Algorithmica (2003) 35: 146–175
DOI: 10.1007/s00453-002-0990-5

Algorithmica

The Data Broadcast Problem with Non-Uniform Transmission Times[1]

Claire Kenyon[2] and Nicolas Schabanel[3]

Abstract. The Data Broadcast Problem consists of finding an infinite schedule to broadcast a given set of messages so as to minimize a linear combination of the average service time to clients requesting messages, and of the cost of the broadcast. This problem also models the Maintenance Scheduling Problem and the Multi-Item Replenishment Problem. Previous work concentrated on a discrete-time restriction where all messages have transmission time equal to 1. Here, we study a generalization of the model to a setting of continuous time and messages of non-uniform transmission times. We prove that the Data Broadcast Problem is strongly *NP*-hard, even if the broadcast costs are all zero, and give 3-approximation algorithms.

Key Words. Approximation algorithms, Scheduling, Randomized algorithms, *NP*-Completeness, Data broadcasting.

1. Introduction

1.1. *Motivation*. This paper studies an optimization problem which arises in three contexts: data broadcasting, scheduling maintenance service, and multi-item replenishment.

Broadcasting is an efficient means of disseminating data in wireless communication environments, where there is a much larger communication capacity from the information source to the recipients than in the reverse direction. A typical example is satellite access to the Internet: the down link (from the satellite to personal computers equipped with special antenna) is much wider and faster than the up link (using usually phone lines). Another typical situation is mobile clients (e.g., car navigation systems) retrieving information (e.g., traffic information) from a server base-station (e.g., the emitter) through a wireless medium (see [2] and [19]). In these situations, broadcasting protocols reduce the server and up link loads (by eliminating client requests management) and take advantage of the broadcasting media to reduce the down link load (by serving all the clients waiting for the same information at the same time for no extra cost). In broadcasting protocols, items are scheduled continuously, over an infinite time-horizon, thatis to say, the schedule as well as the sequence of requests never terminate, and each request will eventually be served by the broadcast server. The requests do not propagate in the system, but the clients wait for the requested items to be broadcast, thus making the system *pseudo-interactive* or *push-based* (as opposed to *pull-based*): the schedule is

---

[1] A preliminary version of this paper has been published in the *Proceedings of SODA* '99 [16].

[2] LRI, Université Paris XI, Bâtiment 490, F-91405 Orsay, France. http://www.lri.fr/∼kenyon.
[3] CNRS, LIP, UMR CNRS-ENS Lyon-INRIA n°5668, 46 allée d'Italie, F-69364 Lyon Cedex 07, France. http://www.ens-lyon.fr/∼nschaban.

Received December 5, 2000; revised October 3, 2001, and March 9, 2002. Communicated by B. Chazelle. Online publication November 15, 2002.

independent of the incoming requests, since it is oblivious to them. Push-based systems allows us to save bandwidth and to reduce service time by unloading the server from treating the request and by reducing the bandwidth occupied by the most popular items [11], [1]. Acharya [2] and Schabanel [19] present a very complete history of the field. An efficient broadcast scheme seeks to minimize the average service time (i.e., the amount of time spent by a mobile client to obtain a desired piece of information in the broadcast), while also minimizing the resulting broadcast cost (e.g., in the context of data transfers on the World Wide Web, each message has a broadcast cost, the required bandwidth, which is proportional to its length). While most previous work made the simplifying assumption that all items had the same transmission time, our main focus in this paper is to deal rigorously with non-uniform transmission times.

The Maintenance Service Problem schedules m machines for maintenance over an infinite time horizon and seeks to minimize both the costs associated with each maintenance and the operation costs of the machines, where the operation costs of the machines are assumed to increase with the time elapsed since the last maintenance. Here again it seems reasonable to consider the case where sophisticated machines have longer maintenance time than others.

The Multi-Item Replenishment Problem consists of, given m items types, deciding over time when to reorder which item given holding costs, ordering costs, and the rate at which each item is consumed.

All three problems can be modeled similarly.

1.2. *Problem Definition*. In this paper we adopt the data broadcast terminology. The input consists of m messages M_i, $i = 1, \ldots, m$, defined by their *lengths* ℓ_i (the time required to broadcast M_i), their *request probabilities* (or *popularities*) p_i, and their *broadcast costs* c_i. The problem is to decide in what sequence S to schedule the broadcast messages over an infinite time-horizon, so as to minimize the sum of the average service time and of the average broadcast cost, i.e., so as to minimize the lim sup of $\{\mathcal{A}\mathrm{ST}(S, [0, T]) + \mathcal{A}\mathrm{BC}(S, [0, T])\}$ when T tends to infinity; here $\mathcal{A}\mathrm{ST}(S, [0, T])$ denotes the average service time to any request which (1) is generated at a random uniform instant between 0 and T, (2) requests message M_i with probability p_i, and (3) must then wait until the start of the next broadcast of M_i to (4) start downloading M_i (which takes time ℓ_i);[4] and $\mathcal{A}\mathrm{BC}(S, [0, T])$ is the average broadcast cost of the messages whose broadcast starts between the dates 0 and T. This definition can be translated into an algebraic formula, see Lemma 1. As remarked above, the broadcast scheme is pseudo-interactive: the actual requests do not propagate beyond the client, and the schedule is oblivious to the actual sequence of requests. Note that when this definition is specialized to the uniform-lengths setting, our definition agrees with the literature on data broadcasting (see Lemma 1).

In the Maintenance Scheduling Problem [7], [8] the input consists of m machines M_i, $i = 1, \ldots, m$, defined by the *length* ℓ_i *of their maintenance*, their *operation cost*

[4] Note that $\mathcal{A}\mathrm{ST}(S, [0, T])$ is independent of the number of requests arriving in $[0, T]$. If there is only one request, its average service is $\mathcal{A}\mathrm{ST}(S, [0, T])$; if there are many requests, each of them has average service time $\mathcal{A}\mathrm{ST}(S, [0, T])$. Thus our objective function is independent of the density of requests. The only important property is that the request arrival times are uniform in $[0, T]$. This is realized in particular if the requests arrival times are generated by a Poisson process [13].

rate p_i, and their *maintenance cost* c_i. The instantaneous operation cost of a machine is an affine function of the time elapsed since its last maintenance, with linear rate p_i and constant $b \cdot p_i$ for some $b \geq 0$ (the cumulated cost since the last maintenance thus increases quadratically). The problem is to find a maintenance schedule over an infinite time-horizon, so as to minimize the total costs incurred, i.e., the sum of the operation costs and of the maintenance cost. This problem is, as we will see in Proposition 3, identical to the Data Broadcast Problem. The Multi-Item Replenishment Problem consists of, given m items types, deciding when to reorder which items, given holding costs, ordering costs, and the rate at which each item is consumed. Bar-Noy et al. [8] and Anily et al. [7] prove that the Multi-Item Replenishment Problem is a special case of the Maintenance Scheduling Problem.

1.3. *Background*. As far as we know, almost all previous work focused on the uniform-lengths and discrete-time model, when all messages (or machines) have length (or operating time) equal to 1.

We first review results on the discrete-time and uniform-lengths Data Broadcast Problem without broadcast costs. This problem was first studied in the context of Teletext. In [15] Gecsei analyzes the mean response time of memoryless randomized algorithms and proves that any optimal algorithm from this class should choose the next message to broadcast with probability proportional to the square root of its popularity. In [5] Ammar and Wong study periodic broadcast schedules. They derive an algebraic expression for the average service time (which is essentially Lemma 1 below), and prove a lower bound (essentially the one in Lemma 7 below), from which they get inspiration to derive a greedy-type algorithm. They provide numerical evidence that their algorithm is efficient by studying it when the message popularities follow the Zipf distribution, which is claimed to approximate closely real user behavior. In [6] they define the mean response time of arbitrary (not necessarily periodic) schedules, analyze structural properties of optimal schedules, and prove that there exists an optimal schedule which is periodic. From this, they deduce a finite-time algorithm for computing the optimal solution. They also present an algorithm for constructing a good schedule, which uses the golden ratio sequence [6] (they do not provide a performance ratio analysis). Further work [4] attacks a model where successive requests are not independent and prefetching may be used. Greedy heuristics for the same problem are empirically tested in [21] and [7]. This study on the uniform-length case was pursued by Anily et al. [7], who obtained a deterministic 2.5-approximation algorithm, and moreover by Bar-Noy et al. [8] who obtained a deterministic 2-approximation algorithm and proved that the golden ratio heuristic of [6] has approximation ratio $9/8$. In [17] the authors improve this result by giving a polynomial time approximation scheme ($(1+\varepsilon)$-approximation for any $\varepsilon > 0$) for this uniform length case. Finally, in [18] the authors study indexed data broadcast where the objective function to minimize is a combination of response time and tuning time. Note that in the middle 1990s Acharya et al. [3] introduced a simplified model: the broadcast disks. In this model the goal is to partition the set of messages. Each subset is seen as a rotating disk. On each disk the messages are scheduled in round robin order. The rotation speed of each disk is determined according to the popularities of its items. The schedules are obtained by merging the output of the disks. This restrictive setting simplifies the model to study prefechting and caching strategies [2] but prevents one from approaching the optimal strategy. An extension of this approach has been recently proposed by Bar-Noy et al. [9]. They study perfectly periodic schedules in which every message is broadcast at regular fixed time intervals. Note that, again, this restriction to perfect periodic schedules prevents one from closely approaching the optimal strategy.

Even in the uniform-length model, very few articles consider a multi-channel variant of the problem where up to W (the number of channels) messages can be broadcast at the same time. Previous work for the general uniform-length Maintenance Scheduling Problem can be found in [8]. There, the authors prove that there is an optimal schedule which is periodic. Using the operation costs c_i, they prove that the Maintenance Scheduling Problem is *NP*-hard (but are unable to prove the *NP*-hardness of the Data Broadcast Problem). They design a $9/8$-approximation algorithms for the Data Broadcast and the Maintenance Scheduling problems, even when there is more than one channel.

The only references which considers non-uniform length messages are [22] and [23]. There a first lower bound (basically our LB_0 in Section 4) is proposed and experimental results are reported for several heuristics on one or two channels in comparison with this lower bound. More recently, Schabanel has proposed another way to deal with non-uniform-length messages by allowing preemption, which significantly changes the solution [20]. Besides theoretical motivation, the present paper applies to settings where preemption is not allowed.

1.4. *The Results*. This paper focuses on the case where the message lengths (transmission times) are not equal. This non-uniform-length case is significantly different from the uniform-length case: in [6] the authors justify their uniform-length assumption in those terms: "This [uniform lengths] assumption [. . .] is required to render the problem under consideration tractable." In fact, there are some important structural differences between our model and the uniform-length model: for example, in our model an optimal schedule need no longer be periodic (see Section 7). In general it is not even clear a priori whether an optimal solution can be described in finite time! Thus we try to provide a careful definition and treatment of this model. After proving that periodic schedules are arbitrarily close to optimal, our results are then obtained "by density" of the periodic schedules.

We show that even when there are no broadcast costs, the problem is *NP*-hard (note that in [8] *NP*-hardness is proved only when broadcast costs are present).

Then we present a polynomial 3-approximation for the problem. To design that algorithm, we first observe that, as stated by the authors of [8], the natural extension of the lower bound of [5] to our model is no longer tight in the non-uniform-length case, and can, in fact, be arbitrarily bad and can yield arbitrarily bad approximations. One important ingredient of our algorithm here consists in designing a second lower bound which is tight up to a constant factor. This is the key to finding a constant-factor approximation. Our approximation outputs a periodic schedule with quadratic period. It relies on a randomized algorithm that is derandomized.

At the end of the paper we introduce a multi-channel framework. Very little is known about this extension. This last section presents only a tour by examples of some of the differences with the single channel case and with the multiple channel case with uniform transmission times.

1.5. *Organization of the Paper.* The paper is organized as follows. Section 2 presents the notations and facts about the cost of a schedule. It also shows that our definition of the cost function is equivalent to the definition from [4]–[8]. In Section 3 we prove that the cost of periodic schedules is arbitrarily close to the optimal cost, allowing us to manipulate only periodic schedules. Section 4 presents a known lower bound for the problem which is shown to be arbitrarily far from the optimal cost. All the same this lower bound is used in Section 5 to prove that the problem is strongly *NP*-hard even if zero broadcast costs are assumed. We give in Section 6 a new lower bound from which we design a randomized and a deterministic 3-approximation algorithm. Section 7 finally presents a multiple channel framework where several messages can be broadcast at the same time. The last section concludes by giving some directions for further research on that subject.

2. Definitions and Notations

2.1. *The Problem*

The Input. The messages M_i, $i = 1, \ldots, m$, are defined by their *lengths* $\ell_i > 0$, their *request probabilities* or *popularities* $p_i > 0$ such that $p_1 + \cdots + p_m = 1$, and their *broadcast costs* $c_i \geq 0$.

The Schedule. A *schedule* S of $M_1, \ldots, M_m$ is an infinite sequence $M_{s(1)}M_{s(2)}\cdots$, where $1 \leq s(i) \leq m$. We denote by $t(n)$ the starting time of the broadcast of $M_{s(n)}$, which must satisfy $t(n+1) \geq t(n) + \ell_{s(n)}$.

A schedule S is *periodic* with *period* $T > 0$, if there exists $N > 0$ such that for any $n \geq 1$, $s(n+N) = s(n)$ and $t(n+N) = t(n) + T$. Because of their lack of structure, non-periodic schedules are hard to handle as opposed to periodic schedules. Periodic schedules are a useful tool to obtain properties of optimal schedules.

The Objective Function. We are interested in minimizing a combination of two quantities on S. The first one, denoted by $\mathcal{A}\mathrm{ST}(S)$, is the *average service time* to a random request (where the average is taken over the moments when requests occur *and* over the type M_i of message requested). If we define for a time interval I, $\mathcal{A}\mathrm{ST}(S, I)$ as the average service time of a random request arriving in I, $\mathcal{A}\mathrm{ST}(S)$ is defined as

$$\mathcal{A}\mathrm{ST}(S) = \limsup_{T\to\infty} \mathcal{A}\mathrm{ST}(S, [0, T]).$$

Note that Lemma 1 gives an algebraic expression for $\mathcal{A}\mathrm{ST}(S, I)$ in terms of S, I, and the p_i's.

The second quantity is the *average broadcast cost* $\mathcal{A}\mathrm{BC}(S)$ of the messages, defined as the asymptotic value of the average broadcast cost of S over any time interval (where the average is over time). By definition, each broadcast of a message M_i costs c_i. For a time interval I, the average broadcast cost of S over I, $\mathcal{A}\mathrm{BC}(S, I)$, is defined as the sum of the cost of all the messages whose broadcast begins in I, divided by the length of I. $\mathcal{A}\mathrm{BC}(S)$ is then defined as

$$\mathcal{A}\mathrm{BC}(S) = \limsup_{T\to\infty} \mathcal{A}\mathrm{BC}(S, [0, T]).$$

The quantity which we want to minimize is the *cost* of a schedule S which we define as follows:

$$\mathrm{COST}(S) = \mathcal{A}\mathrm{ST}(S) + \mathcal{A}\mathrm{BC}(S).$$

General Notations. We adopt the following notations for the rest of the paper:

- $x_{\max} =_{\mathrm{def}} \max_i x_i$ is the maximum value of the x_i's (similarly, $x_{\min} =_{\mathrm{def}} \min_i x_i$).
- $\mathbb{E}_\pi[x] =_{\mathrm{def}} \sum_i \pi_i x_i$ is the expected value of the x_i's given the discrete distribution (π_i). For example, $\mathbb{E}_p[\ell]$ is the average downloading time because message M_i is requested (and downloaded) with probability p_i.
- $\mathcal{L} =_{\mathrm{def}} \ell_1 + \cdots + \ell_m$ is the sum of the lengths of the messages and $\mathcal{C} =_{\mathrm{def}} c_1 + \cdots + c_m$ is the sum of the broadcast costs of the messages.
- OPT is the optimal cost: $\mathrm{OPT} = \inf_S \mathrm{COST}(S)$.

2.2. *An Algebraic Expression for the Cost*. Lemma 1 introduces notations which are used throughout the paper. Proposition 3 relates our definition of cost to the algebraic definition given in previous work [5], [8] for discrete time and uniform lengths on either data broadcasting or maintenance scheduling.

LEMMA 1 (Algebraic Definition of COST). *Consider a schedule S of m messages M_1, ..., M_m and a time interval I. Take a request generated at a random uniform instant of I and asking for M_i with probability p_i. Then the average service time to the request is*

$$\mathcal{A}\mathrm{ST}(S, I) = \mathbb{E}_p[\ell] + \frac{1}{2|I|}\sum_{i=1}^{m} p_i \left\{ \sum_{j=1}^{n_i+1} (t_j^i)^2 - (\Delta t^i)^2 \right\}$$

and the average broadcast cost of S over I is

$$\mathcal{A}\mathrm{BC}(S, I) = \frac{1}{|I|}\sum_{i=1}^{m} n_i c_i,$$

where (*see Figure* 1):

- *n_i is the number of broadcasts of M_i starting in I according to S.*
- *t_1^i is the time elapsed from the beginning of I to the beginning of the first broadcast of M_i; and t_j^i, for $j \geq 2$, denotes the time elapsed from the* $(j-1)$st *to the* jth *broadcast of M_i since the beginning of I.*
- *Δt^i is the length of the interval from the end of I to the first broadcast of M_i starting after I.*

PROOF. Consider a request Q in I. Q requests for M_i with probability p_i. With probability $t_j^i/|I|$, for $1 \leq j \leq n_i$, Q arrives in an interval of length t_j^i and waits $t_j^i/2$ on average until the next broadcast of M_i and then ℓ_i more until the end of the downloading. With probability $(t_{n_i+1}^i - \Delta t^i)/|I|$, Q arrives in the last interval and waits $((t_{n_i+1}^i - \Delta t^i)/2 + \Delta t^i) = (t_{n_i+1}^i + \Delta t^i)/2$ on average until the next broadcast of M_i, plus ℓ_i more until the end of the downloading. Summing over all the intervals and messages yields the expression given for $\mathcal{A}\mathrm{ST}(S, I)$.

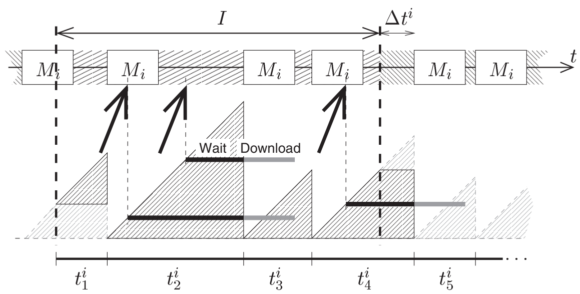

Fig. 1. Illustration of the notations and expression in Lemma 1. Here, $n_i = 3$. The top of the figure shows the broadcast sequence over time, with broadcasts of M_i singled out. The three black arrows indicate requests for M_i. To each request is associated wait and download times, indicated by a black and grey line, respectively. The triangles are such that at time t, the height of the triangle is proportional to the average number of requests for M_i waiting to be served. The heavily shaded part of the triangles correspond to requests which originated inside interval I. The cumulated wait of the requests for M_i can be seen as the sum of the areas of the triangles whose bases are the intervals between consecutive broadcasts of M_i.

The average cost for broadcasting M_i during I is $n_i c_i/|I|$. Summing over all the messages yields the expression given for $\mathcal{A}\mathrm{BC}(S)$. □

COROLLARY 2. *If S is a periodic schedule of period T, then*

$$\mathcal{A}\mathrm{ST}(S) = \mathbb{E}_p[\ell] + \frac{1}{2T}\sum_{i=1}^{m} p_i \sum_{j=1}^{n_i}(t_j^i)^2 \quad \textit{and} \quad \mathcal{A}\mathrm{BC}(S) = \frac{1}{T}\sum_{i=1}^{m} n_i c_i,$$

where t_j^i denotes the time elapsed from the $(j-1)$st to the jth broadcast of M_i in a period of S.

PROOF. For $\mathcal{A}\mathrm{BC}(S)$, the statement is clear. To calculate the average service time, consider requests for M_i. We apply Lemma 1 to an interval I of length T starting exactly at the beginning of a broadcast of M_i. By construction of I, taking the notation of Lemma 1, we have $\Delta t^i = 0$ and $|I| = T$. Lemma 1 then yields the claimed expression for $\mathrm{COST}(S, I)$. Since S is periodic with period T, we have $\mathrm{COST}(S) = \mathrm{COST}(S, I)$, which concludes the proof. □

2.3. *An Equivalent Definition of the Cost.* This section establishes that the problem studied here is the same as in [8] and [7] by showing that the service time is equivalently defined by the average time elapsed since the last broadcast, plus the message length. In other terms, we show here that time is reversible for this problem. This is straightforward for periodic schedules but requires some work for general, non-periodic schedules. This proposition is used in particular to design the deterministic approximation algorithms in Section 6.

DEFINITION. For any given schedule S and any time t, we define the *penalty* $\mathrm{PE}(S, t, M_i)$ of a schedule S at time t for message M_i as p_i multiplied by the time elapsed since the beginning of the last broadcast of M_i. Note that the penalty is proportional to the expected number of unserved requests for M_i at time t.

If t_i is the date of the beginning of the last broadcast of M_i before time t, we have $\mathrm{PE}(S, t, M_i) = p_i(t - t_i)$ (we assume that a broadcast of M_i occurs at time $t = 0$ in the definition). We define then the *average penalty* $\mathcal{A}\mathrm{PE}(S, I)$ *of S over a time interval I* as follows:

$$\mathcal{A}\mathrm{PE}(S, I) = \frac{1}{|I|} \int_I \sum_{i=1}^{m} \mathrm{PE}(S, t, M_i)\, dt;$$

and the *average penalty* of S as

$$\mathcal{A}\mathrm{PE}(S) = \limsup_{T\to\infty} \mathcal{A}\mathrm{PE}(S, [0, T]).$$

We then define a new cost $\widehat{\mathrm{COST}}$ for schedule S as

$$\widehat{\mathrm{COST}}(S) = \mathbb{E}_p[\ell] + \mathcal{A}\mathrm{PE}(S) + \mathcal{A}\mathrm{BC}(S). \tag{1}$$

This definition of the cost matches the one in [8] and [7]. The next claim proves that the two cost functions COST and $\widehat{\mathrm{COST}}$ are identical. (Note that this property is clear for periodic schedules, since in that case both quantities (see Corollary 2) sum the squares of the inter-broadcast lengths.)

PROPOSITION 3 (Time Reversibility). *For any schedule S,*

$$\mathcal{A}\mathrm{ST}(S) = \mathcal{A}\mathrm{PE}(S) + \mathbb{E}_p[\ell].$$

PROOF. Consider a schedule S. We adopt the following notations:

- $n_i(T)$ is the number of broadcasts of M_i that begin in $[0, T]$ in S.
- t^i_j is the time elapsed between the $(j-1)$st and jth broadcasts of M_i (we assume that a 0th broadcast of M_i occurs at time $t = 0$).

According to Lemma 1 and also Figure 2, we have

$$\mathbb{E}_p[\ell] + \frac{1}{2T} \sum_{i=1}^{m} p_i \sum_{j=1}^{n_i(T)} (t^i_j)^2 \le \mathcal{A}\mathrm{ST}(S, [0, T]) \le \mathbb{E}_p[\ell] + \frac{1}{2T} \sum_{i=1}^{m} p_i \sum_{j=1}^{n_i(T)+1} (t^i_j)^2$$

and

$$\mathbb{E}_p[\ell] + \underbrace{\frac{1}{2T} \sum_{i=1}^{m} p_i \sum_{j=1}^{n_i(T)} (t^i_j)^2}_{=_{\mathrm{def}} A(T)} \le \mathbb{E}_p[\ell] + \mathcal{A}\mathrm{PE}(S, [0, T]) \le \mathbb{E}_p[\ell] + \underbrace{\frac{1}{2T} \sum_{i=1}^{m} p_i \sum_{j=1}^{n_i(T)+1} (t^i_j)^2}_{=_{\mathrm{def}} B(T)}.$$

We will prove that the lower and upper bounds, $A(T)$ and $B(T)$, above both have the same lim sup. This will immediately imply that $\mathcal{A}\mathrm{ST}(S) = \mathcal{A}\mathrm{PE}(S) + \mathbb{E}_p[\ell]$.

If the lim sup of $A(T)$ is $+\infty$, then clearly $\mathcal{A}\mathrm{ST}(S) = \mathcal{A}\mathrm{PE}(S) = +\infty$.

Otherwise, take any κ such that for all T large enough, $A(T) \le \kappa$. In particular, for every i, take $A(T_i)$ with $T_i = t^i_1 + \cdots + t^i_{n_i(T)+1} \le T + t^i_{n_i(T)+1}$, we have $A(T_i) \le \kappa$ and

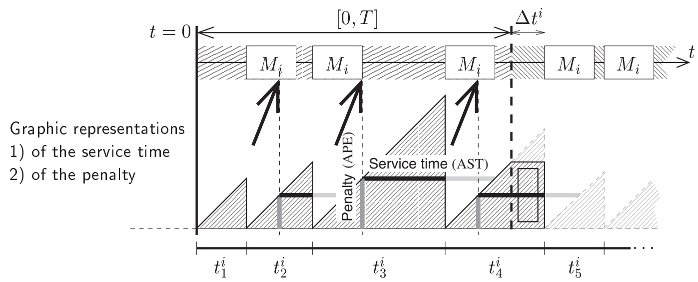

Fig. 2. Illustration of Proposition 3. Here, $n_i(T) = 3$.

thus

$$\frac{p_i}{2(t_1^i + \cdots + t_{n_i(T)+1}^i)} \sum_{j=1}^{n_i(T)+1} (t_j^i)^2 \le \kappa.$$

That is to say,

$$\sum_{j=1}^{n_i(T)+1} (t_j^i)^2 \le 2\frac{\kappa}{p_i}(t_1^i + \cdots + t_{n_i(T)+1}^i) \le 2\frac{\kappa}{p_i}(T + t_{n_i(T)+1}^i). \tag{2}$$

Equation (2) implies two things:

- First, that for any i, $(t_{n_i(T)+1}^i)^2 \le (2\kappa/p_i)(T + t_{n_i(T)+1}^i)$, hence $t_{n_i(T)+1}^i = O(\sqrt{T})$.
- Second, (2) with this result yields that

$$B(T) = \frac{1}{2T}\sum_{i=1}^{m} p_i \sum_{j=1}^{n_i(T)+1} (t_j^i)^2 \le \frac{\kappa}{T}\left(T + \max_{1\le i\le m} t_{n_i(T)+1}^i\right) \le \kappa + O\left(\frac{1}{\sqrt{T}}\right).$$

This is true for any κ such that $A(T) \le \kappa$ for all T large enough, it follows that the lower and upper bounds, $A(T)$ and $B(T)$, have the same lim sup, and so $\mathcal{A}\mathrm{ST}(S) = \mathcal{A}\mathrm{PE}(S) + \mathbb{E}_p[\ell]$. □

3. A Reduction to Periodic Schedules. Previous work relied on the existence of a periodic optimal schedule. Although this statement still holds when there are no broadcast costs (see Lemma 10), we do not currently know whether there is an optimal schedule which is periodic in the case of broadcast costs. However, for the purpose of approximation, it is sufficient to consider periodic schedules, as the following lemma shows.

LEMMA 4 (Reduction to Periodic Schedules). *For any set of messages $M_1, \dots, M_m$,*

$$\mathrm{OPT} = \inf_{S \text{ periodic}} \mathrm{COST}(S).$$

PROOF. It is enough to show that for any schedule S of the messages $M_1, \dots, M_m$, and for any $\varepsilon > 0$, their exists a periodic schedules S' with cost $\mathrm{COST}(S') \le \mathrm{COST}(S) + \varepsilon$.

Let S be a schedule of $M_1, \dots, M_m$ and $\varepsilon > 0$. By definition of the cost of S, take T so that for any $t \geq T$, $\mathrm{COST}(S, [0, t]) \leq \mathrm{COST}(S) + \varepsilon/2$. We choose $t \geq T$, matching the date of the end of the broadcast of some message in S, that will be fixed later. Let S' be the periodic schedule with period $(t + \mathcal{L})$ which broadcasts the same messages as S during $[0, t]$, and then broadcasts during $[t, t + \mathcal{L}]$ the m messages in order of next appearance in S after time t. This ensures that the requests which arise in S' during $[0, t]$ are served no later than in S. A request that arises in S' during $[t, t+\mathcal{L}]$ is served after at most $(\mathcal{L} + \mathrm{ST}(S, 0))$ time, where $\mathrm{ST}(S, 0)$ denotes the average service time for a request that arises at time 0 in S. Then

$$\mathcal{A}\mathrm{ST}(S') \leq \mathcal{A}\mathrm{ST}(S, [0, t]) + \underbrace{\frac{\mathcal{L}(\mathcal{L} + \mathrm{ST}(S, 0))}{t + \mathcal{L}}}_{a(t) = O(1/t)}.$$

As for the broadcast costs, we have

$$\mathcal{A}\mathrm{BC}(S') \leq \mathcal{A}\mathrm{BC}(S, [0, t]) + \underbrace{\frac{\mathcal{C}}{t + \mathcal{L}}}_{b(t) = O(1/t)}.$$

Hence,

$$\mathrm{COST}(S') \leq \mathrm{COST}(S, [0, t]) + (a(t) + b(t)) \leq \mathrm{COST}(S) + \frac{\varepsilon}{2} + (a(t) + b(t)),$$

and choosing t large enough (so that $a(t) + b(t) = O(1/t) \leq \varepsilon/2$) yields the result. □

4. About a Known Lower Bound. Finding good lower bounds is a key point to designing provably efficient algorithms. The first lower bound LB_0 below is a natural generalization of the one from [22] and [8]. Unfortunately, this lower bound is not tight when the messages have different lengths (see Fact 8), that is to say, it can take values arbitrarily far from the optimal and then cannot be used to analyze the performances of algorithms. This lower bound will however be used in the *NP*-hardness proof of Theorem 11 and some other examples, which is why we present it in detail. In Section 6 we design a second tight lower bound.

We define LB_0 as the following minimization problem:[5]

$$\mathrm{LB}_0 \begin{cases} \mathbb{E}_p[\ell] + \min_{\boldsymbol{\tau} > \mathbf{0}} \sum_{i=1}^{m} \left(p_i \frac{\tau_i}{2} + \frac{c_i}{\tau_i} \right) \\ \text{subject to} \quad \sum_{i=1}^{m} \frac{\ell_i}{\tau_i} \leq 1. \end{cases}$$

Lemma 7 will show that its value is a lower bound for our problem. However, first we see how to solve this minimization problem.

[5] Note that the minimum within the optimization domain $\{\boldsymbol{\tau} : \boldsymbol{\tau} > \mathbf{0}$ and $\sum_{i=1}^{m} (\ell_i/\tau_i) \leq 1\}$ is well defined, since whenever a τ_i tends to zero or to infinity, the objective function tends to infinity.

LEMMA 5. *The minimization problem* LB_0 *admits a unique solution* $\boldsymbol{\tau}^*$ *verifying*

$$\tau_i^* = \sqrt{\frac{2c_i + \lambda^* \ell_i}{p_i}}$$

for a certain $\lambda^* \geq 0$ (*independent of* i), *such that* $\lambda^* = 0$ *if* $\sum_{i=1}^m \ell_i \sqrt{p_i/2c_i} \leq 1$; *otherwise*, λ^* *is the unique positive solution to* $\sum_{i=1}^m \ell_i \sqrt{p_i/(2c_i + \lambda^* \ell_i)} = 1$.

With a slight abuse of notation, we use LB_0 to denote both the problem and its optimal value; the meaning should be clear from the context.

Newton's algorithm can be used to compute efficiently the numerical approximation of $\boldsymbol{\tau}^*$ within arbitrary accuracy. Section 6.5 at the end of the algorithmic section presents some of these implementation aspects.

PROOF. As in [8] we use Lagrangian relaxation. We explain this part in some detail to justify rigorously the validity of the Lagrangian relaxation technique in this setting. Let $f(\boldsymbol{\tau}) = \sum_{i=1}^m (p_i \tau_i + 2c_i/\tau_i)$ denote the variable term of the objective function, and let $g(\boldsymbol{\tau}) = 1 - \sum_{i=1}^m (\ell_i/\tau_i)$ denote the (main) constraint of the system. The constraints define a subdomain D of $\mathbb{R}_+^m$. There are two cases:

- The minimum of f in D is reached in the interior of D, in which case it is a critical point[6] of f. Note that the particular f has a unique critical point $\boldsymbol{\tau}^\#$ in $\mathbb{R}_+^m$, defined by $\tau_i^\# = \sqrt{2c_i/p_i}$, and which is the global minimum of f in $\mathbb{R}_+^m$. Thus this case occurs if and only if $\sum_{i=1}^m \ell_i \sqrt{p_i/2c_i} < 1$. In this case the formula in the statement of the lemma holds for $\lambda^* = 0$.
- The minimum of f in D is reached on the boundary of D. Since $f(\boldsymbol{\tau}) \to \infty$ as soon as some τ_i goes to zero or to infinity, this minimum has to be on the surface defined by the constraint $g(\boldsymbol{\tau}) = 0$. Then, according to classical minimization results, the gradients of f and g are collinear, and moreover they point in the same direction since f increases when $\boldsymbol{\tau}$ enters the domain $g(\boldsymbol{\tau}) > 0$. Thus $\boldsymbol{\tau}^*$ verifies two equations, for some $\lambda^* \geq 0$:

$$(\Sigma) \qquad \begin{cases} \operatorname{grad} f(\boldsymbol{\tau}^*) = \lambda^* \ \operatorname{grad} g(\boldsymbol{\tau}^*), \\ g(\boldsymbol{\tau}^*) = 0. \end{cases}$$

Classically, this system can be rewritten by defining $F(\boldsymbol{\tau}, \lambda) = f(\boldsymbol{\tau}) - \lambda g(\boldsymbol{\tau})$, as follow:

$$\begin{aligned} (\Sigma) \quad &\iff \quad (\boldsymbol{\tau}^*, \lambda^*) \text{ is a critical point of } F \\ &\iff \quad \begin{cases} \dfrac{\partial F}{\partial \tau_i}(\boldsymbol{\tau}^*, \lambda^*) = 0, & \text{for all } i, \\ \dfrac{\partial F}{\partial \lambda}(\boldsymbol{\tau}^*, \lambda^*) = 0. \end{cases} \end{aligned}$$

[6] Recall that a critical point is a point $\boldsymbol{\tau}^\#$ such that for all i, $(\partial f/\partial \tau_i)(\boldsymbol{\tau}^\#) = 0$.

However,

$$\frac{\partial F}{\partial \tau_i}(\boldsymbol{\tau}, \lambda) = p_i - \frac{2c_i}{\tau_i^2} - \frac{\lambda \ell_i}{\tau_i^2} \quad \text{and} \quad \frac{\partial F}{\partial \lambda}(\boldsymbol{\tau}, \lambda) = \sum_{i=1}^{m} \frac{\ell_i}{\tau_i} - 1.$$

So (Σ) has a unique solution $(\boldsymbol{\tau}^*, \lambda^*)$ verifying $\tau_i^* = \sqrt{(2c_i + \lambda^* \ell_i)/p_i}$, $\lambda^* \geq 0$, and $\sum_{i=1}^{m} \ell_i \sqrt{p_i/(2c_i + \lambda^* \ell_i)} = 1$. We conclude that F admits a *unique* critical point $\tau_i^* = \sqrt{(2c_i + \lambda^* \ell_i)/p_i}$ for some $\lambda^* \geq 0$ satisfying $\sum_{i=1}^{m} \ell_i \sqrt{p_i/(2c_i + \lambda^* \ell_i)} = 1$. This is then the unique minimum for f on domain D.

In both cases we have shown that the objective function f in LB_0 admits a unique minimum $\boldsymbol{\tau}^*$ with the claimed value. □

Note that when all messages have zero broadcast cost, Corollary 6 gives the "square-root rule" in [22] inspired from the analysis of the uniform-length case in [5].

COROLLARY 6 (Square-Root Rule). *If all the messages have zero broadcast cost* ($c_i = 0$, *for all* i), *then*

$$\mathrm{LB}_0 = \mathbb{E}_p[\ell] + \frac{1}{2} \left(\sum_{i=1}^{m} \sqrt{p_i \ell_i} \right)^2 \quad \text{and} \quad \tau_i^* = \sqrt{\frac{\ell_i}{p_i} \lambda^*}$$

with $\lambda^* = (\sum_{j=1}^{m} \sqrt{p_j \ell_j})^2$.

PROOF. According to Lemma 7, when all messages have zero broadcast cost, $\boldsymbol{\tau}^*$ verifies

$$\sum_{i=1}^{m} \frac{\ell_i}{\tau_i^*} = 1. \tag{3}$$

Indeed, if the constraint was not an equality in LB_0, one could decrease one of the τ_i^*'s and then decrease the objective function in LB_0.

Now, $\tau_i^* = \sqrt{\ell_i \lambda^* / p_i}$ for some $\lambda^* \geq 0$. Putting this expression in (3) then gives the claimed value of λ^*, from which we compute the values of τ_i^* and LB_0. □

LEMMA 7 (LB_0). LB_0 *is a lower bound to the cost of any schedule of* $M_1, \ldots, M_m$. *Furthermore, a periodic schedule* S *has cost* LB_0 *if and only if* S *schedules each message* M_i *periodically at intervals of lengths exactly* τ_i^*, *where* $\boldsymbol{\tau}^*$ *is the unique solution to* LB_0.

PROOF. Without loss of generality (from Lemma 4), we restrict ourselves to periodic schedules. The proof is then along the lines of [5] and [8], relaxing the problem by allowing message broadcasts to overlap. Consider a periodic schedule S with period T. We use the results and notations in Corollary 2. The contribution of each message M_i to the cost is $\mathrm{COST}(S, M_i) = p_i \ell_i + (\sum_{j=1}^{n_i} p_i ((t_j^i)^2 / 2T)) + (n_i/T) c_i$.

For every given n_i, since $\sum_{j=1}^{n_i} t_j^i = T$, the sum of the squares is minimized if and only if all terms t_j^i are equal: if $t_j^i = T/n_i$ for all j. We define $\tau_i = T/n_i$, we get that

$$\mathrm{COST}(S, M_i) \geq p_i \ell_i + p_i \frac{\tau_i}{2} + \frac{c_i}{\tau_i}.$$

Summing up the different contributions then yields a lower bound on the cost of S:

$$\mathrm{COST}(S) \geq \mathbb{E}_p[\ell] + \sum_{i=1}^{m} \left(p_i \frac{\tau_i}{2} + \frac{c_i}{\tau_i} \right). \tag{4}$$

Since the total available bandwidth over a period is T, we have $\sum_{i=1}^{m} n_i \ell_i \leq T$, which yields the constraint on the τ_i's. LB_0 is thus a lower bound on the cost of S.

Realizability of the lower bound. According to the proof above, a periodic schedule S has cost $\leq \sum_{i=1}^{m} (p_i \tau_i/2 + c_i/\tau_i)$ where $\tau_i = T/n_i$ if and only if it schedules message M_i exactly every τ_i. Since LB_0 has a unique solution $\boldsymbol{\tau}^*$ (Lemma 5), a periodic schedule whose cost is equal to LB_0, necessarily broadcasts M_i exactly every τ_i^*. □

We say that a lower bound LB for a minimization problem is *tight* if the ratio OPT / LB is bounded. The lower bound above was tight in the setting of [8]. However, in our model where the message lengths vary, this is no longer the case: OPT / LB_0 can be arbitrarily large; moreover, using the frequencies $1/\tau_i^*$ as in [5] and [8] to broadcast the messages (M_i) can generate arbitrarily bad schedules.

FACT 8. LB_0 *is not tight*, *even when messages have zero broadcast cost*.

PROOF. Consider the following two messages M_1 and M_2 with zero broadcast cost: M_1 has length 1 and popularity $L/(L+1)$; and M_2 has length L and popularity $1/(L+1)$. Lemma 6 gives $\mathrm{LB}_0 = 4L/(L+1)$ ($\tau_1^* = 2$ and $\tau_2^* = 2L$). Then $\mathrm{LB}_0 = \Theta(1)$, when L goes to infinity. However, according to [10], the optimal schedule alternates scheduling message M_1 $\Theta(L\sqrt{L})$ times and message M_2 once, and has cost $\mathrm{OPT} = \sqrt{L} + O(1) \gg \Theta(1) = \mathrm{LB}_0$.

Furthermore, note that if one tries to schedule message M_1 with frequency proportional to $1/\tau_1^* = \Theta(1)$ and message M_2 with frequency proportional to $1/\tau_2^* = \Theta(1/L)$, broadcasting M_2 occupies $L \cdot \Theta(1/L) = \Theta(1)$ of the channel. Then $\Omega(1)$ of the requests for M_1 arise during a broadcast of M_2 and then wait on average $L/2$. The cost of the resulting schedule S is then at least $(L/2)\Omega(1) = \Omega(L)$ and

$$\mathrm{COST}(S) = \Omega(L) \gg \mathrm{OPT} = \Theta(\sqrt{L}) \gg \mathrm{LB}_0 = \Theta(1).$$ □

The next section uses this lower bound to prove that the Data Broadcast Problem with non-uniform lengths is *NP*-hard. In Section 6 we design a new lower bound which is tight and use it to design an algorithm with constant approximation ratio.

5. *NP*-Hardness of the Data-Broadcast Problem. We will show that even when there are no broadcast cost, the Data Broadcast Problem is already strongly *NP*-hard. In all of this section we assume that all broadcast costs are equal to zero.

5.1. *Structural Properties of an Optimal Solution*

LEMMA 9 (Maximum Interval). *Let S be a periodic schedule of $M_1, \dots, M_m$. There exists a periodic schedule S' with* $\mathrm{COST}(S') \leq \mathrm{COST}(S)$ *and such that for any i, the intervals between two broadcasts of M_i all have length at most $\mathcal{K}$*:

$$\mathcal{K} = \frac{5\ell_{\max}}{p_{\min}}.$$

PROOF. The proof works by induction.

Construction of S_1. Consider in S an interval with maximum length, A_1, between two consecutive broadcasts of the same message. Without loss of generality, this message is M_1 and the first of those two broadcasts of M_1 occurs at time 0. Assume $A_1 \geq \mathcal{K}$. We construct a periodic schedule S_1 by inserting in S a broadcast of M_1 at time $(A_1 + \ell_1 - \ell_{\max})/2$, or as soon as possible after that time (that is, if there is a message being broadcast, we wait until the broadcast is finished before inserting M_1); let t_1 be this date: $(A_1 + \ell_1 - \ell_{\max})/2 \leq t_1 < (A_1 + \ell_1 + \ell_{\max})/2$.

Analysis of S_1. If T is the period of S, then S_1 has period $T + \ell_1$. We show that the cumulated cost of S_1 is smaller than the cumulated cost in S, i.e., $\Delta =_{\mathrm{def}} (T + \ell_1)\,\mathrm{COST}(S') - T\,\mathrm{COST}(S) \leq 0$.

- According to Lemma 1, in S the cumulated service time for M_1 has a term $p_1(A_1^2/2)$, accounting for the interval in which M_1 is inserted. In S_1 that term disappears and is replaced by
$$\frac{p_1}{2}(t_1^2 + (A_1 + \ell_1 - t_1)^2) \leq p_1\left(\frac{A_1 + \ell_1 + \ell_{\max}}{2}\right)^2$$
since the insertion time t_1 of M_1 satisfies $(A_1+\ell_1-\ell_{\max})/2 \leq t_1 < (A_1+\ell_1+\ell_{\max})/2$.
- For $i \geq 2$, let A_i denote the length of the interval in S between two broadcasts of M_i, in which M_1 is inserted. In S the cumulated service time for M_i ($i \geq 2$) has a term $p_i(A_i^2/2)$, accounting for the interval in which M_1 is inserted. In S_1 that term disappears and is replaced by $p_i((A_i + \ell_1)^2/2)$.

The overall variation Δ is thus bounded by

$$\begin{aligned}\Delta \;&\leq\; \frac{p_1}{4}((A_1 + \ell_1 + \ell_{\max})^2 - 2A_1^2) + \sum_{i=2}^{m} \frac{p_i}{2}((A_i + \ell_1)^2 - A_i^2)\\ &=\; \frac{p_1}{4}(-A_1^2 + 2A_1(\ell_1 + \ell_{\max}) + (\ell_1 + \ell_{\max})^2) + \sum_{i=2}^{m} \frac{p_i}{2}(2\ell_1 A_i + \ell_1^2).\end{aligned}$$

Using $\ell_1 \leq \ell_{\max}$ and $A_i \leq A_1$, and replacing $p_2 + \cdots + p_m$ by $1 - p_1$, we get

$$\Delta \leq -\frac{p_1}{4}A_1^2 + \ell_{\max}A_1 + \ell_{\max}^2.$$

In terms of A_1, this last expression is a polynomial of degree 2 with negative leading coefficient. Since $A_1 \geq \mathcal{K}$ and $\mathcal{K}$ is larger than the maximum root of that polynomial, we conclude that Δ is negative.

The induction. Repeating the construction, we note that any two insertions of the same message must be at distance at least $\mathcal{K}/2$ from each other, hence there can be only a finite number of iterations, after which we obtain the requested schedule S'. □

Note that Schabanel [19] extends the above lemma to the setting where there are positive broadcast costs.

Since all the messages have zero broadcast cost, it never helps for a schedule to remain idle: one can always decrease the service time of a schedule by skipping all the idle periods. Consequently, we can restrict our search for optimal schedules to schedules that never idle.

The following lemma states that in this situation, there is an optimal schedule which is periodic (note that we currently do not know whether an optimal periodic schedule exists if broadcast costs are present).

LEMMA 10. *If the messages have no broadcast costs, then there exists an optimal schedule S^* which is periodic.*

PROOF. Since there are no broadcast costs, we only consider schedules with no idle times. We use Lemma 4 again. We will prove that *among periodic schedules*, there exists a schedule S^* of minimum cost. Since

$$\mathrm{COST}(S^*) = \inf_{S \text{ periodic}} \mathrm{COST}(S) = \mathrm{OPT},$$

S^* will be optimal among all schedules. Henceforth we only consider periodic schedules. Moreover, thanks to Lemma 9, we can restrict our search to schedules in which any interval where M_i is not broadcast has length at most $\mathcal{K}$.

We use the penalty-based definition of the cost (see Section 2.3). By Proposition 3,

$$\mathrm{COST}(S) = \mathbb{E}_p[\ell] + \mathcal{A}\mathrm{PE}(S).$$

We use a graph construction similar to that in [6] and [7]. We consider the following weighted directed graph G with costs. A vertex of G is an m-tuple $\langle a_1, \ldots, a_m \rangle$ with $0 \leq a_i \leq \mathcal{K}$. Semantically, being at that vertex at time t means that for all i, the last broadcast of M_i started at time $t - a_i$. Thus a_i is a sum of values in $\{\ell_1, \ell_2, \ldots, \ell_m\}$ bounded by $\mathcal{K}$, hence can only take a finite number of distinct values; G is thus a finite graph. For every i, there is an edge e from $\langle a_1, \ldots, a_i, \ldots, a_m \rangle$ to $\langle a_1 + \ell_i, \ldots, \ell_i, \ldots, a_m + \ell_i \rangle$, of *length* $\ell(e) = \ell_i$. Following that edge means that M_i is the next message broadcast. We associate to that edge a *cost* $c(e)$ which is the sum of the cumulated average downloading time $\ell_i \mathbb{E}_p[\ell]$ and of the cumulated penalty over the corresponding period of time. During this period, the average penalty for M_j is: $p_j(a_j + \ell_i/2)$, if $j \neq i$, and $p_i(\ell_i/2)$ if $j = i$. So the cost of e is defined as

$$c(e) = \ell_i \cdot \left(\mathbb{E}_p[\ell] + \frac{\ell_i}{2} + \sum_{j=1, j \neq i}^{m} p_j a_j \right).$$

The schedules we consider are in bijection with the cycles of G, and, by Proposition 3, the cost of a schedule is exactly the mean cost of the corresponding cycle γ: $\sum_{e\in\gamma} c(e)/\sum_{e\in\gamma} \ell(e)$. Since the cost of every cycle γ of H is the weighted sum of the cost of the elementary cycles composing it, and there are only a finite number of elementary cycles, the minimum is reached by a particular cycle γ^*. The corresponding schedule is optimal. □

5.2. *NP-Hardness*. In order to stay in the Turing machine framework for the *NP*-hardness proof, we restrict the input to rational numbers.

THEOREM 11 (*NP*-Hardness). *The decision problem associated to the restriction of the Data Broadcast Problem to the case of messages of rational length and popularity and with zero broadacst cost is strongly NP-hard.*

PROOF. We use a reduction of N-partition [14], which, given a sequence of m integers $x_1, \ldots, x_m$, decides whether there exists a partition of $\{1, \ldots, m\}$ into N sets $I_1, \ldots, I_N$ such that $\sum_{i\in I_1} x_i = \cdots = \sum_{i\in I_N} x_i$. Let $\mathcal{S} =_{\text{def}} x_1 + \cdots + x_m$. Consider the following instance of data broadcast: $(m + 1)$ messages $M_1, \ldots, M_m, M_{m+1}$ with zero broadcast costs such that

$$\begin{cases} M_{i\le m}\colon \quad \ell_i = x_i \quad & \text{and} \quad p_i = \dfrac{x_i}{2\mathcal{S}}, \\ M_{m+1}\colon \quad \ell_{m+1} = \dfrac{\mathcal{S}}{N^2} \quad & \text{and} \quad p_{m+1} = \frac{1}{2}. \end{cases}$$

Note that all the numbers are rational with polynomial sizes. We reduce N-partition to the following decision problem: (A) "*Does there exist a schedule with average service time less than or equal to* LB_0?" First, we must calculate LB_0.

According to Corollary 6, the solution $\boldsymbol{\tau}^*$ of the lower bound LB_0 verifies

$$\tau_i^* = \sqrt{\frac{\ell_i}{p_i}\lambda^*} \qquad \text{with} \quad \lambda^* = \left(\sum_{i=1}^{m+1} \sqrt{p_i \ell_i}\right)^2.$$

That is to say

$$\lambda^* = \left(\frac{\sqrt{\mathcal{S}}}{\sqrt{2}N} + \sum_{i=1}^{m} \frac{x_i}{\sqrt{2\mathcal{S}}}\right)^2 = \frac{\mathcal{S}}{2}\left(\frac{1}{N} + \sum_{i=1}^{m} \frac{x_i}{\mathcal{S}}\right)^2 = \frac{\mathcal{S}}{2}\left(\frac{1}{N} + 1\right)^2.$$

Then

$$\tau_{m+1}^* = \sqrt{\frac{2\mathcal{S}}{N^2}\frac{\mathcal{S}}{2}\left(1 + \frac{1}{N}\right)^2} = \frac{\mathcal{S}}{N} + \frac{\mathcal{S}}{N^2},$$

and, for $1 \le i \le m$,

$$\tau_i^* = \sqrt{\frac{2\mathcal{S}}{x_i} x_i \frac{\mathcal{S}}{2}\left(1 + \frac{1}{N}\right)^2} = \mathcal{S} + \frac{\mathcal{S}}{N}.$$

We conclude that

$$\tau_{m+1}^* = \frac{\mathcal{S}}{N} + \ell_{m+1} \quad \text{and} \quad \tau_1^* = \cdots = \tau_m^* = \mathcal{S} + N\ell_{m+1} = N \cdot \tau_{m+1}^*$$

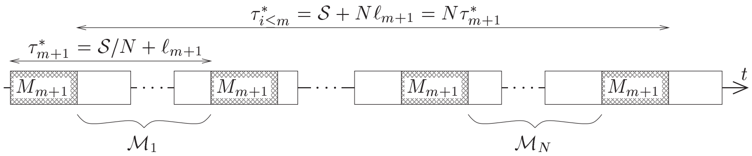

Fig. 3. A schedule that satisfies the decision problem (A).

and the lower bound LB_0 is

$$\mathrm{LB}_0 = \mathbb{E}_p[\ell] + \frac{1}{2}\left(\sum_{i=1}^{m+1}\sqrt{p_i\ell_i}\right)^2 = \frac{\mathcal{S}}{2N^2} + \frac{\sum_{i=1}^m x_i^2}{2\mathcal{S}} + \frac{\mathcal{S}}{4}\left(1+\frac{1}{N}\right)^2.$$

We then reduce N-partition to the following decision problem: (A) "*Does there exist a schedule with average service time less than or equal to* $(\mathcal{S}/2N^2 + (\sum_{i=1}^m x_i^2)/2\mathcal{S} + (\mathcal{S}/4)(1+1/N)^2)$?"

According to Lemma 10, there exists an optimal periodic schedule of the messages $M_1, \ldots, M_m, M_{m+1}$. Thus (A) is equivalent to: (A$'$) "*Does there exist a* periodic *schedule of the messages* $M_1, \ldots, M_m, M_{m+1}$ *with average service time less than or equal to* $(\mathcal{S}/2N^2 + (\sum_{i=1}^m x_i^2)/2\mathcal{S} + (\mathcal{S}/4)(1+1/N)^2)$?" According to Lemma 7, the answer is "yes" if and only if there exists a periodic schedule S that broadcasts M_{m+1} exactly every $\tau^*_{m+1} = \ell_{m+1} + \mathcal{S}/N$, and each $M_{i\le m}$ exactly every $\tau^*_i = (N\ell_{m+1} + \mathcal{S}) = N \cdot \tau^*_{m+1}$. Such a schedule exists if and only if one can partition $\{M_1, \ldots, M_m\}$ into N sets $\mathcal{M}_1, \ldots, \mathcal{M}_N$ such that (see Figure 3)

$$\sum_{M_i\in\mathcal{M}_1} \ell_i = \cdots = \sum_{M_i\in\mathcal{M}_N} \ell_i. \qquad \square$$

6. 3-Approximation Algorithms

6.1. *A New Lower Bound.* We present here a new lower bound LB_α for the data broadcast with non-uniform transmission times. The analysis of the algorithms will show in the next section that this lower bound is indeed tight.

Upon closer examination, one realizes that the cost of any schedule for the set of messages in Fact 8 is unavoidably high because whenever a really long message is broadcast, all incoming requests have to wait. This observation led us to the following improved lower bound. We will see in Theorem 17 that this improved lower bound is tight up to a constant factor.

We define the quantity LB_α as the following minimization problem, for $0 \le \alpha < 1$:

$$\mathrm{LB}_\alpha \begin{cases} \mathbb{E}_p[\ell] + \min\limits_{\boldsymbol{\tau}>\mathbf{0}} \sum\limits_{i=1}^m \left((1-\alpha)p_i\dfrac{\tau_i}{2} + \alpha\dfrac{\ell_i^2}{2\tau_i} + \dfrac{c_i}{\tau_i}\right) \\ \text{subject to} \quad \sum\limits_{i=1}^m \dfrac{\ell_i}{\tau_i} \le 1. \end{cases}$$

Note that for $\alpha = 0$, we obtain LB_0.

As before this minimization problem is solved as follows:

LEMMA 12. *For* $0 \le \alpha < 1$, *the minimization problem* LB_α *admits a unique solution* $\tau^\circledast$ *verifying*

$$\tau_i^\circledast = \sqrt{\frac{2c_i + \alpha\ell_i^2 + \lambda^\circledast \ell_i}{(1-\alpha)p_i}}$$

for a certain $\lambda^\circledast \ge 0$ *(independent of* i*), such that* $\lambda^\circledast = 0$ *if* $\sum_{i=1}^m \ell_i \sqrt{(1-\alpha)p_i/(2c_i+\alpha\ell_i^2)} \le 1$; *otherwise,* $\lambda^\circledast$ *is the unique positive solution to* $\sum_{i=1}^m \ell_i \sqrt{(1-\alpha)p_i/(2c_i+\alpha\ell_i^2+\lambda^\circledast\ell_i)} = 1$.

PROOF. Straightforward application of Lemma 5 with the modified cost $c_i' = (\alpha\ell_i^2 + 2c_i)/2(1-\alpha)$. □

PROPOSITION 13. *For any* $0 \le \alpha < 1$, *the minimization problem* LB_α *is a lower bound to the cost of any schedule of* $M_1, \ldots, M_m$.

PROOF. Thanks to Lemma 4, we restrict ourself to periodic schedules. Let S be a periodic schedule. We denote by T its period, and by n_i the number of broadcasts of M_i during a period. We define $\tau_i = T/n_i$. We know from (4) that

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

We show here another lower bound on $\mathrm{COST}(S)$.

Since every message M_i is broadcast n_i times in a period of S, the average broadcast cost for S is $\mathcal{A}\mathrm{BC}(S) = \sum_{i=1}^m (n_i c_i/T) = \sum_{i=1}^m (c_i/\tau_i)$.

Consider now a random request for message M_j in S. With probability $n_i\ell_i/T = \ell_i/\tau_i$, it arrives during a broadcast of M_i and has to wait until the end of the current broadcast, i.e., $\ell_i/2$ on average, before starting to download anything. The downloading time of the requested message is ℓ_j. Summing over i, we get that the average service time for the request is at least

$$\ell_j + \sum_{i=1}^m \frac{\ell_i}{\tau_i} \cdot \frac{\ell_i}{2} = \ell_j + \sum_{i=1}^m \frac{\ell_i^2}{2\tau_i}.$$

Summing over j, we get that the service time is at least

$$\mathcal{A}\mathrm{ST}(S) \ge \mathbb{E}_p[\ell] + \sum_{i=1}^m \frac{\ell_i^2}{2\tau_i}.$$

Summing the lower bounds on the service time and broadcast cost, we get the following lower bound on the cost of S:

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

A linear $(1-\alpha, \alpha)$ combination with (4) then gives that LB_α is also a lower bound on the cost of S. □

The analysis of our randomized algorithm in the next section shows that this lower bound is indeed tight.

6.2. *A Randomized* 3*-Approximation*. According to the lower bound LB_α, one should try to broadcast the messages M_i regularly proportionally to $\tau_i^{\circledast}$. A classical method in designing randomized algorithms (see [15], [5], [8], and [22] in the data broadcasting context) is to choose randomly to broadcast message M_i with probability proportional to $1/\tau_i^{\circledast}$. We show in this subsection that this leads to a 3-approximation.

Lemma 14 analyzes the expected performance of an algorithm that broadcasts each message M_i with some probability s_i and stays idle for a period of time ℓ_0 with probability s_0. Note that the idle periods are necessary: when broadcast costs are very high, it is worth, from time to time, not to broadcast any messages. Theorem 15 concludes that this algorithm is a 3-approximation when (s_i) and ℓ_0 are chosen as in the preprocessing step.

LEMMA 14. *The randomized schedule output by Algorithm* 1 *has expected cost*

$$\mathbb{E}[\mathrm{COST}(S)] = \frac{\mathbb{E}_s[\ell^2]}{2\mathbb{E}_s[\ell]} + \mathbb{E}_p\left[\frac{1}{s}\right] \cdot \mathbb{E}_s[\ell] + \frac{\mathbb{E}_s[c]}{\mathbb{E}_s[\ell]}.$$

Algorithm 1. A randomized scheduler

INPUT:

1. m messages $M_1, \ldots, M_m$ with popularities $(p_i)_{i=1,\ldots,m}$, with lengths $(\ell_i)_{i=1,\ldots,m}$ and broadcast costs $(c_i)_{i=1,\ldots,m}$.

PREPROCESSING:

Consider $\alpha = \frac{1}{3}$ and $\tau^{\circledast}$ the unique solution to LB_α:

$$\mathrm{LB}_\alpha \begin{cases} \mathbb{E}_p[\ell] + \min\limits_{\boldsymbol{\tau}>\mathbf{0}} \sum\limits_{i=1}^{m} \left((1-\alpha) p_i \dfrac{\tau_i}{2} + \alpha \dfrac{\ell_i^2}{2\tau_i} + \dfrac{c_i}{\tau_i} \right) \\ \text{subject to} \quad \sum\limits_{i=1}^{m} \dfrac{\ell_i}{\tau_i} \leq 1 \end{cases}$$

as presented in Lemma 12. Take $\ell_0 = \mathbb{E}_p[\ell]$. Define $\tau_0^{\circledast}$ by $\ell_0/\tau_0^{\circledast} = 1 - \sum_{i=1}^{m} (\ell_i/\tau_i^{\circledast})$. For $i = 0, \ldots, m$, take

$$s_i = \frac{1/\tau_i^{\circledast}}{\sum_{j=0}^{m} (1/\tau_j^{\circledast})}.$$

OUTPUT:

loop

Draw $i \in \{0, \ldots, m\}$ with probability s_i.

if $i = 0$ **then**

The server remains idle for an time period of length ℓ_0.

else

The server broadcasts message M_i.

PROOF. We analyze separately the expected average broadcast cost and the expected average service time. Let $N_i(t)$ be the number of broadcasts of message M_i that start during the time period $[0, t]$.

The broadcast cost. By definition, $\mathcal{A}\mathrm{BC}(S, [0, T]) = \sum_{i=1}^{m}(c_i N_i(T)/T)$. By the Borel–Cantelli theorem, $\lim_{T\to\infty}(N_i(T)/T) = s_i/\mathbb{E}_s[\ell]$. The linearity of expectation then implies

$$\mathbb{E}[\mathcal{A}\mathrm{BC}(S)] = \sum_{i=0}^{m} \frac{c_i s_i}{\mathbb{E}_s[\ell]} = \frac{\mathbb{E}_s[c]}{\mathbb{E}_s[\ell]}.$$

The service time. Consider a request Q for M_i arriving at time t. Its service time $\mathrm{ST}(S, M_i, t)$ is

$$\mathrm{ST}(S, M_i, t) = Y(t) + Z_i + \ell_i,$$

where $Y(t)$ is the time elapsed from t to the end of the current broadcast at time t, and Z_i is the time elapsed from $t + Y(t)$ to the beginning of the next broadcast of M_i. Note that $Y(t)$ is independent of i and Z_i is independent of t.

First we compute $\mathbb{E}[Y(t)]$. From the Borel–Cantelli theorem, the scheduler spends on expectation an $s_j\ell_j/\mathbb{E}_s[\ell]$ fraction of the time broadcasting M_j. Thus, with probability $s_j\ell_j/\mathbb{E}_s[\ell]$, request Q arises during a broadcast of message M_j and then waits $\ell_j/2$ on expectation. Thus,

$$\mathbb{E}[Y(t)] = \sum_{j=0}^{m} \frac{s_j\ell_j}{\mathbb{E}_s[\ell]} \cdot \frac{\ell_j}{2} = \frac{\mathbb{E}_s[\ell^2]}{2\,\mathbb{E}_s[\ell]}.$$

Second, to compute $E(Z_i)$, observe that there are on expectation $\sum_{t\geq 0} t s_i(1-s_i)^t = (1-s_i)/s_i$ messages broadcast before M_i is sent, and each of those messages has expected length $(1/(1-s_i))\sum_{j=0,\ j\neq i}^{m} s_j\ell_j$. So,

$$\mathbb{E}[Z_i] = \frac{\sum_{j=0,\ j\neq i}^{m} s_j\ell_j}{s_i}. \tag{5}$$

Finally, the expected service time for request Q (including the downloading time) is

$$\begin{aligned}\mathbb{E}[ST(S, M_i, t)] &= \mathbb{E}[Y(t)] + \mathbb{E}[Z_i] + \ell_i \\ &= \frac{\mathbb{E}_s[\ell^2]}{2\mathbb{E}_s[\ell]} + \frac{\sum_{j=0,\ j\neq i}^{m} s_j\ell_j}{s_i} + \ell_i = \frac{\mathbb{E}_s[\ell^2]}{2\mathbb{E}_s[\ell]} + \frac{1}{s_i}\mathbb{E}_s[\ell].\end{aligned}$$

Summing over all messages (weighted with their respective popularities) and adding the expected broadcast cost concludes the proof of the lemma. □

THEOREM 15. *Algorithm* 1 *is a randomized polynomial* 3*-approximation*.

PROOF. Plugging the values of s_i into the lemma, a straightforward calculation gives the expected cost of the randomized schedule S output by Algorithm 1. Compare this cost with LB_α term by term:

$$\mathbb{E}[\mathrm{COST}(S)] = \sum_{i=1}^{m} p_i \tau_i^{\circledast} + \sum_{i=0}^{m} \frac{\ell_i^2}{2\tau_i^{\circledast}} + \sum_{i=1}^{m} \frac{c_i}{\tau_i^{\circledast}},$$

$$\mathrm{LB}_\alpha = \mathbb{E}_p[\ell] + \sum_{i=1}^{m} \left(\frac{1-\alpha}{2} p_i \tau_i^{\circledast} + \alpha \frac{\ell_i^2}{2\tau_i^{\circledast}} + \frac{c_i}{\tau_i^{\circledast}} \right).$$

We obtain

$$\text{(6)} \qquad \mathbb{E}[\mathrm{COST}(S)] \le \frac{\ell_0}{2} \cdot \frac{\ell_0}{\tau_0^{\circledast}} + \max\left(\frac{2}{1-\alpha}, \frac{1}{\alpha}, 1 \right) \cdot (\mathrm{LB}_\alpha - \mathbb{E}_p[\ell]).$$

Using $\ell_0/\tau_0^{\circledast} \le 1$, $\mathrm{LB}_\alpha \le \mathrm{OPT}$, $\ell_0 = \mathbb{E}_p[\ell]$, and $\alpha = \frac{1}{3}$, we finally get

$$\text{(7)} \qquad \mathbb{E}[\mathrm{COST}(S)] \le 3\,\mathrm{OPT} - \frac{\mathbb{E}_p[\ell]}{2}.$$

Note that the term $-\mathbb{E}_p[\ell]/2$ could be omitted but it will be useful to prove Theorem 17. ☐

As a corollary of the proof, we observe that for any $0 < \alpha < 1$, LB_α is a tight lower bound.

6.3. *Derandomizing*: *a Deterministic Greedy* 3-*Approximation*. The algorithm so far constructs a randomized schedule, which has the inconvenience of not being periodic. Periodicity is a particularly useful property when designing a cache or prefetching strategy [2] as well as indexing to allow power saving by reducing the monitoring time [18]. In this section we derandomize Algorithm 1 into a deterministic algorithm, which we then, in the next section, truncate and wrap around cyclically to produce a deterministic periodic schedule with quadratic length.

Our approach is inspired by the analysis in [8] of the greedy algorithm of [5]. A close look at the potential function used in [8] reveals that the algorithm of [5] could be obtained from the randomized algorithm of [15] by applying the method of conditional expectations. Similarly, we apply the method of conditional expectations to the randomized algorithm in the previous section.

DEFINITION. We denote by $\sigma_i^t(S)$ the time elapsed at time t since the beginning of the last broadcast of M_i before time t in a schedule S (by convention, we assume that $\sigma_i^t(S) = t$ for all time t before the first broadcast of M_i). Let $\sigma^t(S) = (\sigma_i^t(S))$ be the *state* of the schedule at time t.

The following lemma shows that the greedy Algorithm 2 performs at least as well as the randomized Algorithm 1.

Algorithm 2. A greedy scheduler

INPUT: As in Algorithm 1.
PREPROCESSING: As in Algorithm 1.
OUTPUT:
loop
Let σ_i be the time elapsed since the beginning of the last broadcast of M_i.
Choose $i \in \{0, 1, \ldots, m\}$ which minimizes

$$c_i + \frac{\ell_i^2}{2} - p_i \sigma_i \frac{\mathbb{E}_s[\ell]}{s_i} - \ell_i \underbrace{\left\{ \frac{1}{2} \frac{\mathbb{E}_s[\ell^2]}{\mathbb{E}_s[\ell]} + \frac{\mathbb{E}_s[c]}{\mathbb{E}_s[\ell]} - \sum_{j=1}^{m} p_j \sigma_j \right\}}_{\text{independent of } i}$$

if $i = 0$ **then**
The server remains idle for a time period ℓ_0.
else
The server broadcasts message M_i.

LEMMA 16 (Greedy Algorithm). *Algorithm* 2 *outputs a schedule S whose cost is at most the expected cost of Algorithm* 1, *given in Lemma* 14.

NOTE 1. When all the lengths are equal to 1, Algorithm 2 is the algorithm of [8].

PROOF. Let M_{g_n} denote the nth message broadcast, where by convention M_0 denotes an idle period of length ℓ_0. We use the penalty-based definition of the cost presented in Section 2.3. According to (1) and Proposition 3,

$$\text{COST}(S) = \widehat{\text{COST}}(S) = \limsup_{N \to \infty} \frac{\widehat{\text{cost}}(1, S) + \cdots + \widehat{\text{cost}}(N, S)}{\ell_{g_1} + \cdots + \ell_{g_N}}, \tag{8}$$

where $\widehat{\text{cost}}(n, S)$ is the sum of $\ell_{g_n} \mathbb{E}_p[\ell]$ and of the cumulated penalty and broadcast costs of a schedule S during the broadcast of M_{g_n}.

Let $\sigma^n(S)$ denote the state at the end of the nth broadcast in S. Our analysis uses a potential function defined by

$$\Phi(n, S) =_{\text{def}} \sum_{j=1}^{m} p_j \sigma_j^n(S) \left(\frac{\mathbb{E}_s[\ell]}{s_j} - \ell_j \right). \tag{9}$$

The definition of Φ generalizes the potential function in [8] using the following intuition: $\Phi(n, S)$ is exactly the expected wait of the requests still unserved at the end of the nth broadcast in S *if* the schedule uses randomized Algorithm 1 after the nth broadcast. A request for M_j would then wait $\sum_{k=0, k\neq j}^{m} (\ell_k s_k / s_j) = \mathbb{E}_s[\ell]/s_j - \ell_j$ in expectation (see (5) in the proof of Lemma 14). Φ thus takes into account the future cost which is not part of the penalty.

Let Avg denote the expected cost of the randomized algorithm, as given in Lemma 14. We will show that the selection rule ensures that at every step n that broadcasts M_i,

$$\widehat{\text{cost}}(n, S) - \Phi(n-1, S) + \Phi(n, S) \leq \ell_i \, \mathsf{Avg}. \tag{10}$$

Summing over n and dividing by the sum of the lengths then yields

$$\frac{\sum_{n=1}^{N} \widehat{\text{cost}}(n, S) + \Phi(N, S) - \Phi(0, S)}{\ell_{g_1} + \cdots + \ell_{g_N}} \leq \mathsf{Avg},$$

which concludes the proof since $\Phi(N, S)$ is always non-negative and $\Phi(0, S) = 0$.

We now concentrate on proving (10). Let S_i be the schedule which is identical to S for the first $(n-1)$ steps but broadcasts message M_i during the nth step. We show two facts: first, that (10) holds on average when i is selected according to distribution s; and, second, that the greedy selection rule selects the i that minimizes the difference between the left- and the right-hand sides in (10), thus ensuring that (10) holds.

In S_i we have $\sigma_j^n(S_i) = \sigma_j^{n-1}(S) + \ell_i$ for $j \neq i$, and $\sigma_i^n(S_i) = \ell_i$. A simple calculation then gives

$$\Phi(n, S_i) - \Phi(n-1, S_i) = \ell_i \sum_{j=1}^{m} p_j \left(\frac{\mathbb{E}_s[\ell]}{s_j} - \ell_j \right) - p_i \sigma_i^{n-1}(S) \left(\frac{\mathbb{E}_s[\ell]}{s_i} - \ell_i \right).$$

Moreover, the average penalty for M_j during the broadcast of M_i is $p_j(\sigma_j^{n-1}(S) + \ell_i/2)$ if $j \neq i$, and $p_i(\ell_i/2)$ if $j = i$. Thus,

$$\widehat{\text{cost}}(n, S_i) = c_i + \frac{\ell_i^2}{2} + \ell_i \sum_{j=1, j\neq i}^{m} p_j \sigma_j^{n-1}(S) + \ell_i \mathbb{E}_p[\ell].$$

Thus, the left-hand side of (10) equals

$$\text{(11)} \quad \text{L.H.S.}(10) = c_i + \frac{\ell_i^2}{2} + \ell_i \sum_{j=1}^{m} p_j \sigma_j^{n-1}(S) + \ell_i \mathbb{E}_p\left[\frac{1}{s}\right] \cdot \mathbb{E}_s[\ell] - p_i \sigma_i^{n-1}(S) \frac{\mathbb{E}_s[\ell]}{s_i}.$$

Taking the expectation over i gives

$$\mathbb{E}_s[\text{L.H.S.}(10)] = \mathbb{E}_s[c] + \tfrac{1}{2}\mathbb{E}_s[\ell^2] + \mathbb{E}_p\left[\frac{1}{s}\right] \cdot \mathbb{E}_s[\ell]^2 = \mathbb{E}_s[\ell]\,\mathsf{Avg},$$

which concludes the first fact. Second, using (11) and the expression of Avg given in Lemma 14, note that the difference between the left- and the right-hand sides in (10) can be rewritten as

$$-p_i \sigma_i^{n-1}(S) \frac{\mathbb{E}_s[\ell]}{s_i} + c_i + \frac{\ell_i^2}{2} - \ell_i \left\{ \frac{\mathbb{E}_s[\ell^2]}{2\mathbb{E}_s[\ell]} + \frac{\mathbb{E}_s[c]}{\mathbb{E}_s[\ell]} - \sum_{j=1}^{m} p_j \sigma_j^{n-1}(S) \right\}.$$

The greedy rule selects at each step the message M_i that minimizes this quantity which ensures that (10) holds at each step. □

6.4. *Truncating*: *a Deterministic Periodic* 3-*Approximation*. In this section we prove that one can stop the greedy algorithm after a quadratic number of steps and obtain a periodic 3-approximation by wrapping the schedule around.

The construction is illustrated in Figure 4. Algorithm 3 defines a periodic schedule of period $T + \mathcal{L}$, with $T_0 \leq T < T_0 + \ell_{\max}$.

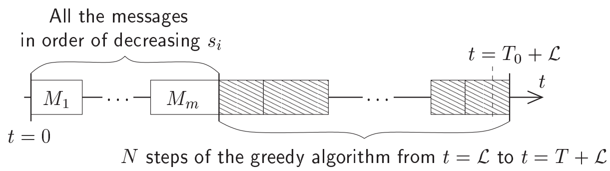

Fig. 4. The periodic approximation.

THEOREM 17 (Deterministic and Periodic Approximation). *Algorithm* 3 *outputs a periodic schedule S with period* $\leq 2(\mathcal{L}^2 + \mathcal{C})/\mathbb{E}_p[\ell] + \ell_{\max}$ *and with cost* $\leq 3\,\mathrm{OPT}$.

PROOF. We adopt the same notations ($\sigma^n(S), \Phi(n, S), \ldots$) as in the proof of Lemma 16. As in that lemma, we use the penalty-based definition of the cost of S (see Section 2.3). S has period $(T + \mathcal{L})$.

Analysis of the time interval $[\mathcal{L}, T + \mathcal{L}]$. According to (10), the cumulated cost of S for this time interval is bounded by

$$T\,\mathsf{Avg} + \Phi(m, S) - \Phi(N + m, S), \tag{12}$$

where N is the number of broadcasts in S during that time interval.

Analysis of the time interval $[0, \mathcal{L}]$. Every message is broadcast once during this period, the total broadcast cost for this period is then clearly $\mathcal{C}$. If t_i denotes the time of the first broadcast of M_i, then the cumulated penalty of S during $[0, \mathcal{L}]$ is

$$\sum_{i=1}^{m} p_i \left(t_i \left(\frac{t_i}{2} + \sigma_i^{N+m}(S) \right) + (\mathcal{L} - t_i) \frac{(\mathcal{L} - t_i)}{2} \right) \leq \sum_{i=1}^{m} p_i t_i \sigma_i^{N+m}(S) + \mathcal{L}^2.$$

Since the messages are broadcast by order of decreasing s_i, t_i is less than the expected time, $(\mathbb{E}_s[\ell]/s_i - \ell_i)$, of the first broadcast of M_i in Algorithm 1, so, $t_i \leq \mathbb{E}_s[\ell]/s_i - \ell_i$. Recalling the definition of Φ, (9), we get

$$\Phi(N + m, S) \geq \sum_{i=1}^{m} p_i t_i \sigma_i^{N+m}(S).$$

Adding the cumulated costs during the two time intervals and using (8), we obtain that the total cost satisfies

$$(T + \mathcal{L})\,\mathrm{COST}(S) \leq T\,\mathsf{Avg} + \Phi(m, S) + \mathcal{L}^2 + \mathcal{C}.$$

Algorithm 3. A periodic scheduler

INPUT: As in Algorithms 1 and 2.
PREPROCESSING: As in Algorithms 1 and 2. Let $T_0 = 2(\mathcal{L}^2 + \mathcal{C})/\mathbb{E}_p[\ell] - \mathcal{L}$.
OUTPUT: The first period of the schedule is as follows.

1. Broadcast each message M_i once ($i = 1, \ldots, m$), in order of decreasing s_i.
2. Run Algorithm 2 until time $T_0 + \mathcal{L}$, then finish the current broadcast.

Now, $\Phi(m, S)$ is the expected future waiting time of requests still unserved at the beginning of the period, if one was to continue with the randomized algorithm. All the requests arose in the past $\mathcal{L}$ time, so $\Phi(m, S) \le \mathcal{L}\,\mathsf{Avg}$. Thus,

$$\mathrm{COST}(S) \le \mathsf{Avg} + \frac{\mathcal{L}^2 + \mathcal{C}}{T + \mathcal{L}}.$$

From our definition of T_0, $T + \mathcal{L} \ge T_0 + \mathcal{L} \ge 2(\mathcal{L}^2 + \mathcal{C})/\mathbb{E}_p[\ell]$. Recalling from (7) that $\mathsf{Avg} \le 3\,\mathrm{OPT} - \mathbb{E}_p[\ell]/2$, we conclude that

$$\mathrm{COST}(S) \le 3\,\mathrm{OPT}.$$

Time complexity (*number of broadcast steps*). The computation of $\tau^{\circledast}$ is discussed in the next section. Sorting the s_i's takes $O(m \log m)$ time. The time complexity here is bounded by the number of steps, N, of the greedy algorithm. Time increases at each step by at least $\ell_{\min}$, so the schedule is computed after $N \le 2(\mathcal{L}^2 + \mathcal{C})/\mathbb{E}_p[\ell]\ell_{\min}$ steps (i.e., $O(m^2)$ steps if lengths and broadcast costs are constant). □

6.5. *Numerical Concerns*. From an implementation viewpoint, the solution to LB_α (and LB_0) in Lemma 12 (and 5) cannot in general be calculated exactly. Solving the minimization problem LB_α reduces to finding the unique positive solution $\lambda^{\circledast}$ to

$$\sum_{i=1}^{m} \ell_i \sqrt{\frac{a_i}{b_i + \lambda \ell_i}} = 1,$$

where $\ell_i > 0$, $0 < a_i \le 1$, and $b_i > 0$. Since $\lambda \mapsto \sum_{i=1}^{m} \sqrt{a_i/(b_i + \lambda \ell_i)}$ is a convex decreasing function, Newton's algorithm is an efficient solution. In order to bound Newton's algorithm convergence time, we just need to bound $\lambda^{\circledast}$. We have

$$\frac{1}{\sqrt{\lambda^{\circledast} + \min_i (b_i/\ell_i)}} \sum_{i=1}^{m} \sqrt{a_i \ell_i} \ge 1.$$

Since $a_i \le 1$ and $b_i \ge 0$, we get

$$0 \le \lambda^{\circledast} \le \left(\sum_{i=1}^{m} \sqrt{\ell_i} \right)^2 - \min_i \frac{b_i}{\ell_i} \le m\mathcal{L}.$$

Thus Newton's algorithm computes an approximation of $\lambda^{\circledast}$ within ε accuracy from the seed $\lambda = 0$ in time $O(\log \log((1/\varepsilon)\mathcal{L}m))$. We can compute in sublogarithmic time an approximation of $\tau^{\circledast}$ within ε accuracy. Since the optimal τ_i^* are bounded away from zero, the cost given in (6) does not increase much under a perturbation of τ near the optimal solution.

7. A Multiple Channels Framework. The problem as stated since the beginning of this paper can be seen a *single channel* data broadcast: the server can only broadcast one message at any time. One can naturally extend the definitions to a multiple channels framework, as in [8], where the scheduler has W channels available for broadcasting the messages:

- *A schedule* S of $M_1, \ldots, M_m$ on W channels is, formally, a set of ordered pairs $(s_w(n), t_w(n))$, where $1 \leq w \leq W$. For any $n \geq 1$, we define $s_w(n) = i$ if M_i is the nth message broadcast on channel w, and the starting time of this broadcast is $t_w(n)$. Since channel w must finish broadcasting a message before it can start broadcasting any other message, we have the constraint

$$(\forall w)\ (\forall n) \qquad t_w(n+1) \geq t_w(n) + \ell_{s_w(n)}.$$

  S is *periodic* if there is a $T > 0$ (the period) such that for every channel w, there exists $N_w > 0$ such that for any $n \geq 1$, $s_w(n + N_w) = s_w(n)$ and $t_w(n + N_w) = t_w(n) + T$.
- *The clients* connect at a uniform random time (given by some Poisson process[7]), ask for a random message M_i (according to the distribution p_i), monitor all the channels simultaneously, and download the requested message as soon as it is broadcast on some channel (clients cannot start downloading in the middle of the broadcast of a message but have to wait until the next broadcast).
- *The cost* is the sum of the average service time (average monitoring time plus average downloading time) and of the average broadcast cost.

The multiple channel problem is not well understood. Schabanel [19] shows that finding an optimal periodic schedule is *NP*-hard. However, no approximation algorithm with constant performance guarantee is known at this point. The lower bound techniques used in Section 6.1 do not seem to work.

We now discuss in some detail some of the difficulties. As we will see the multiple channel case with non-uniform transmission time differs radically from the uniform case studied in [8].

7.1. *What Is Still True*?

Expression and Reversibility of the Cost. The algebraic expression of the cost given in Lemma 1 only relies on the time between two consecutive broadcasts of the same messages. In the multiple channel framework, the two consecutive broadcasts may not occur on the same channel, as can be seen in Figure 5, but in [19] it is shown that the cost still has the same expression. Similarly, in [19] it is shown that the reversibility lemma (Proposition 3) also holds in the multiple channel framework.

The Lower Bound LB_0 *Is Not Tight.* As shown in [19], the lower bound LB_0 extends naturally to a lower bound on W channels, by increasing the available bandwidth from 1 to W. We get

$$\mathrm{LB}_0 \begin{cases} \mathbb{E}_p[\ell] + \min\limits_{\boldsymbol{\tau} > \mathbf{0}} \displaystyle\sum_{i=1}^{m} \left(p_i \frac{\tau_i}{2} + \frac{c_i}{\tau_i} \right) \\ \text{subject to} \quad \displaystyle\sum_{i=1}^{m} \frac{\ell_i}{\tau_i} \leq W. \end{cases}$$

[7] See footnote 4.

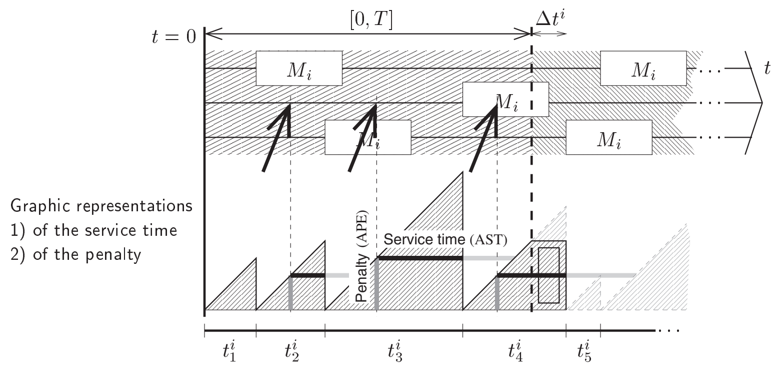

Fig. 5. The cost has the same representation in the multiple channel framework.

As in the one channel case, this lower bound is not tight. We extend the example in Fact 8 as follows: consider $W + 1$ messages $M_1, \ldots, M_{W+1}$ where M_i has length $\ell_i = L^{i-1}$ and popularity $p_i = \alpha/L^{i-1}$ where α is chosen so that $p_1 + \cdots + p_{W+1} = 1$.

FACT 18 [19]. *For the set of messages given above, the optimal schedule on W channels has cost* $\mathrm{OPT} = \Theta(L^{2^{-W}})$ *but* $\mathrm{LB}_0 = \Theta(1)$ *when L goes to infinity.*

PROOF SKETCH. A simple adaptation of Lemma 5 gives $\mathrm{LB}_0 = \Theta(1)$. However, the optimal schedule on W channels has cost $\Theta(L^{2^{-W}}) \gg \mathrm{LB}_0$. It is a periodic schedule with period $L^{W+2^{-W}}$ and constructed recursively as follows: between time $t = 0$ and $t = L^W$, M_{W+1} is broadcast on channel W and $M_1, \ldots, M_W$ are broadcast on the first $W - 1$ channels according to the optimal schedule on $W - 1$ channels; and between time $t = L^W$ and $t = L^{W+2^{-W}}$, messages $M_1, \ldots, M_W$ are broadcast continuously each on one of the W channels. The complete proof can be found in [19]. □

NP-Hardness. The reduction from N-partition given in the proof of Theorem 11 can be extended by adding $W - 1$ messages with popularities and lengths adjusted so that each would be broadcast continuously on one of the channels $2, \ldots, W$ according to LB_0. Thus, finding the optimal cost of a periodic schedule on W channels is strongly *NP*-hard [19].

7.2. *What Is Not True Anymore*?

Lower Bounds. The techniques used to design our lower bound do not work when broadcasting on several channels. LB_α ($\alpha > 0$) is not a lower bound for the multiple channel case: it is no longer true that no request can be served during the broadcast of a message. Unlike in the uniform setting in [8], the optimal cost on W channels depends non-trivially on W. As can be seen with the example of Fact 18, adding a channel can reduce the optimal cost by an arbitrary factor:[8] on W channels the optimal

[8] In the uniform setting only a factor of $W/(W + 1)$ could be saved.

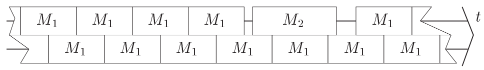

Fig. 6. An optimal schedule.

cost for the $W + 1$ messages is $\Theta(L^{2^{-W}})$, which tends to infinity when L grows; but with $W + 1$ channels, the schedule that broadcasts M_i continuously on channel i has cost $\Theta(1)$. As far as we know, no tight lower bound is known for the multiple channel setting.

Periodicity of an Optimal Schedule. We do not know which condition would guarantee the existence of an optimal periodic schedule when broadcasting on several channels.

First, there does not necessarily exist a periodic optimal schedule when there are several channels.[9] Indeed, consider the scheduling problem on two channels of two messages M_1 and M_2, of lengths $\ell_1 = 1$ and $\ell_2 = \sqrt{2}$, and popularities $p_1 = \sqrt{2}/(1 + \sqrt{2})$ and $p_2 = 1/(1+\sqrt{2})$, and zero costs. A direct modification of Lemma 7 (Section 4) gives that $\mathrm{LB}_0(M_1, M_2) = \sqrt{2}/2(1 + \sqrt{2})$ is a lower bound to the cost of any two-channels schedule of M_1 and M_2. Note that it is realized by the schedule that keeps broadcasting M_1 on the first channel and M_2 on the second.

However, if there was a periodic optimal schedule, it should, according to Lemma 7, broadcast M_1 every $\tau_1^* = 1$ time and M_2 every $\tau_2^* = \sqrt{2}$ time: this is impossible since $1/\sqrt{2}$ is irrational. This example relies on irrationality and may appear artificial (in particular, the periodicity condition is verified on each individual channel). However, it shows that our proof techniques in Lemma 10 (based on the construction of a finite graph) do not work in the multiple channel framework. We do not know whether having only rational lengths implies the existence of an optimal periodic schedule.

Second, when broadcasting on several channels, we cannot assume that an optimal schedule never stays idle, even if the messages have zero broadcast cost (as in Section 5.1). Indeed, consider the broadcast of the following two messages M_1 and M_2 on two channels: $\ell_1 = 2$ and $p_1 = 1-\varepsilon$; $\ell_2 = 3$ and $p_2 = \varepsilon$. For ε small enough, M_2 will be broadcast rarely, and M_1 is the important message. Between two consecutive broadcasts of M_2, M_1 has to be broadcast according to an optimal pattern, that is to say, all the time, on both channels with an offset of one unit of time between the channels (see Figure 6). Suppose that we are given a schedule that never stays idle: after each broadcast of M_2 on some channel, either the pattern for M_1 is broken inducing a huge increase of the cost, or M_2 is broadcast twice in a row delaying the resuming of the pattern, which increases the cost as well. In fact, none of the schedules which never stay idle is optimal. An optimal schedule stays idle for one unit of time around every broadcast of M_2 as shown in Figure 6 (see [19]).

The condition of the existence of an optimal solution in the multiple setting is open. As a consequence, it is not even clear a priori whether an optimal solution can be described in finite time.

[9] In the uniform setting, it is true that there is a periodic optimal schedule.

8. Conclusion. As far as we know this work is the first yielding a constant factor approximation for the non-uniform length case (on one channel). The multiple channel case is still not well understood. Finding a good (tight) lower bound for the multiple channel case with non-uniform length messages is an interesting open question.

In [20] Schabanel introduces a preemptive model for the non-uniform length broadcast problem. It turns out that the problem with preemption is much better understood, even in a multiple channel setting. In particular, it is shown that when preemption is possible, one can reduce the expected service time by an arbitrary factor.

This paper and others focused on minimizing the expected service time, however, one might also want to provide some guarantee in terms of quality of service. For example, an interesting extension of the problem would consist in ensuring that the maximum waiting time is never larger than some bound at least for some "important" messages. Another possible extension would try to schedule at the same time push requests for the most popular messages (based on popularity distribution) and pull requests for less popular messages (based on the real requests of the users for these messages, arriving on-line). Edmonds and Pruhs have recently proposed a first study of this question in [12]. They study a different framework where all the messages are continuously and simultaneously broadcast using the bandwidth of all broadcast channels as a whole; the server decides over time what fraction of the total bandwidth to allocate to each message. They present multiple channel solutions that approximate the single channel optimal cost.

References

[1] A. Acharya, M. Franklin, and S. Zdonik. Balancing push and pull for data broadcast. In *Proc. of the ACM International Conf. on Management of Data* (*SIGMOD* '97), Tucson, AZ, pages 183–194, May 1997. Available at www.cs.brown.edu/research/ddg.

[2] S. Acharya. Broadcast Disks: Dissemination-Based Management for Assymmetric Communication Environments. Ph.D. thesis, Brown University, May 1998. Available at www.bell-labs.com/∼acharya.

[3] S. Acharya, R. Alonso, M. J. Franklin, and S. Zdonik. Broadcast disks: data management for asymmetric communications environment. In *Proc. of the ACM International Conf. Management of Data* (*SIGMOD* '95), Santa Cruz, CA, pages 199–210, June 1995. Available at www.cs.brown.edu/research/ddg.

[4] M. H. Ammar. Response time in a teletext system: an individual user's perspective. *IEEE Trans. Comm.*, COM-35(11):1159–1170, 1987.

[5] M. H. Ammar and J. W. Wong. The design of teletext broadcast cycles. *Perform. Eval.*, 5(4):235–242, 1985.

[6] M. H. Ammar and J. W. Wong. On the optimality of cyclic transmission in teletext systems. *IEEE Trans. Comm.*, COM-35(11):1159–1170, 1987.

[7] S. Anily, C. A. Glass, and R. Hassin. The scheduling of maintenance service. *Discrete Appl. Math.*, 82:27–42, 1998.

[8] A. Bar-Noy, R. Bhatia, J. (S.) Naor, and B. Schieber. Minimizing service and operation costs of periodic scheduling. In *Proc. of the* 9*th Annual ACM–SIAM Symp. on Discrete Algorithms* (*SODA* '98), pages 11–20, 1998.

[9] A. Bar-Noy, A. Nisgav, and B. Patt-Shamir. Nearly optimal perfectly periodic schedules. In *Proc. of the ACM Symp. on Principles of Distributed Computing* (*PODC* 2001), pages 107–116, 2001.

[10] A. Bar-Noy and Y. Shilo. Optimal broadcasting of two files over an asymmetric channel. *J. Parallel Distrib. Comput.*, 60(4):474–493, 2000.

[11] A. Bestavros. Speculative data dissemination and service to reduce server load, network traffic and service time in distributed information systems. In *Proc. of the* 1996 *International Conf. on Data Engineering* (*ICDE* '96), pages 180–189, March 1996. Available at cs-www.bu.edu/∼best.

[12] J. Edmonds and K. Pruhs. Broadcast scheduling: when fairness is fine. In *Proc. of the* 13*th ACM–SIAM Symp. on Discrete Algorithms* (*SODA* 2002), pages 421–432, Jan. 2002. Available at www.cs.pitt.edu/∼kirk/papers.

[13] W. Feller. *An Introduction to Probability Theory*, volume I, 3rd edition. Wiley, New York, 1968.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[15] J. Gecsei. *The Architecture of Videotex Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[16] C. Kenyon and N. Schabanel. The data broadcast problem with non-uniform time. In *Proc. of the* 10*th Symp. on Discrete Algorithms* (*SODA* '99), pages 547–556, Jan. 1999. Available at www.ens-lyon.fr/∼nschaban.

[17] C. Kenyon, N. Schabanel, and N. Young. A polynomial-time approximation scheme for the data broadcast problem. In *Proc. of the* 32*nd ACM Symp. on Theory of Computing* (*STOC* 2000), pages 659–666, Portland, OR, May 2000. Available at www.ens-lyon.fr/∼nschaban.

[18] S. Khanna and S. Zhou. On indexed data broadcast. In *Proc. of the* 30*th ACM Symp. on Theory of Computing* (*STOC* 1998), pages 463–472, 1998.

[19] N. Schabanel. Algorithmes d'approximation pour les télécommunications sans fils: Ordonnancement pour la dissémination de données et allocation statique de fréquences. Ph.D. Thesis n°00ENSL0145, LIP, ENS-Lyon, Jan 2000. Available (in French) at www.ens-lyon.fr/∼nschaban.

[20] N. Schabanel. The data broadcast problem with preemption. In *Proc of the* 17*th International Symp. on Theoretical Computer Science* (*STACS* 2000), pages 181–192, Feb. 2000. LNCS 1770. Springer-Verlag, Berlin. Available at www.ens-lyon.fr/∼nschaban.

[21] C.-J. Su and L. Tassiulas. Broadcast scheduling for information distribution. In *Proc. of INFOCOM*, pages 109–117, 1997.

[22] N. Vaidya and S. Hameed. Efficient algorithms for scheduling single and multiple channel data broadcast. Tech. Report 97-002, Texas A&M University, Feb. 1997.

[23] N. Vaidya and S. Hameed. Log time algorithms for scheduling single and multiple channel data broadcast. In *Proc. of the* 3*rd ACM/IEEE Conf. on Mobile Computing and Networking* (*MOBICOM*), pages 90–99, Sept. 1997.