Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 5668

The Data Broadcast Problem with Preemption

Nicolas Schabanel

Septembre 1999

Research Report N° RR1999-49

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

The Data Broadcast Problem with Preemption

Nicolas Schabanel

Septembre 1999

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.

Keywords: Data broadcast, broadcast disk, push technology, preemption, approximation algorithm, wireless networks

Résumé

La diffusion de données est une réponse apportée au problème de la surcharge des réseaux et serveurs d'information (de type info-trafic, info-bourse, info-sport), où le nombre de clients est très élevés et la plus part d'entre eux demande majoritairement le même petit nombre de messages ; et plus généralement adaptées aux réseaux asymétriques où la bande passante des serveurs vers les clients est bien plus grande que des clients vers les serveurs (les réseaux satellites, téléphones mobiles/base,...). L'idée est de réserver des canaux (typiquement hertzien) pour la diffusion de ce petit nombre de messages qui seront diffusés indépendamment des requêtes effectives : un client intéressé par un de ces messages, se connecte sur ces canaux et attend que le message demandé soit diffusé. Le problème de la diffusion de données *[Data broadcast problem]* consiste à trouver un ordonnancement qui diffuse les messages, en minimisant l'attente moyenne des utilisateurs (défini par un profil-utilisateur : les probabilités de demande pour chaque message), et le coût de la diffusion des messages.
Alors que les travaux précédents étudient l'ordonnancement de messages indivisibles, nous étudions ici une généralisation du modèle au cas où la diffusion des messages peut être interrompue. Nous démontrons que le problème est *NP*-dur, même quand les coûts de diffusion sont nuls, et proposons un algorithme qui génère un algorithme de coût inférieur au double de l'optimal. Nous montrons aussi que l'ajout de préemption permet d'améliorer d'un facteur arbitraire la qualité d'un ordonnancement de messages de temps de transmission non-uniforme.

Mots-clés: Diffusion de données, broadcast disk, push technology, préemption, algorithme d'approximation, réseaux sans fil

The Data Broadcast Problem with Preemption

Nicolas Schabanel

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

September 1999

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.* route planning computers in cars) 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 average response time to 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 average response time to 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. [19] 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 [9, 3]. It is designed for applications that requires dissemination among a huge number of clients: the Advanced Traffic Information System (ATIS) [16], 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; and 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, 12]. In the maintenance scheduling problem, one try to minimize the risk of failing of a given set of machines, by maintaining the different machines. The risk of failing for a given machine increases linearly with time elapsed since the last maintenance. In this framework, a schedule determines the date of maintenance of the machines, and tries to minimizes the risk of failure and the cost of the maintenance. The multi-item replenishment problem consists in, given a set of items, determining when the item stock should be replenished, given the shipping cost and the rate at which each item is consumed.

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 [9, 3, 4, 11, 8, 5, 6, 13] 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. In particular, Ammar and Wong [3, 4] give an algebraic expression of the average response time of periodic schedules, prove a lower bound, and prove the existence of an optimal schedule which is periodic. Our Lemmas 2, 5 and Proposition 4 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 [13] design a PTAS for the problem. The papers [2, 1, 10, 17, 14, 15] 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. Vaidya and Hameed [18] report some experimental results for heuristics on one or two channels. Kenyon and Schabanel [12] show that, when the messages do not have the same transmission time, the data-broadcast problem is *NP*-hard, even if the messages have zero broadcast cost, and it does not always admit an periodic optimal schedule; they also show that the natural extension of the lower bound given in [3, 6] is arbitrarily far from the optimal, and design a constant factor randomized approximation. 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 an 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 we show that preemption helps: indeed, there are input messages for which the lower bound in the non-preemptive setting [12] can be arbitrarily larger than the cost of the preemptive schedule produced by our algorithm (See Remark 4 in Section 4), and thus there is an infinite gap between the preemptive and non-preemptive problem.

We adopt the following model. Time is divided into *slots*: given an integer t, time slot t is the period of time between time $t-1$ and t. Given W broadcast channels, the input consists of m messages $M_1, \ldots, M_m$ and a user profile determined by the *probabilities* $(p_i)_{1 \leqslant i \leqslant m}$ that a user requests Message M_i. Each message M_i, $i = 1..m$, is composed of ℓ_i *packets* with transmission time 1 and with *broadcast cost* $c_i \geqslant 0$ each. Given a schedule S of the packets of the messages into the slots, over the W channels, a client requesting for 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 average response time to Poisson requests and of the average broadcast cost, i.e. so as to minimize $\limsup_{T \to \infty} \big(\mathrm{ART}(S, [0,T]) + \mathrm{BC}(S, [0,T])\big)$; here, $\mathrm{ART}(S, [0,T])$ denotes the average response time to 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 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 schedules which verify this Round Robin property. 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 5. 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 3 (note that the *NP*-hardness proof given in [6] requires non-zero broadcast cost); 2) there exists polynomial algorithms which constructs a schedule with cost at most twice the optimal, in Section 5.

Our lower bound also reveals some important structural differences between our model and the previous models. First, surprisingly, as opposed to *all* the previous studies, our lower bound *cannot* be realized by *scheduling the packets regularly* but by gluing them together (see Lemma 5): from the individual point of view of a request for a given message, the message should not be preempted paradoxically. This allows to derive some results from the non-preemptive case studied in [12]. But, whereas a non-preemptive strategy 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 more a straight forward relaxation on the constraints on the schedule and requires the use of Lagrangian relaxation. Moreover, its objective function is no longer convex and its resolution (in particular the unicity of its solution) needs a careful adaptation of the methods introduced in [6, 12], presented in Section 5.5.

Note that our preemptive setting models also the behavior of a client browsing among web pages (the packets) which are linked one to the others in cliques (the messages). Minimizing the average response time in our setting, is also minimizing the average time needed to browse a set of the web pages linked together given that some prefetching/caching strategies are used by the browser. In that sense, our paper is also a study of broadcasting given a certain kind of dependencies between messages; in this sense, it is a generalization of some of the results in [7].

2 The preemptive case, definition and notation

The input consists of m messages $(M_i)_{1\leqslant i\leqslant m}$, composed of ℓ_i packets of length 1. Message M_i is requested with probability $p_i > 0$, such that $p_1 + \cdots + p_m = 1$. Its packets have a *broadcast cost* $c_i \geqslant 0$.

A *schedule* S of messages $M_1, \ldots, M_m$ on W channels is formally a sequence $(s_t)_{t\geqslant 0}$ of W-tuples: $s_{t,w} = (i,j)$ means that the j^{th} packet of Message M_i is broadcast during time slot t, between time $t-1$ and t, on channel w. During the broadcast of a given schedule S, *requests* occur at any (continuous) time t, start monitoring all the channels, download the packets of the requested message M_i, *one at a time*, when they are broadcast on some channel, and are served as soon as all the ℓ_i packets of M_i have been downloaded. The order in which packets are downloaded is irrelevant, as in TCP/IP.

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 $\mathrm{ART}(S)$, is the *average response time* to 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 $\mathrm{ART}(S, I)$, the average response time to a random request arrived in time interval I, $\mathrm{ART}(S)$ is:

$$\mathrm{ART}(S) = \limsup_{T\to\infty} \mathrm{ART}(S, [T_0, T]), \quad \text{for any } T_0$$

If we denote by $\mathrm{RT}(S, M_i, t)$, the response time to a request for M_i arrived at time t, and by $\mathrm{ART}(S, M_i, I)$ the average response time to a request for M_i arriving in time interval I, we get:

$$\mathrm{ART}(S, M_i, I) = \frac{1}{|I|}\int_I \mathrm{RT}(S, M_i, t)\ dt \quad \text{and} \quad \mathrm{ART}(S, I) = \sum_{i=1}^{m} p_i\, \mathrm{ART}(S, M_i, I)$$

The second quantity is the *average 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:

$$\mathrm{BC}(S) = \limsup_{T\to\infty} \mathrm{BC}(S, [T_0, T]), \quad \text{for any } T_0$$

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:

$$\mathrm{COST}(S) = \mathrm{ART}(S) + \mathrm{BC}(S)$$

Note that up to scaling the costs c_i, any linear combinaison of ART and BC can be considered.

3 Preliminary Results

3.1 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 5 to design efficient approximation algorithms.

Lemma 1 (Round Robin) *Given a schedule S of the packets of the messages $M_1, \dots, M_m$, there exists a schedule S', in which the packets of each message are broadcast in Round Robin order, and so that:*

$$\mathrm{COST}(S') \leqslant \mathrm{COST}(S)$$

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

Proof. Consider a request for M_i in S, and the ℓ_i next time slots in S, where a packet of M_i is broadcast. The response time to the request is minimized if the ℓ_i packets of M_i are broadcast is these ℓ_i slots. Thus, re-scheduling the packets of each message in Round Robin order, minimizes the average response time of a given schedule, while it does not change its broadcast cost. Furthermore, if S is periodic with period T, and does not verify this Round Robin property, the resulting schedule S' is periodic with period $\leqslant T \prod_i \ell_i$ and has a cost strictly smaller than S. □

W.l.o.g. we will now consider only schedules which broadcast the packets of each messages in Round Robin order. We will *w.l.o.g.* also assume that at most one packet of each message is broadcast over all the channels during each time slot.

Lemma 2 (Cost) *Consider a periodic schedule S of the packets of $M_1, \dots, M_m$ 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 from the beginning of the j^{th} broadcast of a packet of Message M_i to the beginning of the $(j+1)^{th}$.*

Then:

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

$$\text{and} \quad \mathrm{BC}(S) = \frac{1}{T}\sum_{i=1}^{m} c_i n_i \ell_i$$

where the indices are considered modulo $n_i\ell_i$.

Proof. Consider i in $\{1,\dots,m\}$. Message M_i is broadcast n_i times per period, its contribution to the average broadcast cost is then $n_i\ell_i c_i/T$.

A request is for Message M_i with probability p_i and is then served once it has downloaded all the packets of M_i. Consider the $n_i\ell_i$ intervals delimited by the date of the beginning of the $n_i\ell_i$ broadcasts of a packet of Message M_i; the request arrives during the j^{th} interval with probability t_j^i/T. It starts downloading the first packet after $t_j^i/2$ time on average and then ends downloading the last packet after $t_{j+1}^i + \cdots + t_{j+\ell_i-1}^i + 1$ other time slots. Summing over all the intervals yields that it is served on average after $\sum_{j=1}^{n_i\ell_i} \frac{t_j^i}{T}\left(\frac{t_j^i}{2} + t_{j+1}^i + \cdots + t_{j+\ell_i-1}^i + 1\right)$. $\square$

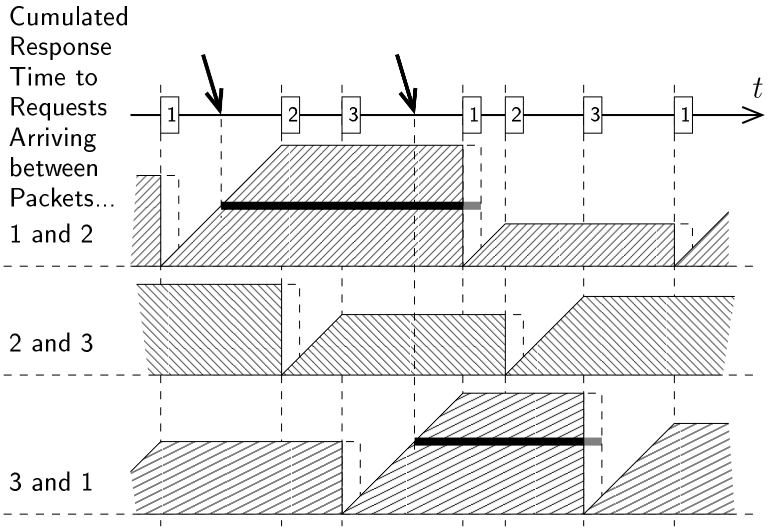

Figure 1: The average response time.

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.

Remark 2 (Time reverse) An algebraic manipulation shows that if $\hat{S}$ is the mirrored schedule of S ($\hat{S}$ schedules the same packets, on the same channels, as S, but in reverse order), then $\mathrm{COST}(\hat{S}) = \mathrm{COST}(S)$. Then, the average response time of a schedule can also be defined as the average response time to a request which waits back in time for the messages.

3.2 Optimality results

Theorem 3 (NP-Hardness) *Finding the optimal schedule of the packets of messages $M_1, \dots, M_m$ is strongly NP-hard on a single channel and with zero cost messages.*

Proof sketch. (Deferred to appendix) The proof is a reduction from N-partition, derived from the *NP*-hardness proof of the non-preemptive case given in [12]. □

Remark 3 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 the costs.

Proposition 4 (Optimal periodic) *There exists an optimal schedule which is periodic. It can be computed in finite time.*

Proof sketch. (Deferred to appendix) 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, 12] but their proofs need to be widely adapted in order to take into account the segmentation of the messages into packets. □

4 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 lower bound, will be given Section 5.5.

Lemma 5 (Lower bound) *The following minimization problem is a lower bound to the cost of any schedule of the packets of $M_1, \dots, M_m$ on W channels:*

$$\mathrm{LB}(M) \begin{cases} \displaystyle\min_{\tau>0} \sum_{i=1}^{m} p_i \left(\frac{\tau_i \ell_i}{2} + \ell_i - \frac{\ell_i - 1}{2\tau_i} \right) + \frac{c_i}{\tau_i} \\ \text{Subject to:} \qquad (i) \quad \forall i,\ \tau_i \geqslant 1 \\ \qquad\qquad\qquad\quad (ii) \quad \displaystyle\sum_{i=1}^{m} \frac{1}{\tau_i} \leqslant W \end{cases}$$

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

Proof. According to Proposition 4, let S be an optimal periodic schedule of the packets of messages $M_1, \dots, M_m$ 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). We relax the constraints on the schedule by authorizing messages to overlap and to be scheduled outside the slots.

The remaining constraints on the (t_j^i) are then that $g(t^i) =_{\mathrm{def}} \sum_{j=1}^{n_i \ell_i} t_j^i = T$ and $t_j^i \geqslant 1$. We denote by $f(t^i) = 1 + \frac{1}{T} \sum_{j=1}^{m} t_j^i \left(t_j^i/2 + t_{j+1}^i + \cdots + t_{j+\ell_i-1}^i \right)$; we have:

$$\mathrm{ART}(S, M_i) \geqslant \min_{g(t^i)=T} f(t^i)$$

In this paragraph we show that f is minimized *iff* Message M_i is scheduled without preemption (possibly on different channels along the time), periodically every T/n_i as in the op part of Figure 2. We use Lagrangian relaxation to minimize $f(t)$ under the constraint $g(t) = T$: according to classical minimization techniques, the minimum is reached at a point t^* where the gradients of f and g are collinear. We introduce a Lagrangian parameter λ; optimal t^* and λ^* verify then two equations:

$$\mathrm{grad} f(t^*) = \lambda^* \ \mathrm{grad} g(t^*) \quad \text{and} \quad g(t^*) = 0$$

(t^*, λ^*) is thus a critical point[1] of the function $F(t, \lambda) = f(t) - \lambda g(t)$. $\dfrac{\partial F}{\partial t_j}(t^*, \lambda^*) = 0 = \dfrac{\partial F}{\partial \lambda}(t^*, \lambda^*)$ yields that:

$$\text{for all } j, \quad \sum_{k=j}^{j+\ell_i-1} t_k^* = -\lambda^* T = T/n_i$$

Thus (t_j^*) is periodic with period ℓ_i. Hence $f(t^*)$ can be rewritten:

$$f(t^*) = 1 - \frac{n_i}{2T} \sum_{j=1}^{\ell_i} (t_j^*)^2 + \frac{T}{n_i} + 1$$

It is then minimized when the sum of the squares is maximized, when for some j, $t_j^* = T/n_i - (\ell_i - 1)$ and $t_{j+1}^* = \cdots = t_{j+\ell_i-1}^* = 1$. Then the minimum value of f is:

$$\frac{T}{2n_i} + \ell_i - \frac{n_i \ell_i}{T} \frac{\ell_i - 1}{2}$$

which is realized *iff* Message M_i is scheduled without preemption, periodically, every T/n_i.

Define $\tau_i = T/(n_i \ell_i)$, the average response time of S is bounded by:

$$\mathrm{ART}(S, M_i) \geqslant \frac{\tau_i \ell_i}{2} + \ell_i - \frac{\ell_i - 1}{2}$$

and the broadcast cost is:

$$\mathrm{BC}(S) = \sum_{i=1}^{m} \frac{c_i}{\tau_i}$$

Finally, the τ_i must verify: (i) $\tau_i \geqslant 1$, since $t_j^i \geqslant 1$; and (ii) $\sum_{i=1}^m 1/\tau_i \leqslant W$, since at most W messages are scheduled during the same time slot in S. Minimizing over the τ_i given the constraints (i) and (ii), yields to the expected lower bound on the cost of S.

Section 5.5 proves the unicity of the solution τ^* to the minimization problem $\mathrm{LB}(M)$; note that the objective is no longer convex as in [6, 12] and, in particular, unicity is not straight forward.

Finally, according to the proof we give above, the lower bound is realized if and only if one can schedule all the packets of each message M_i consecutively (possibly on different channels along the time) periodically exactly every $\tau_i^* \ell_i$. □

Remark 4 The optimal ficticious schedule suggested by the lower bound $\mathrm{LB}(M)$ is not realizable in general. Actually, as shown in [12], if no preemption are used, the optimal cost of a schedule can be arbitrary far from the lower bound $\mathrm{LB}(M)$.

[1]Recall that a *critical point* is a point $x^\#$ such that for all i, $\frac{\partial F}{\partial x_i}(x^\#) = 0$.

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 $\mathrm{LB}(M) = \Theta(1)$.

5 Constant factor approximation algorithms

In this section, we will design heuristics that achieve the lower bound $\mathrm{LB}(M)$ within a factor of 2. According to the remark above, preempting allows then to improve considerably the broadcast of messages with non-uniform transmission time.

In order to minimize the cost of the schedule, we won't follow exactly the ficticious schedule suggested by the lower bound in Lemma 5. In fact, remark that if we spread regularly the packets of each message M_i, every τ_i, in this ficticious schedule, the average response 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 case.

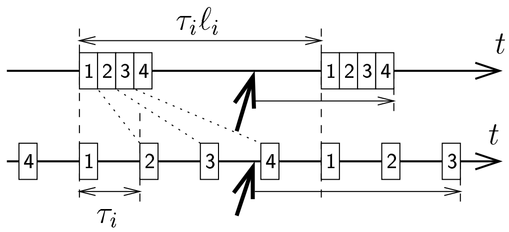

Figure 2: Spreading the packets regularly.

We will first present algorithms that construct efficient schedules on a *single channel* in Sections 5.1, 5.2 and 5.3; then Section 5.4 shows how to extend these algorithms to the multichannel case, using a result of [6].

5.1 A randomized algorithm

Theorem 6 *Given m messages $M_1, \ldots, M_m$, the expected cost of the one-channel schedule S generated by the randomized algorithm 1 (page 10), is:*

$$\mathbb{E}[\mathrm{COST}(S)] = \frac{3}{2} + \sum_{i=1}^{m} \left(p_i \tau_i \ell_i + \frac{c_i}{\tau_i} \right)$$

Thus if $\tau = \tau^*$ realizes $\mathrm{LB}(M)$: $\quad \mathbb{E}[\mathrm{COST}(S)] \leqslant 2 \cdot \mathrm{LB}(M) - 1/2 \leqslant 2 \cdot \mathrm{OPT} - 1/2.$

Proof. The expected cost of S is the sum of its expected average response time, and its expected broadcast cost: $\mathbb{E}[\mathrm{COST}(S)] = \mathbb{E}[\mathrm{ART}(S)] + \mathbb{E}[\mathrm{BC}(S)]$.
The broadcast cost is the weighted sum of the density of each message in the schedule, its expected value is the weighted sum of the expected densities. A packet of M_i is broadcast during any time

Algorithm 1 Randomized algorithm

Input: Some positive numbers $\tau_1, \dots, \tau_m$, verifying $\sum_{i=1}^m 1/\tau_i \leqslant 1$.

if $\sum_{i=1}^m 1/\tau_i < 1$ **then**

. Add a dummy message M_0, with $\ell_0 = 1$, $p_0 = c_0 = 0$, and τ_0 so that $1/\tau_0 = 1 - \sum_{i=1}^m 1/\tau_i$.

Output:

for $t = 1..\infty$ **do**

. Draw $i \in \{0, 1, \dots, m\}$ 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 $\tau_1, \dots, \tau_m$, verifying $\sum_{i=1}^m 1/\tau_i \leqslant 1$.

. Add a dummy message M_0, if needed.

Output:

for $t = 1..\infty$ **do**

. Select $i \in \{0, 1, \dots, m\}$ 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.

slot t with probability $1/\tau_i$, the expected density of M_i is then $1/\tau_i$ and:

$$\mathbb{E}[\mathrm{BC}(S)] = \sum_{i=1}^m c_i/\tau_i$$

Again, the average response time is the weighted sum of the average response time to a request for a particular message. Consider a request for Message M_i arriving during time slot t. It then has to wait on average $1/2$ until the end of slot t. Let $X_i(t)$ the random time from the end of slot t to the beginning of the next broadcast of a packet of M_i. $X_i(t)$ is independent from t, and its expected value is τ_i. The ℓ_i packets of M_i are then downloaded on expectation after $(\ell_i \tau_i) + 1$ time. The expected average response time of S is then:

$$\frac{3}{2} + \sum_{i=1}^m p_i \tau_i \ell_i$$

Note that, $\tau_i^* \geqslant 1$ and $\ell_i \geqslant 1$ yield:

$$2\,\mathrm{LB}(M) \geqslant \sum_{i=1}^m \left(p_i \tau_i^* \ell_i + \frac{c_i}{\tau_i^*} \right) + 2$$

Thus, $\mathbb{E}[\mathrm{COST}(S)] \leqslant 2\,\mathrm{LB}(M) - 1/2$. □

Remark 5 One can show that if τ^* is computed within ε accuracy, the expected cost of the resulting schedule is bounded by $2\,\mathrm{LB}(M) + (\ell_{\max} + c_{\max})\varepsilon - 1/2$, where $x_{\max}$ denotes $\max_i x_i$.

5.2 A greedy approximation

The randomized algorithm above can be derandomized as follow.

As shown Figure 3, we define the *state* of the schedule at time slot t by a vector $\vec{s}^t$, such that: for any i and $j = 1..\ell_i$, the j^{th} of the last ℓ_i broadcasts of a packet of M_i before time t starts at time $(t - (s_{i,j}^t + \cdots + s_{i,\ell_i}))$. Since no request arrives before $t = 0$, we equivalently assume that all the packets of all messages are fictively broadcast at time $t = 0$: initially, at time $t = 0$, for all i and j, $s_{i,j}^0 = 0$;.

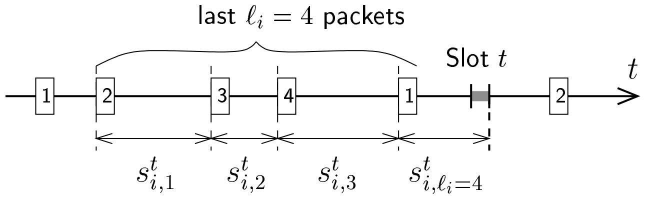

Figure 3: The state $(s^t_{i,j})$ at time t.

Theorem 7 *Given m messages $M_1, \dots, M_m$, the expected 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 $\mathrm{LB}(M)$*:* $\quad \mathrm{COST}(S) \leqslant 2 \cdot \mathrm{LB}(M) - 3/2 \leqslant 2 \cdot \mathrm{OPT} - 3/2.$

Proof sketch. (Deferred to appendix) 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 $1, \dots, t-1$, 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. A careful analysis allows to show that we can save a 1. □

5.3 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 *guaranty* a bound (the period) on the response time to any request.

Theorem 8 *One can construct in polynomial time, a periodic schedule with cost $\leqslant 2 \cdot \mathrm{LB}(M)$ and with a polynomial period in the total length and in the cost of the messages ($6(\sum_i \ell_i)^2 + 2 \sum_i c_i \ell_i$).*

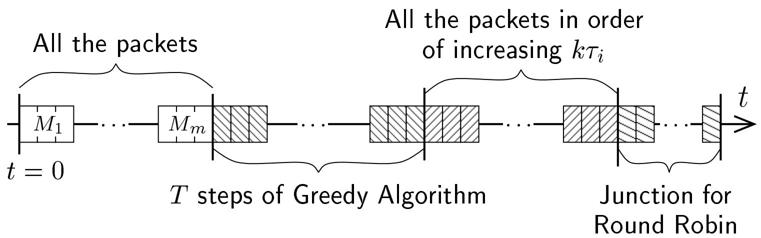

Figure 4: A periodic approximation.

Proof sketch. (Deferred to appendix) The schedule is constructed in four steps, 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, basically, $T \geqslant 6(\sum_i \ell_i)^2 + 2\sum_i c_i\ell_i$. $\square$

5.4 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 5.3, excepted that one uses the multi-channel greedy algorithm instead of the single channel one, during Step 2.

5.5 Solving the lower bound

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

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

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

We present essentially an extension of the method designed by [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 that can be solved algebraically. [6] shows that in the case of non-negative (b_i), if, in the solution τ' to the relaxed problem, for some i_0, $\tau'_{i_0} < 1$, then, in the solution τ^* of the original minimization problem (A), $\tau^*_{i_0} = 1$. Then they can remove this variable from the problem and iterate. We show here that this result also holds if the signs of the (b_i) are arbitrary.

The relaxed minimization problem is solved as follow:

Lemma 9 (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:*

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

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. Let $f(\tau) =_{\text{def}} \sum_i (a_i \tau_i + b_i/\tau_i)$ be the objective function. Note that the objective function is no longer convex as in [6, 12], the unicity of the minimum is then no more straight forward. First, note that $\lim_{||\tau|| \to \infty} f(\tau) = \infty$, thus the continuous function f admits a minimum on the domain defined by the constraints. Two cases need to be considered:

1. *$b_1, \dots, b_m$ are non-negative.* Recall the method presented in [6, 12]: f is a convex unimodular function, then it admits a unique minimum τ' on the closed convex domain defined by the constraint. It admits also a unique minimum $\tau^\#$ over $]0, \infty[^m$, defined by: $\tau^\#_i = \sqrt{b_i/a_i}$. Two cases occur:

   1.a) $\tau^\#$ verifies the constraint and it is the required minimum.

   1.b) $\tau^\#$ does not verify the constraint and thus the unique minimum verifying the constraint, τ' belongs to the boundary of the domain: $\sum_i 1/\tau'_i = W$. This later case is then solved by introducing a Lagrangian parameter $\lambda \geqslant 0$, and yields that: $\tau'_i = \sqrt{(b_i + \lambda)/a_i}$, where λ is the unique solution to $\sum_i 1/\tau'_i = W$.

2. *There exists i_0, such that $b_{i_0} < 0$.* Let τ' be *a* minimum of f on the domain. If $\sum_i 1/\tau'_i < W$, then decreasing τ'_{i_0} would improve the value of the minimum. Thus τ' verifies $\sum_i 1/\tau'_i = W$. We derive the expression of τ'_i, as well as its unicity, by the same technique used in case 1.b) above.

□

Lemma 10 *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. (Deferred to Appendix) 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 11 (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/or careful reading of the paper.

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] Chan, M. Y., and Chin, F. Schedulers for larger classes of pinwheel instances. *Algorithmica 9* (Jun. 1993), 425–462.

[9] Gecsei, J. The architecture of videotex systems. *Prentice Hall, Englewood Cliffs, N.J.* (1983).

[10] Imielinski, T., Viswanathan, S., and Badrinath, B. Energy efficient indexing on air. In *SIGMOD* (May 1994).

[11] Itai, A., and Rosberg, S. A golden ratio control policy for a multiple-access channel. *IEEE Transactions on Autom. Contr. AC-29* (Aug. 1984).

[12] Kenyon, C., and Schabanel, N. The data broadcast problem with non-uniform transimission times. In *Proc. of the 10th SODA* (1999), pp. 547–556.

[13] Kenyon, C., Schabanel, N., and Young, N. E. A PTAS for the data-broadcast problem. In preparation.

[14] Khanna, S., and Liberatore, V. On broadcast disk paging. In *P*roceedings of 30th STOC (1998), vol. 30, pp. 634–643.

[15] Khanna, S., and Zhou, S. On indexed data broadcast. In *P*roceedings of 30th STOC (1998), vol. 30, pp. 463–472.

[16] Shekhar, S., and Liu, D. Genesis: An approach to data dissemination in Advanced Traveller Information Systems (ATIS). In *I*EEE Data Engineering Bulletin (Sept. 1996), vol. 19(3).

[17] Tan, K., and Xu, J. Energy efficient filtering of nonuniform broadcast. In *P*roc. of the 16^{th} Int. Conf. in Distributed Computing System (1996), pp. 520–527.

[18] Vaidya, N., and Hameed, S. Log time algorithms for scheduling single and multiple channel data broadcast. In *P*roc. of the 3rd ACM/IEEE Conf. on Mobile Computing and Networking (MOBICOM) (Sep. 1997).

[19] Vishwanath, S. *P*ublishing in wireless and wireline environments. PhD thesis, Rutgers University, 1994.

[20] Young, N. E., Tarjan, R. E., and Orlin, J. B. Faster parametric shortest path and minimum balances algorithms. In *N*etworks (Mar. 1991), vol. 21(2).

6 Deferred proofs

6.1 *NP*-hardness

Proof of Theorem 3. As in [12], we will reduce N-Partition to the Data-Broadcast problem with preemption and without costs as follows. Recall that N-Partition takes as input m integers $x_1, \dots, x_m$ and must decide whether there exists a partition of $\{1, \dots, m\}$ in N disjoint sets $I_1, \dots, I_N$ such that $\sum_{i \in I_1} x_i = \cdots = \sum_{i \in I_N} x_i$. Let $\mathcal{S} =_{\text{def}} x_1 + \cdots + x_m$.

We consider the following one-channel Data-Broadcast instance: $m+1$ messages $M_0, M_1, \dots, M_m$ such that:

$$\begin{cases} M_0: \quad \ell_0 = \mathcal{S}/N, \quad \text{and} \quad p_0 = \alpha \Big/ \left(\dfrac{5\mathcal{S}}{N} - 1\right) \quad (c_0 = 0) \\ M_{i \geqslant 1}: \quad \ell_i = x_i \quad \text{and} \quad p_i = \alpha \Big/ \left(\dfrac{2\mathcal{S}^2}{x_i} + x_i - 1\right) \quad (c_i = 0) \end{cases}$$

where α is chosen so that: $p_0 + p_1 + \cdots + p_m = 1$.

In that case, one can prove that the lower bound LB(M), defined in Lemma 5, is obtained with $\tau_0^* = 2$ and $\tau_i^* = \frac{2\mathcal{S}}{x_i}$. (Note that the value of the Lagrangian parameter λ is $\alpha \geqslant 0$). Let us denote by A the (ugly) corresponding value of LB(M). Note that, the p_i, ℓ_i and A are all rational numbers computed in polynomial time. We consider the decision problem "Does there exist a schedule of the packets of $M_0, \dots, M_m$ with cost less than or equal to A?".

Note that this reduction is clearly polynomial. According to Lemmas 5, the answer to the question is "yes" if and only if there exists a periodic schedule which broadcasts all the packets of M_i, consecutively, periodically, every $\tau_i^* x_i$. Such a schedule must then be of the following form: between two consecutive broadcasts of M_0, a sequence of messages of total length at most $\mathcal{S}/N$ can be broadcast. Then the answer is "yes" if and only if the set of messages $M_1, \ldots, M_m$ can be split into N sets $\mathcal{M}_1, \ldots, \mathcal{M}_N$ such that $\sum_{M_i \in \mathcal{M}_1} \ell_i = \cdots = \sum_{M_i \in \mathcal{M}_N} \ell_i$. □

Remark 6 This proof can easily be generalized to prove the *NP*-hardness of the problem "Does there exists a schedule of m messages, with rational lengths and probabilities, and zero broadcast costs, on W channels whose cost is less or equal than A?". Taking the reduction above, it suffices to add $W-1$ messages $M_{m+1}, \ldots, M_{m+W-1}$ of lengths $\ell_{i>m} = \mathcal{S} + N\ell_0$ and probabilities adjusted so that $\tau^*_{i>m} = 1$.

6.2 An optimal schedule which is periodic

Lemma 12 (Density of periodic) *Given $\varepsilon > 0$ and a schedule S of the packets of $M_1, \ldots, M_m$ on W channels, there exists a periodic schedule S' of $M_1, \ldots, M_m$ with cost:*

$$\mathrm{COST}(S') \leqslant \mathrm{COST}(S) + \varepsilon$$

Proof. By definition of the cost of S, let T such that, for all $t \geqslant T$: $\mathrm{COST}(S, [0,t]) \leqslant \mathrm{COST}(S) + \varepsilon/2$. Let $\mathcal{L} =_{\mathrm{def}} \ell_1 + \cdots + \ell_m$. Let S_t be the periodic schedule with period $t + \mathcal{L}$, identical to S up to time t and broadcasting between time $t+1$ and $t+\mathcal{L}$ on the different channels, each packet of the messages in order of first appearance in S after time t. By construction, the requests arriving in $[0,t]$ are served in S_t at most as late as in S. Furthermore, the requests arriving in $[t, t+\mathcal{L}]$ are served after at most $\mathcal{L} + \mathrm{RT}(S,0)$ time slots on average. Thus: $\mathrm{COST}(S_t) = \frac{t}{t+\mathcal{L}} \mathrm{COST}(S_t, [0,t]) + \frac{\mathcal{L}}{t+\mathcal{L}} \mathrm{COST}(S_t, [t, t+\mathcal{L}]) \leqslant \mathrm{COST}(S, [0,t]) + \varepsilon/2 + \frac{\mathcal{L}(\mathcal{L}+\mathrm{RT}(S,0))}{t+\mathcal{L}}$. Thus for t sufficiently large, we have the expected result. □

Note 1 *There exists an optimal schedule.* Consider a sequence of periodic schedules (S_i) approaching the optimal cost: for all i, $\mathrm{COST}(S_i) \leqslant \mathrm{OPT} + 1/2^i$. We construct an optimal schedule S iteratively. Suppose that we have constructed S upto time slot t at step i, so that $\mathrm{COST}(S, [0,t]) \leqslant \mathrm{OPT} + 1/2^{i-1}$, we concatenate to S sufficiently many periods of S_{i+1} (let u be the new end) so that, basically, the cost over $[0,u]$ is bounded by: $\mathrm{COST}(S, [0,u]) \leqslant \mathrm{OPT} + 1/2^{(i+1)-1}$. The resulting schedule S is clearly optimal.

Lemma 13 (Bounded interval) *Consider a periodic schedule S of the packets of $M_1, \ldots, M_m$ on W channels, there exists a periodic schedule S' of $M_1, \ldots, M_m$, with cost $\mathrm{COST}(S') \leqslant \mathrm{COST}(S)$, so that, for all i, any interval in which no packet of M_i is broadcast, has length at most K_i, with:*

$$K_i = \frac{1}{p_i}\left((2\mathcal{L}+1)^2 + c_i\right)$$

where $\mathcal{L} =_{\mathrm{def}} \ell_1 + \cdots \ell_m$.

Proof. Consider i in S, so that there exists an interval I of length $\geqslant K_i$ in which no packet of M_i is broadcast:

1. *There is no empty slot in S during the interval I.* Since $K_i \geqslant 2\mathcal{L}+1$, there exists j, so that $2\ell_j+1$ packets of Message M_j are broadcast in an interval J of length $2\mathcal{L}+1$, included in I. Consider S', the periodic schedule identical to S, except that the (ℓ_j+1) packets of M_j in J is replaced by a packet of M_i (in order to maintain the Round Robin property in S', the period of S' is multiplied by $\ell_i\ell_j$ and the substitution is applied in parallel in each of the $\ell_i\ell_j$ copies of the pattern).
   If $t_1, \dots, t_{2\ell_j}$ denote the lengths of the intervals between the beginnings of the k^{th} and the $(k+1)^{\text{th}}$ broadcasts of a packet of M_j in J, one can show that the variation of the cumulated average response time to request for M_j is: $\displaystyle\sum_{k=1}^{\ell_j} t_k \cdot t_{k+\ell_j} - 1$. The Cauchy-Schwarz inequality implies that it is $\leqslant \sqrt{\sum_{k=1}^{\ell_j} t_k^2}\sqrt{\sum_{k=\ell_j+1}^{2\ell_j} t_k^2} \leqslant (2\mathcal{L}+1)^2$, since the length of the interval J is bounded by $2\mathcal{L}+1$. If u_k denote as well the lengths of the intervals between the k^{th} and $(k+1)^{\text{th}}$ broadcasts of a packet of M_i among the $2\ell_i$ packet broadcasts around the insertion point of the new packet in S', the variation of the cumulated average response time to request for M_i is: $\displaystyle -\sum_{k=1}^{\ell_j} u_k \cdot u_{k+\ell_j} + 1 \leqslant -u_{\ell_i+1} - u_{\ell_i} \leqslant -K_i$ by construction. The variation of the cumulated cost from S to S' is then bounded by: $p_j(2\mathcal{L}+1)^2 - c_j - p_iK_i + c_i \leqslant 0$ by definition of K_i. Thus, $\mathrm{COST}(S') \leqslant \mathrm{COST}(S)$.

2. *If there is an empty slot in S during I.* S' is constructed from S by broadcasting a packet of M_i during this empty slot (As above, the period is multiplied by ℓ_i in order to maintain the Round Robin property). The proof given above yields again that the cost of S' is less than the cost of S.

The proof is completed by iterating this scheme on all the problematic intervals. The period of the resulting schedule S' is then bounded by $\prod_i \ell_i$ times the period of S. $\square$

Proof of Proposition 4. Thanks to Lemma 12, it suffices to prove that there exists an optimal schedule among the periodic schedule, it will be optimal among all the schedule. Furthermore, Lemma 13 shows that we can restrict our search to schedules which, for all i, broadcasts a packet of M_i at least every K_i time slots. The existence proof relies then on a graph construction which is inspired from [4, 5, 12]. We give the construction for the single channel case, it naturally extends to the multichannel case.

We define the following finite graph G with cost. The vertices of G are the m-tuples $\langle \vec{s}_1, \dots, \vec{s}_m \rangle$, where $\vec{s}_i$ is a ℓ_i-vector with values in $\{1, \dots, K_i\}$. $s_{i,j}$, with $1 \leqslant j \leqslant \ell_i$, denotes the j^{th} coordinate of $\vec{s}_i$. $\sigma(\vec{s}_i)$ denotes the *left shift* operation:

$$\sigma(\vec{s}_i)_{\ell_i} = 0, \text{ and } \sigma(\vec{s}_i)_j = s_{i,j+1}, \text{ for } j < \ell_i$$

and $\alpha(\vec{s}_i)$ increases the last coordinate:

$$\alpha(\vec{s}_i)_{\ell_i} = s_{i,\ell_i} + 1, \text{ and } \alpha(\vec{s}_i)_j = s_{i,j}, \text{ for } 1 \leqslant j < \ell_i$$

There is an edge e_{M_i} from any m-tuple $\langle \vec{s}_1, \ldots, \vec{s}_i, \ldots, \vec{s}_m \rangle$ to every $\langle \alpha(\vec{s}_1), \ldots, \alpha(\sigma(\vec{s}_i)), \ldots, \alpha(\vec{s}_m) \rangle$, which is labeled "$M_i$". It is given a cost:

$$\mathrm{cost}(e_{M_i}) =_{\mathrm{def}} c_i + 1 - p_i s_{i,1} + \sum_{j=1}^{m} p_j (1/2 + s_{j,1} + \cdots + s_{j,\ell_i})$$

There is also an edge e_X from any m-tuple $\langle \vec{s}_1, \ldots, \vec{s}_m \rangle$ to $\langle \alpha(\vec{s}_1), \ldots, \alpha(\vec{s}_m) \rangle$, which is labeled "X". It is given a cost:

$$\mathrm{cost}(e_X) =_{\mathrm{def}} 1 + \sum_{j=1}^{m} p_j (1/2 + s_{j,1} + \cdots + s_{j,\ell_j})$$

Semantically, being at Node (or State) $\langle \vec{s}_1, \ldots, \vec{s}_m \rangle$ at time t, means that for all i, the j^{th} of the last ℓ_i broadcasts of a packet of M_i occurred at time $\big(t - (s_{i,j} + \cdots + s_{i,\ell_i})\big)$ as shown Figure 3. Following the edge e with label "M_i" between time $t-1$ and t, means that a packet of M_i is scheduled during time slot t. Following an edge e with label "X" between time $t-1$ and t means that no packet is broadcast during time slot t. The cost of edge e is the extra-cost induced by the broadcast during time slot t: graphically, it is the weighted sum of the sections of the areas of the trapezoids below the time slot t, Figure 1; thanks to Remark 2 it can also be interpreted as the average response to a request arriving during time slot t and waiting for the messages back in time.

More precisely, there is a natural surjection from the cycles of G passing through a vertex $\langle \vec{s}_1, \ldots, \vec{s}_i \rangle$, with $s_{1,\ell_1} = 1$, to the finite-cost periodic schedules where, for any i, any interval where no packet of M_i is broadcast as length at most K_i. To each cycle γ going through a vertex $\langle \vec{s}_1, \ldots, \vec{s}_i \rangle$, with $s_{1,\ell_1} = 1$ we associate the periodic schedule $S(\gamma)$ which is the concatenation of the labels of the edges of γ. With our definition of the cost of the edges, according to Lemma 2:

$$\mathrm{COST}(S(\gamma)) = \frac{1}{|\gamma|} \sum_{e \in \gamma} \mathrm{cost}(e) =_{\mathrm{def}} \mathrm{COST}_G(\gamma)$$

Since, G is finite and the cost are positive, there exists an cycle γ^* with minimum cost. The corresponding schedule $S(\gamma^*)$ is thus optimal among the schedules of the packets of $M_1, \ldots, M_m$. It can also be computed in finite time [20]. □

6.3 Performance ratio of the greedy algorithm 2

Proof of Theorem 7. Algorithm 2 is a derandomized version of Algorithm 1: we will prove that the greedy choice for every slot t in Algorithm 2 is the one that minimizes the expected cost of the schedule given the choices made in slots 1 to $(t-1)$; this ensures that the cost of the greedy schedule is at most the expected cost of the randomized one. In order to get the result, we define the *cost of a slot* t, $\mathrm{cost}(t)$, as in the proof of Proposition 4 above, as the sum of the cost of the broadcast during time slot $(t+1)$ and of the cumulated wait of the requests that have not been served at time t. Given the state $(s^t_{i,j})$ at time t, $p_i s^t_{i,j}$ requests for M_i arrive between the j^{th} and the $(j+1)^{\text{th}}$ of the last ℓ_i broadcasts of a packets of M_i and are not yet served at time t, for $j < \ell_i$;

and $p_i(s^t_{i,\ell_i} - 1/2)$ requests raise on expectation between the last broadcast of a packet of M_i and time t which are not yet served at time t; we also need to add 1 more for downloading the last packet, thus:

$$\text{cost}(t) =_{\text{def}} c_{t+1} + \sum_{i=1}^{m} p_i \sum_{j=1}^{\ell_i} s^t_{i,j} + \frac{1}{2}$$

where c_{t+1} denotes the cost of the broadcast during time slot $(t+1)$. The average cost of a slot corresponds exactly to the average response time to a random Poisson request, since each request is counted as long as it has not been served. Remark also that the cost of a slot corresponds exactly: 1) to the section of the trapezoids defined Figure 1 below time slot t; 2) to the average response time to a random request raising uniformly during time slot t. Then, Remarks 1 and 2 also yields that the average response time of the schedule is the average cost of a slot.
In order to bound the average cost of a slot, we define a potential function similar to [6]: $\Phi(S,t)$ is basically the expected total wait of the requests that have not been served at time t, if the schedule would continue with the randomized scheme after time t. We will show that the sum of the average cost of the slots and of the expected extra wait of the unserved requests, $\Phi(S,t)$, is bounded by:

$$\text{AVG} =_{\text{def}} 1/2 + \sum_{i=1}^{m} (p_i \ell_i \tau_i + c_i/\tau_i)$$

which will yield the result.

The requests for M_i that raise between the j^{th} and the $(j+1)^{\text{th}}$ broadcast of the last ℓ_i packets, have to download j more packets and would wait on expectation $j \cdot \tau_i$. We define then:

$$\Phi(S,t) =_{\text{def}} \sum_{i=1}^{m} p_i \sum_{j=1}^{\ell_i} s^t_{i,j} j \tau_i$$

We define for a schedule S, at time t:

$$Y(S,t) =_{\text{def}} \text{AVG} - \text{cost}(t-1) + \Phi(S,t-1) - \Phi(S,t)$$

We will show that for the greedy schedule S: for all t, $Y(S,t) \geqslant 0$, since $\Phi(S,t) \geqslant 0$ and $\Phi(S,0) = 0$, this yields by summing on $t = 0..T$, that the average cost of the slot is bounded by $(1 + 1/T)$ AVG, which tends to AVG, when T goes to infinity.
Let S_k be the schedule which is identical to the greedy schedule up to time slot $(t-1)$, and schedules during time slot t Message M_k, $k \in \{0, 1, \ldots, m\}$. We will show that $\mathbb{E}_k[Y(S_k,t)] = 0$, and that the greedy choice chooses k that maximizes $Y(S_k,t)$.

An algebraic manipulation proves that:

$$\begin{aligned}
\Phi(S_k, t-1) - \Phi(S_k, t) &= -\sum_{i=0}^{m} p_i \ell_i \tau_i + p_k \sum_{j=1}^{\ell_k} s_{k,j}^{t-1} \tau_k \\
\text{Thus,}\quad Y(S_k, t) &= \text{AVG} - c_k - \frac{1}{2} - \sum_{i=0}^{m} p_i \sum_{j=1}^{\ell_i} s_{i,j}^{t-1} - \sum_{i=1}^{m} p_i \ell_i \tau_i + p_k \sum_{j=1}^{\ell_k} s_{k,j}^{t-1} \tau_k \\
\text{and}\quad \mathbb{E}_k[Y(S_k, t)] &= \sum_{k=0}^{m} \frac{1}{\tau_k} Y(S_k, t) = 0 \\
\text{And since,}\quad Y(S_k, t) &= Y(S_0, t) - \left(c_k - p_k \sum_{j=1}^{\ell_k} s_{k,j}^{t-1} \tau_k\right)
\end{aligned}$$

the greedy choice ensures that for all t, $Y(S, t) \geqslant 0$, which completes the proof. □

6.4 Performance ratio of Algorithm in Theorem 8

Proof of Theorem 8. The schedule S constructed Figure 4, verifies the Round Robin property, then Remark 2 implies that we can compute the average response time by *considering that the requests wait back in time.*

Let $\mathcal{L} = \sum_i \ell_i$, and $T + 2\mathcal{L} + U$ be the period of S. The proof of the performance ratio of the greedy algorithm above shows that, the cumulated average response time of a request arriving between time $\mathcal{L}$ and $T + \mathcal{L}$ is bounded by:

$$(T+1)\left\{\frac{1}{2} + \sum_{i=1}^{m} \left(p_i \ell_i \tau_i + \frac{c_i}{\tau_i}\right)\right\} - \Phi(S, T+\mathcal{L}) + \Phi(S, \mathcal{L})$$

Consider the requests which arrive between time $T + \mathcal{L}$ and $T + 2\mathcal{L}$, and which are served before time $T + \mathcal{L}$: their cumulated average response time is bounded by:

$$\Phi(S, T+\mathcal{L}) = \sum_{i=1}^{m} p_i \sum_{j=1}^{\ell_i} s_{i,j}^{T+\mathcal{L}} \tau_i$$

We finally need to consider the requests which arrive between time 0 and T and time $T + \mathcal{L}$ to the end, and are served after time $T + \mathcal{L}$. These request are served after at most $3\mathcal{L}$ time.

Note that: $\Phi(S, \mathcal{L}) \leqslant \mathcal{L} \sum_i p_i \tau_i \ell_i$. Finally, the cost of S is bounded by:

$$\begin{aligned}
\text{COST}(S) &\leqslant \frac{1}{T + 2\mathcal{L} + U} \left\{ (T+1)\left\{\frac{1}{2} + \sum_{i=1}^{m} \left(p_i \ell_i \tau_i + \frac{c_i}{\tau_i}\right)\right\} + \Phi(S, \mathcal{L}) + (3\mathcal{L})^2 + 3\sum_{i=1}^{m} \ell_i c_i \right\} \\
&\leqslant 2\,\text{LB}(M) - \frac{3}{2} + \frac{9\mathcal{L}^2 + 3\sum_{i=1}^{m} \ell_i c_i}{T + 2\mathcal{L} + U}, \quad \text{for } \tau = \tau^*
\end{aligned}$$

which is less than $2\,\text{LB}(M)$ when, $\tau = \tau^*$ and, when the period of S, $T + 2\mathcal{L} + U$, is larger than $6\mathcal{L}^2 + 2\sum_i c_i \ell_i$. □

6.5 Construction the solution of (A) from (A')

Proof of Lemma 10. This proof can be read through the lines of [6]. Consider the solution τ' of (A') with $\tau'_{i_0} < 1$. *W.l.o.g.*, $i_0 = 1$. According to Lemma 9, we have $\lambda \geqslant 0$ so that: for all i, $\tau'_i = \sqrt{(b_i + \lambda)/a_i}$.

By contradiction, suppose that a solution τ^* of (A) does not verify: $\tau_1^* = 1$; thus $\tau_1^* > 1$. We can assume that $\sum_i 1/\tau_i^* = W$, since decreasing τ_1^* improves the solution: 1) if $b_1 \geqslant 0$, because the function $\tau \mapsto (a_1\tau + b_1/\tau)$ is unimodular with minimum $\sqrt{b_1/a_1} \leqslant \tau'_1 < 1$; 2) If $b_1 < 0$, because the function $\tau \mapsto (a_1\tau + b_1/\tau)$ is an increasing function. Thus, since $\sum_i 1/\tau'_i \leqslant \sum_i 1/\tau_i^* = W$, there exists an index, *w.l.o.g.* 2, such that: $\tau_2^* < \tau'_2$.

Consider the following modified minimization problem:

$$(B)\begin{cases} \min\limits_{\tau>0} \sum\limits_{i=1}^{m} \ a_i\tau_i + \dfrac{b_i+\lambda}{\tau_i} \\ \text{Subject to:} \quad (i)\ \forall i,\ \tau_i \geqslant 1 \\ \qquad\qquad\qquad (ii)\ \sum\limits_{i=1}^{m} \dfrac{1}{\tau_i} \leqslant W \end{cases}$$

Since for all i, $(b_i + \lambda) \geqslant 0$, each term $(a_i\tau_i + (b_i + \lambda)/\tau_i)$ of the objective function is unimodular and independently minimized with $\tau_i = \tau'_i$ over $[0, \infty[$. Because of its unimodularity, each term is independently minimized on the domain defined in (B), at the closest point $\tau_i = \max(1, \tau'_i)$ to its minimum τ'_i (Note that:$\sum_i 1/\max(1, \tau'_i) \leqslant \sum_i 1/\tau'_i \leqslant W$). Then, (B) admits a *unique* solution τ'', verifying: $\tau''_i = \max(1, \tau'_i)$.

Then, τ^* is not a solution to (B), since, $\tau_1^* > 1 = \tau''_1 > \tau'_i$, and $1 \leqslant \tau_2^* < \tau'_2 = \tau''_2$. Furthermore, because of the unimodularity of the terms $(a_i\tau_i+(b_i+\lambda)/\tau_i)$, any small decrease δ_1 (resp. increase δ_2) of τ_1^* (resp. τ_2^*), which ensures that (ii) holds, strictly improves the solution. Consider $\delta_1 < 0$ and $\delta_2 > 0$, such that: $\delta_1/\tau_1^{*2} + \delta_2/\tau_2^{*2} = 0$ (this later condition ensures that (ii) holds). The value of the objective function in (B) varies from τ^* to $(\tau^* + \delta)$ of:

$$a_1\delta_1 + a_2\delta_2 - \left(\frac{b_1+\lambda}{\tau_1^{*2}}\ \delta_1 + \frac{b_2+\lambda}{\tau_2^{*2}}\ \delta_2\right) < 0$$

But, this reduces to: $a_1\delta_1 + a_2\delta_2 - \left(b_1\delta_1/\tau_1^{*2} + b_2\delta_2/\tau_2^{*2}\right) < 0$, which is the variation of the objective function of (A) from τ^* to the realizable point $(\tau^* + \delta)$. We obtain then a contradiction: τ^* is not a solution to (A). □