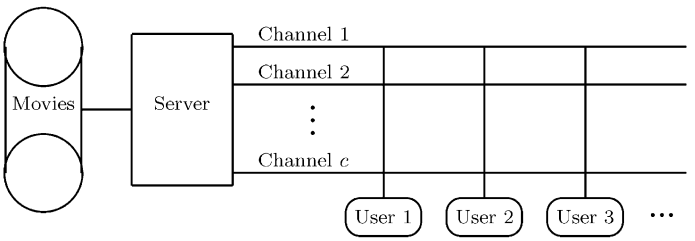

Fig. 1. The model.

than one users without causing any extra overhead to the server, and therefore, multiple users can participate in a single session. Let M be the set of movies stored in the server, U be the set of users making requests, and C be the set of channels in the system, and let m, u, and c be their cardinalities, respectively. The system is as in Fig. 1.

The system has three time parameters:

T: The *duration* of a movie (we assume it is the same for all movies).

τ: The maximum delay between a request and the start of the movie (if the request is accepted).

v: The *notification time* (the maximum delay between a request and its response).

Obviously, $v \leqslant \tau \leqslant T$. In this paper we consider $\tau = v$.

Requests to the system are triples $(q\_time, user, movie)$ and responses are tuples $(r\_time, user, movie, channel, servetime)$, where r_time is a positive number in the interval $[q\_time, q\_time + v]$, $user \in U$, $movie \in M$, $channel \in C$, and $servetime$ is a real number in the interval $[r\_time, q\_time + \tau]$. By convention, a refusal to serve is a response with negative $servetime$.

A sequence of requests (t_i, u_i, m_i) is valid if the time values t_i are monotonically increasing and there is at least time T between any two subsequent requests of the same user, i.e. we assume that each user can place at most one request in each interval of time T. We, also, assume that users are "blocked" when seeing a movie, i.e. they cannot place other requests at that time.

A scheduling algorithm determines the responses of the system to the users at any moment, and allocates movies to channels based on the requests to the system up to the moment the scheduling is taking place. If $v = 0$ then a request must be responded as soon as is it is presented. In this case the scheduling algorithm performs "immediate notification". Otherwise (i.e., $v > 0$), the scheduling algorithm performs "delayed notification" and the response to the user is issued at most v time units after the presentation of the request. More formally a scheduling algorithm is a function from a time, a state and a sequence of requests (all before the given time), to a time (for next wakeup, if no requests would occur before), a new state, and possibly one or more responses.

A scheduling algorithm is said to "refuse by choice" if it has a response (t, u, m, c, s) with negative servetime s, while there exists a free channel at the response time t. A scheduling algorithm is allowed to use random choices as a step (randomized scheduling algorithm).

Assumption 1 (The popularity assumption). Let r be a request and e a movie. We assume that the request r will be for movie e with probability $p(e)$, i.e., we assume that requests are independently created according to the distribution $p()$ which indicates the movie popularities.

Smart schedulers accumulate arriving requests into queues Q_j one for each movie e_j. The idea is to serve for each j, all the requests for movie e_j accumulated into Q_j with a single transmission (over a single channel).

We assume that we know a class of distributions $\mathbb{D}$, such that the input is generated by some distribution $D \in \mathbb{D}$. The online algorithm ALG knows $\mathbb{D}$ and chooses a process that attempts to perform well for all $D \in \mathbb{D}$. The adversary, seeing ALG then chooses $D \in \mathbb{D}$ in order to make the ratio of the expected performance of ALG to the expected performance of OPT (the optimal off-line algorithm, i.e. the algorithm that knows the whole request sequence in advance, and achieves the maximum revenue out of each request sequence) as small as possible. That is, we define

$$ALG(D) = E_\sigma[ALG(\sigma) \mid \sigma \text{ generated by } D],$$

where $ALG(\sigma)$ is the (expected over ALG's random choices, if ALG is randomized) revenue of ALG on input sequence σ.

We say that ALG is δ-competitive against the class $\mathbb{D}$ ($\delta > 1$) if there exists a constant a such that $\forall D \in \mathbb{D}$

$$OPT(D) \leqslant \delta ALG(D) + a.$$

The $\mathbb{D}$-restricted competitive ratio of ALG, denoted $\delta(ALG, \mathbb{D})$ is then defined as

$$\sup\{\delta \mid ALG \text{ is } \delta\text{-competitive against the class } \mathbb{D}\}.$$

Note that this is exactly the restricted Bayessian Compromise ([5, p. 74]) and exactly the approach advocated in [15] ([5, p. 232]).

Also note that $\mathbb{D}$ in our case is the class of general k-skewed distributions (see next pages in Section 3 for a precise definition of this class of distributions which technically tries to capture the fact that requests concentrate to a few *popular* movies).

3. The online movie-scheduling S

3.1. The class of general k-skewed movie request distributions

If n users place overlapping requests (i.e. requests than can be served concurrently) then, by using the popularity assumption, we have

$$\begin{aligned} P_i^e &= \text{Prob}\{\text{movie } e \text{ will be chosen by exactly } i \text{ users}\} \\ &= \binom{n}{i} p^i(e)(1 - p(e))^{n-i}, \end{aligned}$$

where $p(e)$ is the probability of selection of movie e in the popularity assumption.

Recall that each user can place at most one request.

Definition 2. Let

$$Q_i^e = Prob\{e \text{ will be chosen by at least } i \text{ users}\}.$$

Clearly, then

$$Q_i^e = \sum_{j=i}^{n} P_j^e.$$

Definition 3. Let $\tilde{f}_i$, f_i be the actual and the expected number of movies that will be chosen by at least i users to be seen "concurrently".

Definition 4. Let $y_i^e = 1$ with probability Q_i^e and 0 otherwise.

Then, clearly

$$E(y_i^e) = Q_i^e$$

and

$$\sum_e y_i^e = \tilde{f}_i.$$

So,

$$\begin{aligned} f_i &= E(\tilde{f}_i) \\ &= E\left(\sum_e y_i^e\right) \\ &= \sum_e Q_i^e \end{aligned}$$

by linearity of expectation.

(We may use $n = u$ for the worst case demands. Actually, one may adaptively use n equals u minus the number of users that are currently seeing a movie.)

Assume now that the sequence $f_1, f_2, \ldots, f_n$ can be partitioned into an, independent of n, (fixed) number of k subsequences

$$\{f_1\}, \{f_2, \ldots, f_{j_2}\}, \ldots, \{f_{j_{l-1}+1}, \ldots, f_{j_l}\}, \ldots, \{f_{j_{k-1}+1}, \ldots, f_n\},$$

where

$$f_1' = f_1$$

and for $l > 1$

$$f'_l = \sum_{i=j_{l-1}+1}^{j_l} f_i$$

so that the new smaller sequence $f'_1, f'_2, \ldots, f'_k$ satisfies the following two rules:

Rule 5:

$$\forall i, \quad i f'_i \geqslant f_1$$

and

Rule 6: *If* $F = \sum_{i=1}^{k} f'_i$ *then*

$$F \leqslant f_1 + f_1 G(k),$$

where $G(k)$ *is a slowly increasing* (*reversible*, *monotone*) *function of* k *so that* $\lim_{k\to\infty} G(k)/k = 0$.

Definition 7. The class of movie request distributions for which such a partition exists, is called the class $\mathbb{D}$ of *general k-skewed* distributions.

As an example let us use for Rule 6 the more restricted assumption that

Rule 8:

$$f'_i \leqslant \frac{f_1}{i-1} \quad \left(\text{and } f'_k \leqslant \frac{f_1}{k-1}\right)$$

Then, $G(k) = 1 + \frac{1}{2} + \cdots + 1/(k-1) = H_{k-1}$ (the $(k-1)$th Harmonic number) and, in fact, $H_k = \ln k + \gamma + \mathrm{O}(1/k)$, where γ is the Euler's constant.

Definition 9. The class of movie request distributions for which there is a partition of $\{f_i\}$ following Rules 5 and 8, is called the *strict k-skewed* distributions.

Note that k can be independent of u, c, m. The actual value of k depends on the way that the sequence $f_1, f_2, \ldots$ decreases. For example, if the sequence $f_1, f_2, \ldots$ decreases as the sequence $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (i.e. $f_1 = 128$, $f_2 = 64$, $f_3 = 32$, $f_4 = 16$, $f_5 = 8$, $f_6 = 4$, $f_7 = 2$, and $f_8 = 1$, then $f'_1 = f_1 = 128$, $f'_2 = f_2 = 64$, $(128/2 \leqslant f'_3 < 128)$, $f'_3 = f_3 + f_4 = 32 + 16 = 48$ $(128/3 \leqslant f'_3 < 128/2)$, $f'_4 = f_5 + f_6 + f_7 + f_8 = 8 + 4 + 2 + 1 = 15$ $(f'_3 < 128/3)$, and $k = 4$.

Note, that, even if k is not constant, if for the distribution $p()$ we get that $k < u/c$ then we will have an improvement over the harmonic partition scheme of [1].

Consider the sequence $s_1 = 1, s_2 = f_2/f_1, \ldots, s_n = f_n/f_1$.

Theorem 10. *If the sum* $\sum_{j=1}^{n} s_j$ *converges* (*as* $n \to \infty$) *to a number*, ε, *independent of* n *then* k *is a constant independent of* n.

Proof. Clearly,

$$\sum_{i=1}^{k} f'_i = \sum_{i=1}^{n} f_i.$$

Thus,

$$f_1(1+G(k)) \geqslant \sum_{i=1}^{n} f_i \Rightarrow 1+G(k) \geqslant \sum_{i=1}^{n} s_i$$
$$\Rightarrow (\text{as } n \to \infty) \lim_{n\to\infty} G(k) \geqslant \varepsilon,$$

i.e. the minimum k satisfies

$$k \geqslant G^{-1}(\varepsilon),$$

Conclusion 11 (Characterisation).
If $p()$ is such that $\lim_{n\to\infty} \sum_{j=1}^{n} s_j$ is a real number, then $p()$ belongs to the class $\mathbb{D}$ of general k-skewed distributions.

3.2. The online movie-scheduling algorithm S

We now present an online movie-scheduling algorithm S for the Video-on-Demand problem. We assume that S is aware of the statistics f_i. Then S uses this information to divide the set C of channels into classes $C_1, C_2, \ldots, C_k$ so that channels in partition C_i, $i \in \{1,2,\ldots,k\}$ will be used for movies to be seen by at least $h(i)$ users on the average. Here $h()$ is an increasing function from $\{1,2,\ldots,k\}$ to $\{1,2,\ldots,u/c\}$ and k is determined as in Section 3.1. The goal is to allocate channels in such a way that the channel revenue is optimized and unused channels are reduced. The sets C_i may change dynamically, i.e. each channel may belong to different classes at different time period of the execution.

The scheduling algorithm S employs m queues Q_j, $j=1,2,\ldots,m$ one for each movie e_j, $j=1,2,\ldots,m$. Initially, Q_j is empty for all j. When a request for movie e_j is made it is inserted into Q_j. the purpose of Q_j is to accumulate as many requests as possible for movie e_j, so that it can serve all of them with a single transmission at some time before any time limit expires.

Each Q_j has a "start time" $start_j$ which is the time when the earliest request for that movie arrived. At time $start_j + v$ the scheduler decides whether to serve the requests in Q_j. If there is a free channel in a set C_i with $h(i) \leqslant |Q_j|$, then all movie requests in Q_j are served on that channel by a single transmission. If, however, no such channel is available, then S rejects only those requests in Q_j made at time $start_j$ and resets $start_j$ to the time of the earlier request now in Q_j.

When a channel is freed it is chosen to be placed to a set C_i with probability f'_i/F where f'_i, F as in Section 3.1.

From the above $h(i)$ is clearly the number of requests that each transmission should, at least, serve, if it is using a channel in set C_i. Set then

$$h(1) = 1$$

and

$$h(l) = l\,\frac{u}{ck}, \quad l = 2,\ldots,k.$$

Definition 12. Let g_i be the expected number of channels currently in set C_i, $i = 1,2,\ldots,k$.

Remark. $g_i = c(f'_i/F)$, $i = 1,2,\ldots,k$ because of the way S places the free channels.

Note that any transmission made on a channel in set C_i must serve at least one request for $i = 1$ and at least $i(u/ck)$ requests for $i = 2,\ldots,k$.

4. The performance of *S*

In order to analyze the performance of S we follow steps similar to those of [1]. The *saturation level at instant t* is the highest i such that all channels in sets $C_1, C_2, \ldots, C_i$ are occupied at time t. The saturation level of an interval of time is the highest saturation level achieved during the interval.

We divide executions into intervals $I_0, I_1, \ldots$ of time T each, i.e., $I_j = [jT, (j+1)T)$.

For all j, let $A(j)$ be the number of requests accepted by S at interval I_j in response to a request sequence, and $R(j)$ the number of requests rejected by S but accepted by the offline algorithm OPT in its execution in response to the same request sequence. Let $\bar{A}(j)$, $\bar{R}(j)$ be the expected values of $A(j)$ and $R(j)$, respectively. Let σ_j be the saturation level of interval I_j.

Note that, without loss of generality, $\sigma_j \geqslant 2$, $\forall j$ since else scheduling is trivial. Thus, there is some $t \in I_j$ such that all channels in sets $C_1, C_2, \ldots, C_{\sigma_j}$ are occupied. Since such requests must have been scheduled to run no earlier than $t - T$ we have $\forall j$ (where $\bar{x}$ denotes the expected value of x, as well as $E(x)$)

$$\begin{aligned}
\bar{A}(j-1) + \bar{A}(j) &\geqslant h(1)g_1 + h(2)g_2 + \cdots + h(E(\sigma_j))g_{E(\sigma_j)} \\
&= \frac{c(h(1)f'_1 + h(2)f'_2 + \cdots + h(E(\sigma_j))f'_{E(\sigma_j)})}{F} \\
&= \frac{ch(1)f'_1}{F} + \sum_{i=2}^{E(\sigma_j)} \frac{ch(i)f'_i}{F} \\
&= \frac{cf_1}{F} + \frac{u}{kF} \sum_{i=2}^{E(\sigma_j)} i f_i \\
&\geqslant \frac{cf_1}{F} + \frac{u}{kF} \sum_{i=2}^{E(\sigma_j)} f_1 \quad \text{(by Rule 5)} \\
&\geqslant \frac{cf_1 + (u/k)f_1(\sigma_j - 1)}{F}.
\end{aligned}$$

But $F \leqslant f_1(1 + G(k))$ (Rule 6)

Thus,

$$\bar{A}(j-1)+\bar{A}(j) \geqslant \frac{cf_1+(u/k)f_1(E(\sigma_j)-1)}{f_1(1+G(k))} \geqslant \frac{u}{k}\,\frac{E(\sigma_j)-1}{(1+G(k))}. \quad (1)$$

Now, all requests in $R(j)$ were rejected by S during I_j, so each such request was made in the interval $[jT-v,(j+1)T-v)$. The offline algorithm OPT could thus serve these requests anytime in $[jT-v,(j+1)T-v)$. Since each channel is freed after T time units, OPT can utilize *each* channel twice in this interval.

To bound the number of requests (for the same movie) that such a transmission by OPT could serve we distinguish cases depending on the value of σ_j.

Suppose first that $\sigma_j < k$. Then, any offline transmission serving more than $h(\sigma_j)$ requests would also be served by S. Therefore, $\forall j$

$$R(j) \leqslant 2h(\sigma_j)c = 2\,\frac{u}{k}\,\sigma_j$$

because *OPT* cannot neither use more than c, nor utilize each channel for more than two transmissions, nor gain revenue more than $h(\sigma_j)$ for each transmission rejected by S (otherwise S would have accepted this transmission also, since there are channels available in C_{σ_j+1}).

But then,

$$\bar{R}(j) \leqslant 2\,\frac{u}{k}\,E(\sigma_j).$$

Let $\bar{A}=\sum \bar{A}(j)$ be the *expected* total number of requests accepted by S, and $\bar{R}=\sum \bar{R}(j)$ the *expected* total number of requests rejected by S but accepted by the optimal offline algorithm *OPT*.

Then

$$\delta(S) \leqslant \frac{\bar{A}+\bar{R}}{\bar{A}} = 1+\frac{\bar{R}}{\bar{A}}.$$

Thus,

$$\delta(S) \leqslant 1+\frac{2u/kE(\sigma_j)}{\frac{(u/k)(E(\sigma_j)-1)}{1+G(k)}} \quad \text{(by Eq. (1))}$$

i.e.,

$$\delta(S) \leqslant 1+2(1+G(k))\,\frac{E(\sigma_j)}{E(\sigma_j)-1}$$

and since $E(\sigma_j) \geqslant 2$ we get

$$\delta(S) \leqslant 1+2(1+G(k))$$

i.e.,

$$\delta(S) \leqslant 3+2G(k). \quad (2)$$

Consider now the case where $\sigma_j = k$. Note that

$$\bar{R}(j) \leqslant 2h(k)c = 2u$$

and

$$\bar{A}(j) \geqslant \frac{c+u}{1+G(k)}$$

and, for the whole problem to make sense, $c \ll u$.

Then again

$$\begin{aligned}\delta(S) &\leqslant 1 + 2(1+G(k))\frac{u}{c+u} \\ &\leqslant 1 + 2(1+G(k)).\end{aligned} \tag{3}$$

So we have proved

Theorem 13. *If $p()$ is general k-skewed then the competitive ratio of the scheduler S is bounded above by $3 + 2G(k)$.*

Corollary 14. *If $p()$ is strict k-skewed then $\delta(S) \leqslant 3 + 2H_{k-1}$, where $H_{k-1} = 1 + \frac{1}{2} + \cdots + 1/(k-1)$ is the $(k-1)$th harmonic number.*

Note that in both cases, $\delta(S)$ is constant.

5. An adaptive scheduler

In this section we present an *adaptive* online movie-scheduling scheme R which is not aware of the movie popularities, i.e., the distribution $p()$ in not known. R uses an initial partition of the channels in C into classes $C_1, C_2, \ldots, C_\lambda$, and a mechanism to reallocate channels dynamically to the sets C_i, and *adjusts* to the initially unknown distribution $p()$, thus achieving an asymptotic competitive ratio equal to $\delta(S)$.

The adaptive scheduler R partitions the channels into λ classes $C_1, \ldots, C_\lambda$ where $\lambda = \lceil u/c \rceil$ and $|C_i| = \lceil c/iH_i \rceil$, i.e., the initial allocation is the full Harmonic allocation.

R works initially as in [1] (i.e. with $h(i)=i$). However, it keeps m additional counters S_j, $j = 1, 2, \ldots, m$ one per movie. Each time movie e is requested, S_e is incremented by 1 (initially $S_j = 0$, $\forall j$).

Let e be a particular movie and $p(e)$ its probability of request.

After N total requests from the beginning, the number of requests, S_e, for e satisfies $\forall \beta \in (0,1)$

$$(1-\beta)Np(e) \leqslant S_e \leqslant (1+\beta)Np(e)$$

with probability at least

$$1 - \exp\left(-\frac{\beta^2}{2}Np(e)\right)$$

from the Chernoff bound for the Bernoulli of N trials and success probability $p(e)$.

From the frequency definition of probability we know that the ratio $S_e/\sum_e S_e \to p(e)$ as $N \to \infty$.

Now let $q_0 = \min_e\{1/p(e)\}$ and let $N \geqslant q_0 N' 2/\beta^2$ (N' arbitrary). Then S_e/N estimates $p(e)$ with prob $\geqslant 1 - \exp(-N')$, $\forall e$.

So, after $N_0 = q_0 N' 2/\beta^2$ total requests, we use the estimates of $p(e)$ and switch the scheduler to S.

For an unbounded sequence of requests, the initial segment of N_0 requests does not matter in the estimation of $\delta(R)$. Thus,

Corollary 15. *If $p()$ is general k-skewed, then the asymptotic competitive ratio of scheduler R is equal to $\delta(S)$.*

Acknowledgements

The authors wish to thank J. Garay for motivating discussions during ESA '95, and the anonymous referees for their constructive comments on previous versions of this work.

References

[1] S. Aggarwal, J.A. Garay, A. Herzberg, Adaptive video on demand, in: Proceedings of the Third Annual European Symposium on Algorithms (ESA'95), Lecture Notes in Computer Science, Vol. 959, Springer, Berlin, pp. 538–553.

[2] B. Awerbuch, Y. Azar, S. Plotkin, Throughput competitive on-line routing, in: Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS'93), 1993.

[3] B. Awerbuch, Y. Bartal, A. Fiat, A Rosén, Competitive non-preemptive call control, in: Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'94), 1994.

[4] B. Awerbuch, R. Gawlick, T. Leighton, Y. Rabani, On-line admission control and circuit routing for high performance computing and communication, in: Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'94), 1994, pp. 412–423.

[5] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, Cambridge, 1998.

[6] C. Bouras, V. Kapoulas, G. Pantziou P. Spirakis, Randomized adaptive video on demand, Proceedings of the 15th ACM-PODC, Philadelphia PA, USA, 1996 (short paper).

[7] C. Bouras, V. Kapoulas, G. Pantziou, P. Spirakis, Competitive video on demand schedulers for popular movies, Workshop on Algorithmic Aspects of Communications, July 11–12, Bologna, Italy, 1997.

[8] H.J. Chen, T.D.C. Little, Physical storage organizations for time-dependent multimedia data, in: Proceedings of the ACM FODO, 1993.

[9] S. Christodoulakis, C. Faloutsos, Design and performance considerations for an optical disk-based, multimedia object server, Computer 19 (1986) 45–56.

[10] J.A. Garay, I.S. Gopal, Call preemption in communication networks, in: Proceedings of the INFOCOM '92, Florence, Italy, 1992, pp. 1043–1050.

[11] J.A. Garay, I. Gopal, S. Kutten, Y. Mansour, M. Yung, Efficient on-line call control algorithms, in: Proceedings of the Second Israeli Symposium on Theory of Computing and Systems, June 1993, pp. 285–293.

[12] C.J. Horton, 110 channels without boundaries: why restrict options? Comm. Eng. Design 19 (1) (1993) 29–30.

[13] V. Kapoulas, P. Spirakis, Randomized competitive algorithms for admission control in general networks, in: Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing (PODC'95), August 1995 (short paper).
[14] J. Koegel, A. Syta, Routing of multimedia connections in hybrid networks, Proceedings of the SPIE—International Society on Optical Engineering, Vol. 1786, 1993, pp. 2–10.
[15] E. Koutsoupias, C. Papadimitriou, Beyond competitive analysis, in: Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '94), 1994, pp. 394–400.
[16] W.K.L. Lie, J.C.S. Lui, L. Golubchik, Threshold-based dynamic replication in large-scale video-on-demand systems, in: Proceedings of the Eighth International Workshop on Research Issues in Data Engineering, Orlando, 1998, pp. 52–58.
[17] S. Loeb, Delivering interactive multimedia documents over networks, IEEE Comm. Magazine 30 (1992) 52–59.
[18] S.M. McCarthy, Integrating Telco interoffice fiber transport with coaxial distribution, Proceedings of the SPIE—International Society on Optical Engineering, Vol. 1786, 1993, pp. 23–33.
[19] K. Metz, Next generation CATV networks, Proceedings of the SPIE—International Society on Optical Engineering, Vol. 1786, 1993, pp. 184–189.
[20] R. Ramarao, V. Ramamoorthy, Architectural design of on-demand video delivery systems: the spatio-temporal storage allocation problem, Proceedings of the ICC, 1991.
[21] P.V. Rangan, H.M. Vin, S. Ramanathan, Designing an on-demand multimedia service, IEEE Commun. Magazine 30 (1992) 56–64.
[22] C. Sell, Video on demand internal trial, Proceedings of the SPIE—International Society on Optical Engineering, Vol. 1786, 1993, pp. 168–175.
[23] H. Shachnai, P.S. Yu, On analytic modelling of multimedia batching schemes, Performance Evaluation 33 (1998) 201–213.
[24] W.D. Sincoskie, System architecture for a large scale video-on-demand service, Comput. Networks ISDN Systems 22 (1991) 155–162.
[25] J.L. Wolf, P.S. Yu, H. Shachnai, Disk load balancing for video-on-demand systems, ACM Multimedia Systems J. 5 (1997) 358–375.

For further reading

R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, Cambridge, 1995, pp. 318, 333.