The Data Broadcast Problem with Preemption

Nicolas Schabanel

LIP, ENS Lyon, 46, allée d'Italie, F-69364 Lyon Cedex 07, France

Abstract. The data-broadcast problem consists in finding an infinite schedule to broadcast a given set of messages so as to minimize the average response time to clients requesting messages, and the cost of the broadcast. This is an efficient means of disseminating data to clients, designed for environments, such as satellites, cable TV, mobile phones, where there is a much larger capacity from the information source to the clients than in the reverse direction.
Previous work concentrated on scheduling indivisible messages. Here, we studied a generalization of the model where the messages can be preempted. We show that this problem is *NP*-hard, even in the simple setting where the broadcast costs are zero, and give some practical 2-approximation algorithms for broadcasting messages. We also show that preemption can improve the quality of the broadcast by an arbitrary factor.

1 Introduction

1.1 Motivation

Data-broadcast is an efficient means of disseminating data to clients in wireless communication environment, where there is a much larger capacity from the information source to the recipients than in the reverse direction, such as happens when mobile clients (*e.g.* car navigation systems) retrieve information (*e.g.* traffic information) from base-station (*e.g.* the emitter) through a wireless medium. In a broadcasting protocol, items are broadcast according to an infinite horizon schedule and clients do not explicity send a request for an item to the server, but connect to the broadcast channels (shared by all the clients) and wait until the requested item is broadcast. These system are therefore known as *pseudo-interactive* or *push-based*: the server "pushes" the items, or messages, to the clients (even if disconnected) according to a schedule which is oblivious to the effective requests; as opposed to the "traditional" *pull-based* model, where the clients send a request to "pull" the required item from the server when they need it. The quality of the broadcast schedule is measured by the expected service time of the addressed requests. Furthermore, as each message has a cost for broadcasting (*e.g.* a weather broadcast and a news broadcast may have different costs for the emitter), the server also tries to minimize the resulting cost of service. The server has then to minimize the expected service response time of the requests (quality of service) and the broadcast cost of the resulting schedule (cost of service). The server designs the broadcast schedule from the *profile* of the users: given the messages $M_1, \dots, M_m$, the profile consists of the *popularities* of the different messages, that is to say the probabilities $(p_i)_{1 \leqslant i \leqslant m}$, that Message M_i is requested by a random user. [17] proposes some techniques to gauge user profiles in push-based environment.

With the impressive growth of the wireless, satellite and cable network, the data dissemination protocols have a number applications in research and commercial frameworks. One of the earliest applications was the Boston Community Information System (BCIS, 1982) developed at the MIT to deliver news and information to clients equipped personally with radio receivers in metropolitan Boston. It was also introduced in early 1980's in the context of Teletext and Videotex [8, 3]. It is now used by applications that require dissemination among a huge number of clients. The Advanced Traffic Information System (ATIS) [14], which provides traffic and route planning information to cars specially equipped with computers, may have to serve over 100,000 clients in a large metropolitan city during the rush hours. The news delivery systems on the Internet, such as Pointcast inc. (1997), or Airmedia inc. (1997), require efficient information dissemination system. A comparison of the push-based system to the traditional pull-based approach for those problems can be found in [1].

Note that the data-broadcast problem also models the maintenance scheduling problem and the multi-item replenishment problem [5, 6, 10].

While previous work made the assumption that messages transmission cannot be preempted, we focus in this paper on the case where the messages do not have uniform transmission times and can be split.

1.2 Background

Since the early 1980's, many authors [8, 3–6, 11] have studied the data-broadcast problem in the restrictive setting where all messages have the same length, the broadcast is done on a single channel, and time is discrete (this restricted problem is also known as Broadcast disks problem or Dissemination-based systems). In particular, Ammar and Wong [3, 4] give an algebraic expression of the expected service time of periodic schedules, provide a lower bound, and prove the existence of an optimal schedule which is periodic. Our Lemmas 2, 3 and Proposition 1 are generalizations of these results to our setting. Bar-Noy, Bhatia, Naor and Schieber [6] prove that the problem with broadcast costs is *NP*-hard, and after a sequence of papers giving constant factor approximations [5, 6], Kenyon, Schabanel and Young [11] design a PTAS for the problem. The papers [2, 1, 9, 15, 12, 13] study related questions pertaining to prefetching, to caching and to indexing.

As can be seen from the example of broadcasting weather and news reports, in many applications, it does not make sense to assume that all messages have the same transmission time; thus a couple of recent papers have explored the case of non-uniform transmission times. In [16] Vaidya and Hameed report some experimental results for heuristics on one or two channels. In [10] Kenyon and Schabanel show that the case where the messages do not have the same transmission time, the data-broadcast problem is *NP*-hard, even if message have zero broadcast cost, and does not always admit an periodic optimal schedule. They show that the natural extension of the lower bound given in [3, 6] is arbitrarily far from the optimal when the messages have very different length. The main difficulty is due to the fact that, while a long message is being broadcast, all requests for shorter and more popular messages have to be put on hold. But in that case, it seems reasonable to allow a occasional interruption of a long "boring" message transmission so as to broadcast a short popular message. In other word, one should allow preemption. This is the main motivation to the preemptive model introduced and studied in this paper.

1.3 Our contribution

This paper introduces and studies the model where the messages to be broadcast have non uniform transmission time and where their transmission can be preempted. One of the most interesting contribution from the practical point of view is that our algorithms (Section 4) generate preemptive schedules whose costs can be arbitrarily smaller than the optimal cost of any non-preemptive schedule on some inputs (See Note 1 in Section 4). Thus there is an *infinite* gap between the preemptive and non-preemptive problem.

We adopt the following model. The input consists of m messages $M_1, \ldots, M_m$ and an user profile determined by the *probabilities* $(p_i)_{1 \leqslant i \leqslant m}$ that a user requests Message M_i ($p_1 + \cdots + p_m = 1$). Each message M_i, $i = 1..m$, is composed of ℓ_i *packets* with transmission time 1 and each broadcast of a packet costs $c_i \geqslant 0$. The packets of the messages are broadcast over W identical and synchronized broadcast channels split into *time slots* of length 1 (time slot t is the period of time $[t-1, t]$). Given a schedule S of the packets into the slots, over the W channels, a client requesting Message M_i, starts monitoring all the channels at some (continuous) point, downloads the different packets of M_i *one at a time* when they are broadcast on some channel, and is served as soon as it has downloaded all the ℓ_i packets of Message M_i. The order in which the client has received the packet of M_i is irrelevant, as in TCP/IP.
The problem is to design a sequence S to schedule the packets over time, so as to minimize the sum of the expected service time of Poisson requests and of the average broadcast cost, i.e. so as to minimize $\limsup_{T\to\infty} \big(\mathrm{EST}(S, [0, T]) + \mathrm{BC}(S, [0, T])\big)$; here, $\mathrm{EST}(S, [0, T])$ denotes the expected service time of a request which is generated at a random uniform (continuous) instant between 0 and T, requests Message M_i with probability p_i, and must wait until the ℓ_i packets of M_i have been broadcast and downloaded; and $\mathrm{BC}(S, [0, T])$ is the average broadcast cost of the packets whose broadcast starts between 0 and T. Note that this definition agrees with the one in the literature (*e.g.* [6]), in the uniform-length case where the messages are composed of a single packet.

The results presented in this paper are obtained thanks to the simple but crucial observation made in Lemma 1: for all i, *an optimal schedule broadcasts the packets of Message M_i in Round Robin order.* We can thus restrict our search to Round Robin schedules. From this observation, we get an tractable algebraic expression for the cost of such a schedule in Lemma 2, from which we derive the lower bound in Lemma 3. This lower bound is the key to the two main results of the papers: 1) the problem is strongly *NP*-hard, even if no broadcast cost are assumed, in Theorem 1 (note that the *NP*-hardness proof given in [6] for the uniform length case requires non-zero broadcast cost); 2) there exists polynomial algorithm which constructs a periodic schedule with cost at most twice the optimal, in Section 4.

The lower bound also reveals some important structural differences between our model and the previous models. First, surprisingly, as opposed to *all* the previous studies, the lower bound *cannot* be realized by *scheduling the packets regularly* but by gluing them together (see Lemma 3): from the individual point of view of a request for a given message, the message should not be preempted. This allows to derive some results from the non-preemptive case studied in [10]. But, whereas non-preemptive strategies cannot approach this lower bound, we obtain, all the same, efficient approximation scheme within a factor of 2 by broadcasting the packets of each message regularly. Second, although the lower bound specializes to the one designed in [6] when all messages are composed of a single packet, deriving the lower bound is no longer a straight forward relaxation on the constraints on the schedule and requires a finer study of the "ideal" schedules. Moreover, its objective function is no longer convex and its resolution (in particular the unicity of its solution) needs a careful adaptation, presented Section 4.5, of the methods introduced in [6, 10].

Note that our preemptive setting models also the case where users do request single messages but batches of messages. We can indeed consider the packets of a message as messages of a batch. The preemptive case studied here is the case where the batches are all disjoint. In that sense the paper is an extension of some results in [7].

1.4 The cost function

We are interested in minimizing the *cost* of the schedule S, which is a combination of two quantities on S. The first one, denoted by $\mathrm{EST}(S)$, is the *expected service time* of a random request (where the average is taken over the moments when requests occur, and the type M_i of message requested). If we define by $\mathrm{EST}(S,I)$, the expected service time of a random request arrived in time interval I, $\mathrm{EST}(S)$ is: $\mathrm{EST}(S) = \limsup_{T\to\infty} \mathrm{EST}(S,[T_0,T])$, for any T_0. If we denote by $\mathrm{ST}(S,M_i,t)$, the service time of a request for M_i arrived at time t, and by $\mathrm{EST}(S,M_i,I)$ the expected service time of a request for M_i arriving in time interval I, we get: $\mathrm{EST}(S,M_i,I) = \frac{1}{|I|}\int_I \mathrm{ST}(S,M_i,t)\ dt$, and $\mathrm{EST}(S,I) = \sum_{i=1}^{m} p_i\,\mathrm{EST}(S,M_i,I)$.

The second quantity is the *broadcast cost* $\mathrm{BC}(S)$ of the messages, defined as the asymptotic value of the broadcast cost $\mathrm{BC}(S,I)$ over a time interval I: $\mathrm{BC}(S) = \limsup_{T\to\infty} \mathrm{BC}(S,[T_0,T])$, for any T_0. By definition, each broadcast of a packet of M_i costs c_i. For a time interval I, $\mathrm{BC}(S,I)$ is the sum of the cost of all the packets whose broadcast begins in I, divided by the length of I. The quantity which we want to minimize is then: $\mathrm{COST}(S) = \mathrm{EST}(S) + \mathrm{BC}(S)$. Note that up to scaling the costs c_i, any linear combinaison of EST and BC can be considered.

2 Preliminary Results

2.1 Structural properties

The following lemma is a crucial observation that will allow to deal with the dependencies in a tractable way. From this observation, we derive an algebraic expression for the cost of periodic schedule. In the next section, we show that this expression yields to a lower bound on the cost of any schedule. The lower bound will be used in Section 4 to design efficient approximation algorithm.

Definition 1. *A schedule S is said* Round Robin *if at most one packet of each message M_i is broadcast in any time slot according to S, and if S schedules the packets of each message in Round Robin order (i.e. according to a cyclic order).*

Lemma 1 (Round Robin). *For any schedule S, consider the Round Robin schedule S' constructed from S by rescheduling in Round Robin order the packets of each message M_i within the slots reserved in S to broadcasting a packet of M_i. Then:* $\mathrm{COST}(S') \leqslant \mathrm{COST}(S)$.

Moreover, if S is periodic and is not Round Robin, then S' is periodic and: $\mathrm{COST}(S') < \mathrm{COST}(S)$.

Proof. First, S and S' have the same broadcast cost. Second, consider a request for M_i arriving at time t in S, and the ℓ_i first time slots where a packet of M_i is broadcast in S after time t. The service time of the request is minimized *iff* the ℓ_i packets of M_i are broadcast in those slots. Thus the expected service time in S' is at most as large as in S. Moreover, if S is periodic with period T and is not Round Robin, then S' is periodic with period $\leqslant T\prod_{i=1}^{m}\ell_i$ and its expected service time is smaller than S's. □

W.l.o.g. we will now only consider Round Robin schedules.

Lemma 2 (Cost). *Consider a periodic schedule S with period T. For each i, n_i is the number of broadcasts of message M_i in a period, and $(t_j^i)_{1\leqslant j\leqslant n_i\ell_i}$ the time elapsed between the beginnings of the j^{th} and the $(j+1)^{th}$ broadcasts of a packet of Message M_i. Then:*

$$\mathrm{EST}(S) = 1 + \sum_{i=1}^{m} p_i \sum_{j=1}^{n_i\ell_i} \frac{t_j^i}{T}\left\{\frac{t_j^i}{2} + \left(t_{j+1}^i + \cdots + t_{j+\ell_i-1}^i\right)\right\}$$

and $\mathrm{BC}(S) = \dfrac{1}{T}\displaystyle\sum_{i=1}^{m} c_i n_i \ell_i$, *where the indices are considered modulo $n_i\ell_i$.*

Proof. Consider i in $\{1, \ldots, m\}$. Message M_i is broadcast n_i times per period, its contribution to the broadcast cost is then $n_i \ell_i c_i / T$. A request is for Message M_i with probability p_i and arrives between the j^{th} and the $(j+1)^{\text{th}}$ broadcasts of a packet of M_i with probability t_j^i / T. It starts then downloading the first packet after $t_j^i/2$ time on expectation and ends downloading the last packet after $t_{j+1}^i + \cdots + t_{j+\ell_i-1}^i + 1$ other time slots.

Remark 1 (Trapezoids representation). Note that we can represent the cumulated response time to request for a given message over a period of time by the sum of the areas of trapezoids as shown Figure 1; the black arrows are two example of requests, their waits are highlight in black, and the extra cost for downloading the last packet is in grey.

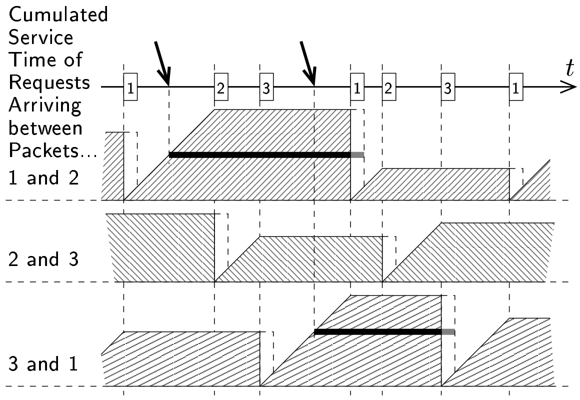


Fig. 1: The expected service time.

2.2 Optimality results

Theorem 1 (NP-Hardness). *Finding the optimal schedule is strongly NP-hard on a single channel and with zero cost messages.*

Proof sketch. (Omitted) The proof is derived from the *NP*-hardness proof of the non-preemptive case given in [10]: we show that deciding whether the lower bound in Lemma 3 is realized is at least as hard as N-partition. □

Remark 2. Note that [6] yields an other *NP*-hardness proof by stating that the uniform length case with non zero cost is already *NP*-hard; however the present proof does not use costs.

Proposition 1 (Optimal periodic). *There exists an optimal schedule which is periodic. It can be computed in exponential time.*

Proof sketch. (Omitted) The proof is based on the search of a minimum cost cycle in a finite graph, and the lemmas are broadly inspired from [4, 5, 10] but their proofs need to be widely adapted in order to take into account the segmentation of the messages into packets. □

3 A lower bound

Finding a good lower bound is a key point to designing and proving efficient approximation algorithms for this problem. An algorithm to compute the value of the following lower bound, will be given Section 4.5.

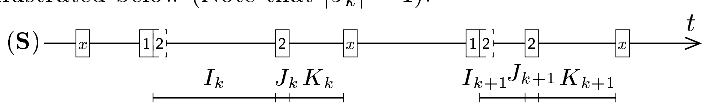
Lemma 3 (Lower bound). *The following minimization problem is a lower bound to the cost of any schedule of the packets of $M_1, \ldots, M_m$ on W channels:*

$$\text{LB}(M) \begin{cases} \min\limits_{\tau > 0} \sum\limits_{i=1}^{m} p_i \left(\dfrac{\tau_i \ell_i}{2} + \ell_i - \dfrac{\ell_i - 1}{2\tau_i} \right) + \dfrac{c_i}{\tau_i} \\ \text{Subject to:} \quad (i) \quad \forall i,\ \tau_i \geqslant 1 \quad \text{and} \quad (ii) \quad \sum\limits_{i=1}^{m} \dfrac{1}{\tau_i} \leqslant W \end{cases}$$

This minimization problem admits a unique solution τ^*. LB(M) *is realized if and only if one can broadcast all the packets of each M_i consecutively periodically exactly every $(\tau_i^* \cdot \ell_i)$.*

Proof sketch. According to Lemma 1, let S be a periodic Round Robin schedule of the packets of messages $M_1, \ldots, M_m$ on W channels with period T. We use the same notations (n_i) and (t_j^i) as in Lemma 2. Given that Message M_i is broadcast n_i times per period, we seek for the optimal value of the (t_j^i) for each message independently. We relax the constraints on the schedule by authorizing messages to overlap and to be scheduled outside the slots. The proof works in three steps:
1. If the expected service time for M_i with $\ell_i \geqslant 2$ is minimized, then for any pair of consecutive broadcasts of the same packet of M_i at time t_1 and t_2 $(t_1 < t_2)$, a packet of M_i is broadcast at time $(t_1 + 1)$ or $(t_2 - 1)$.
2. If the expected service time of M_i is minimized, the packets of M_i are broadcast within blocks of ℓ_i consecutive time slots.
3. The blocks are optimally scheduled periodically every T/n_i.

Step 1. Consider M_i with $\ell_i \geqslant 2$ and two consecutive packets of M_i (w.l.o.g. packets 1 and 2). For $1 \leqslant k \leqslant n_i$, let I_k, J_k, and K_k be the intervals delimited by the end of the k^{th} broadcast of packet 1, the beginning and the end of the k^{th} broadcast of packet 2, and the beginning of the next broadcast of a packet of M_i as illustrated below (Note that $|J_k| = 1$).



Let S' be the schedule that schedules the packets of M_i as in S except that packet 2 is always scheduled next to packet 1. A request for M_i that raises outside intervals I_k, J_k and K_k has the same service time in S and in S'. A request that raises in I_k is served one time unit later in S' than in S. But a request that raises in $J_k \cup K_k$ is served $|I_{k+1}|$ earlier in S' than in S. The expected service time varies then from S to S' by:

$$\sum_{i=1}^{n_i} (|I_k| \times 1 - (1 + |K_k|) \times |I_{k+1}|) = -\sum_{i=1}^{n_i} |K_k| \cdot |I_{k+1}| \leqslant 0$$

Thus, the expected service time in S' is at most as big as in S and smaller if there exists in S a pair of consecutive broadcasts of packet 2 occuring at time t_1 and t_2 $(t_1 < t_2)$ so that no packet of M_i is broadcast at time $(t_1 + 1)$ $(|K_k| \neq 0)$ and $(t_2 - 1)$ $(|I_{k+1}| \neq 0)$.

Step 2 is obtained by contradiction using the transformation in Step 1.

Step 3. We are thus left with n_i blocks of ℓ_i packets of M_i. Let t_k be the time elapsed between the beginning of the k^{th} and the $(k+1)^{\text{th}}$ block. Lemma 2 yields that the expected service time for M_i is:

$$\sum_{k=1}^{n_i} \left(\frac{{t_k}^2}{2T}\right) + \ell_i - \frac{n_i \ell_i (\ell_i - 1)}{2T}$$

which is minimized under the constraint $\sum_{k=1}^{n_i} t_k = T$, when for all k, $t_k = T/n_i$. Define $\tau_i = T/(n_i \ell_i)$. The cost of S is thus bounded from below by:

$$\mathrm{COST}(S) \geqslant \sum_{k=1}^{n_i} \left\{ p_i \left(\frac{\tau_i \ell_i}{2} + \ell_i - \frac{\ell_i - 1}{2\tau_i} \right) + \frac{c_i}{\tau_i} \right\}$$

Finally $n_i \ell_i \leqslant T$ and $\sum_{i=1}^m n_i \ell_i \leqslant WT$ imply: (i) $\tau_i \geqslant 1$ and (ii) $\sum_{i=1}^m 1/\tau_i \leqslant W$. Minimizing over those constraints yields the lower bound on the cost of any schedule. The unicity of the solution τ^* to the minimization problem will be proved Section 4.5.
Moreover by construction, the lower bound is realized *iff* there exists a periodic Round Robin schedule that broadcast the ℓ_i packets of each M_i in consecutive slots, periodically every $\tau_i^* \ell_i$. □

Remark 3. One can derive a trivial lower bound close up to an additive term $\sum_{i=1}^m p_i \ell_i$ to ours by simply optimizing the time needed to download a given packet for each message. If this later lower bound is sufficient to analyze our heurisitics, it is never realized and cannot be used to yield our *NP*-hardness result.

4 Constant factor approximation algorithms

Note 1. The optimal ficticious schedule suggested by the lower bound $\mathrm{LB}(M)$ is not realizable in general. Actually, as shown in [10], if no preemption are used, the optimal cost of a schedule can be arbitrary far from the lower bound $\mathrm{LB}(M)$. Consider the problem of scheduling $W+1$ messages $M_1, \ldots, M_{W+1}$ on W channels, where M_i counts $\ell_i = L^{i-1}$ packets, cost $c_i = 0$, and request probability $p_i = \alpha / L^{i-1}$, where α is such that $p_1 + \cdots + p_m = 1$. In that case, one can show by induction on W that when L goes to infinity, the optimal schedule without preemption has a cost $\mathrm{OPT}_{\text{whithout preemption}} = \Theta(L^{1/2^W})$, but $\mathrm{LB}(M) = \Theta(1)$.

In order to minimize the cost of the schedule, we won't follow exactly the ficticious schedule suggested by the lower bound in Lemma 3. In fact, remark that if we spread regularly the packets of each message M_i, every τ_i, in this ficticious schedule, the expected service time to a random request increases by less than a factor of 2. This will be helpful in order to design an efficient approximation algorithm for the preemptive

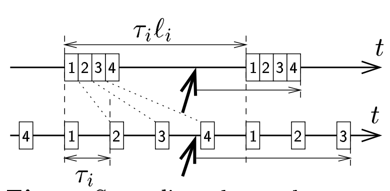


Fig. 2: Spreading the packets regularly.

Algorithm 1 Randomized algorithm

Input: Some positive numbers $\tau_1, \ldots, \tau_m$, verifying $\sum_{i=1}^m 1/\tau_i \leqslant 1$.
Let $\tau_0 > 0$ so that:
$$1/\tau_0 = 1 - \sum_{i=1}^m 1/\tau_i$$
Output:
for $t = 1..\infty$ **do**
Draw $i \in \{0, 1, \ldots, m\}$ with probability $1/\tau_i$. Schedule during slot t, the next packet of Message M_i in the Round Robin order, if $i \geqslant 1$; and Idle during slot t, otherwise.

Algorithm 2 Greedy algorithm

Input: Some positive numbers $\tau_1, \ldots, \tau_m$, verifying $\sum_{i=1}^m 1/\tau_i \leqslant 1$.
Let $c_0 = p_0 = 0$ and $\tau_0 > 0$ so that:
$$1/\tau_0 = 1 - \sum_{i=1}^m 1/\tau_i$$
Output:
for $t = 1..\infty$ **do**
Select $i \in \{0, 1, \ldots, m\}$ which minimizes $(c_i - p_i \tau_i \sum_{j=1}^{\ell_i} s_{i,j}^{t-1})$.
Schedule during slot t, the next packet of M_i in the Round Robin order, if $i \geqslant 1$; and Idle during slot t, otherwise.

case.

We will first present algorithms that construct efficient schedules on a single channel in Sections 4.1, 4.2 and 4.3; then Section 4.4 shows how to extend these algorithms to the multichannel case, using a result of [6].

4.1 A randomized algorithm

Theorem 2. *Given m messages $M_1, \ldots, M_m$, the expected cost of the one-channel schedule S generated by the randomized algorithm 1, is:*
$$\mathbb{E}[\mathrm{COST}(S)] = \frac{1}{2} + \sum_{i=1}^{m} \left(p_i \tau_i \ell_i + \frac{c_i}{\tau_i} \right)$$
Thus if $\tau = \tau^*$ realizes $\mathrm{LB}(M)$*:* $\quad \mathbb{E}[\mathrm{COST}(S)] \leqslant 2 \cdot \mathrm{LB}(M) - 3/2$.

Proof. A packet of M_i is broadcast with probability $1/\tau_i$ in S. The expected frequency of M_i is then $1/\tau_i$ and $\mathbb{E}[\mathrm{BC}(S)] = \sum_{i=1}^m c_i/\tau_i$. A request for M_i is served after ℓ_i downloads of a packet of M_i: it waits on expectation $1/2$ until the end of the current time-slot and $\tau_i \ell_i$ upto the end of the download of the last packet of M_i. Then, $\mathbb{E}[\mathrm{EST}(S)] = 1/2 + \sum_{i=1}^m p_i \ell_i \tau_i$.
Finally $\tau_i^* \geqslant 1$ and $\ell_i \geqslant 1$ imply: $2\,\mathrm{LB}(M) \geqslant \sum_{i=1}^m (p_i \tau_i^* \ell_i + c_i/\tau_i^*) + 2$, which yields the last statement.

4.2 A greedy approximation

We present in this section a derandomized version of the randomized algorithm above.

As shown Figure 3, we define the *state* of the schedule at time slot t as a vector s^t, such that: for any i and $1 \leqslant j \leqslant \ell_i$, the j^{th} of the ℓ_i last broadcasts of a packet of M_i before time t starts at

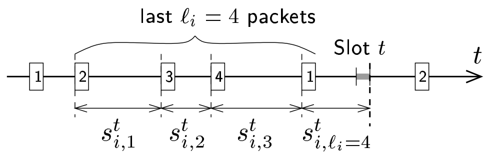


Fig. 3: The state $(s_{i,j}^t)$ at time t.

time $(t - (s^t_{i,j} + \cdots + s_{i,\ell_i})$. Since no request arrive before $t = 0$, we equivalently assume that all the packets of all messages are fictively broadcast at time $t = 0$, and initially, at time $t = 0$: for all i and j, $s^0_{i,j} = 0$;.

Theorem 3. *Given m messages $M_1, \ldots, M_m$, the cost of the one-channel schedule S generated by the greedy algorithm 2, is:*

$$\mathrm{COST}(S) \leqslant \frac{1}{2} + \sum_{i=1}^{m} \left(p_i \tau_i \ell_i + \frac{c_i}{\tau_i} \right)$$

Thus if $\tau = \tau^*$ realizes $\mathrm{LB}(M)$*:* $\quad \mathrm{COST}(S) \leqslant 2 \cdot \mathrm{LB}(M) - 3/2$.

Proof sketch. (Omitted) The greedy algorithm is a derandomized version of the algorithm above. The greedy choice ensures that at any time t, the choice made in time slot t minimizes the expected cost of the already allocated slots $1, \ldots, t-1$, if the schedule would continue with the randomized scheme. Its cost is then, at any time, bounded from above by the expected cost of the randomized schedule. □

4.3 A deterministic periodic approximation

It is sometimes required to have a fixed schedule instead of generating it on the fly. For instance, it helps to design caching strategies [1]. The next result shows that one can construct an efficient periodic schedule with polynomial period. Note that this allow also to *guarantee* a bound (the period) on the service time of any request.

Theorem 4. *One can construct in polynomial time, a periodic schedule with cost $\leqslant 2 \cdot \mathrm{LB}(M)$ and period polynomial in the total length and cost of the messages $(\frac{14}{3}(\sum_{i=1}^{m} \ell_i)^2 + 2\sum_{i=1}^{m} c_i \ell_i)$.*

Proof sketch. (Omitted) The schedule is constructed as shown Figure 4: 1) First, schedule all the packets of each message during the first $\mathcal{L} =_{\mathrm{def}} \sum_i \ell_i$ time slots; 2) Second, executes T steps of the greedy algorithm above; 3) Third, sort the set $X = \{k\tau^*_i : 1 \leqslant k \leqslant \ell_i\}$ in increasing order and schedule during the next $\mathcal{L}$ time slots, the k^{th} packets of the messages M_i in order of increasing $k\tau^*_i$; 4) Finally, complete with some packets of the messages in order to ensure that for all i, the number of broadcasts of a packet of M_i in a period is a multiple of ℓ_i, and thus guaranty the Round Robin property. One can show that the cost of the resulting schedule is at most $2\,\mathrm{LB}(M)$ as soon as the period is bigger than $\frac{14}{3}(\sum_i \ell_i)^2 + 2\sum_i c_i \ell_i$. □

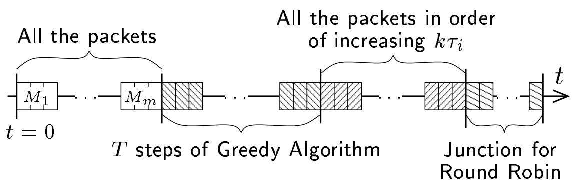


Fig. 4: A periodic approximation.

4.4 Multi-channel 2-approximations

The performance ratio proof for the randomized algorithm given above only rely on the fact that we know how to broadcast the packets of each M_i every τ_i on expectation. In order to extend the result to the multi-channel case, we only need to manage to broadcast the packets of each M_i with probability $1/\tau_i$, while ensuring that two packets of the same messages are not broadcast during the same time slot. A straight forward application of the method designed in [6], to extend the single channel randomized algorithm to the multi-channel, yields then the result.

The multi-channel greedy algorithm is again obtained by derandomizing the schedule, and by extending the greedy choice as in [6]. Finally the extension of the periodic approximation is then constructed exactly as in Section 4.3, except that one uses the multi-channel greedy algorithm instead of the single channel one.

4.5 Solving the lower bound

The aim of this last section is to solve the following generic non-linear program (A), defined by:

$$(A) \quad \min_{\tau>0} \sum_{i=1}^{m} a_i\tau_i + \frac{b_i}{\tau_i} \quad \text{Subject to: } (i)\ \forall i,\ \tau_i \geqslant 1 \text{ and } (ii)\ \sum_{i=1}^{m} \frac{1}{\tau_i} \leqslant W$$

where W is a positive integer, $a_1, \dots, a_m$ are positive numbers, and $b_1, \dots, b_m$ are *arbitrary* numbers.

We present essentially an extension of the method designed in [6] for the special case where for all i, $b_i \geqslant 0$. The results presented are basically the same but the proofs need to be adapted. As in [6], we introduce a relaxed minimization problem (A'), which do not require the constraint (i), and which can be solved algebraically. The solution to the relaxed problem will allow to construct and prove the unicity of the solution to (A).

Lemma 4 (Relaxation). *Given some positive numbers $a_1, \dots, a_m$, a positive integer W and some numbers $b_1, \dots, b_m$, the following minimization problem:*

$$(A') \quad \min_{\tau>0} \sum_{i=1}^{m} a_i\tau_i + \frac{b_i}{\tau_i} \quad \text{Subject to:} \sum_{i=1}^{m} \frac{1}{\tau_i} \leqslant W$$

admits a unique solution τ' verifying: $\tau'_i = \sqrt{(b_i + \lambda')/a_i}$, for a certain $\lambda' \geqslant 0$. If, for all i, $b_i \geqslant 0$ and $\sum_i \sqrt{a_i/b_i} \leqslant W$, then $\lambda' = 0$; else λ' is the unique solution to: $\sum_i \sqrt{a_i/(b_i + \lambda')} = W$.

Proof sketch. (Omitted) Solved by carefull use of Lagrangian relaxation. □

Lemma 5. *Consider the two non-linear minimization problems (A) and (A'), a solution τ^* to (A) and the solution τ' to (A'). Then, for all i, if $\tau'_i < 1$, then $\tau^*_i = 1$.*

Proof. The proof given in [6] is only based on the unimodularity (and not on the convexity) of the terms $a_i\tau_i + b_i/\tau_i$. Their proof then naturally extends to the case where some b_i may be negative. □

Corollary 1 (Unicity). *The minimization problem (A) admits a unique solution τ^* which can be computed in polynomial time.*

Proof. Consider a solution τ^* to (A). We compute the solution τ' to (A'). If for some i_0, $\tau'_{i_0} < 1$, then $\tau^*_{i_0} = 1$. Thus, we remove this variable from Problem (A) by fixing its value to 1, and iterate. If for all i, $\tau'_i \geqslant 1$, τ' is also solution of (A), which is thus unique: $\tau^* = \tau'$. □

Acknowledgment. We'd like to thank Neal E. Young and Claire Kenyon, for useful comments and careful reading of the paper.

The full version of the paper is available at //www.ens-lyon.fr/∼nschaban.

References

1. Acharya, S. *Broadcast Disks: Dissemination-based Management for Assymmetric Communication Environments.* PhD thesis, Brown University, 1998.
2. Ammar, M. H. Response time in a teletext system: An individual user's perspective. *IEEE Transactions on Communications COM-35,11* (Nov. 1987), 1159–1170.
3. Ammar, M. H., and Wong, J. W. The design of teletext broadcast cycles. In *Performance Evaluation* (1985), vol. 5(4), pp. 235–242.
4. Ammar, M. H., and Wong, J. W. On the optimality of cyclic transmission in teletext systems. In *IEEE Trans. on Comm.* (1987), vol. COM-35(11), pp. 1159–1170.
5. Anily, S., Glass, C. A., and Hassin, R. The scheduling of maintenance service. To appear (http://www.math.tau.ac.il/∼hassin).
6. Bar-Noy, A., Bhatia, R., Naor, J. S., and Schieber, B. Minimizing service and operation costs of periodic scheduling. In *Proc. of the 9th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)* (1998), pp. 11–20.
7. Bar-Noy, A., and Shilo, Y. Optimal broadcasting of two files over an asymmetric channel. In *Proc. of Infocom* (1999).
8. Gecsei, J. The architecture of videotex systems. *Prentice Hall, Englewood Cliffs, N.J.* (1983).
9. Imielinski, T., Viswanathan, S., and Badrinath, B. Energy efficient indexing on air. In *SIGMOD* (May 1994).
10. Kenyon, C., and Schabanel, N. The data broadcast problem with non-uniform transimission times. In *Proc. of the 10th SODA* (1999), pp. 547–556.
11. Kenyon, C., Schabanel, N., and Young, N. E. Polynomial time approximation scheme for data broadcast. Submitted, Oct 1999.
12. Khanna, S., and Liberatore, V. On broadcast disk paging. In *Proceedings of 30th STOC* (1998), vol. 30, pp. 634–643.
13. Khanna, S., and Zhou, S. On indexed data broadcast. In *Proceedings of 30th STOC* (1998), vol. 30, pp. 463–472.
14. Shekhar, S., and Liu, D. Genesis: An approach to data dissemination in Advanced Traveller Information Systems (ATIS). In *IEEE Data Engineering Bulletin* (Sept. 1996), vol. 19(3).
15. Tan, K., and Xu, J. Energy efficient filtering of nonuniform broadcast. In *Proc. of the 16th Int. Conf. in Distributed Computing System* (1996), pp. 520–527.
16. Vaidya, N., and Hameed, S. Log time algorithms for scheduling single and multiple channel data broadcast. In *Proc. of the 3rd ACM/IEEE Conf. on Mobile Computing and Networking (MOBICOM)* (Sep. 1997).
17. Vishwanath, S. *Publishing in wireless and wireline environments.* PhD thesis, Rutgers University, 1994.