RC23468 (W0412-040) December 15, 2004
Computer Science

IBM Research Report

Improved Approximation Algorithms for Broadcast Scheduling

Nikhil Bansal, Don Coppersmith, Maxim Sviridenko
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Improved approximation algorithms for Broadcast Scheduling

Nikhil Bansal Don Coppersmith Maxim Sviridenko*

Abstract

We consider two scheduling problems in a broadcast setting. The first problem is that of minimizing the average response time of requests. For the offline version of this problem we give an algorithm with an approximation ratio of $O(\log^2(T+n))$, where n is the total number of pages and T is the time horizon (arrival time of the last request). This substantially improves the previously best known approximation factor of $O(\sqrt{n})$ for the problem [3].

Our second result is for the throughput maximization version of the broadcast scheduling problem. Here each request has a deadline and the goal is to satisfy the maximum number of requests by their deadlines. We give an algorithm with an approximation ratio of $5/6$. This improves the previously best known approximation guarantee of $3/4$ for the problem. [10].

1 Introduction

In this paper we consider two problems in a broadcast setting. The first problem is the problem of minimizing the average response time in *on-demand data broadcasting* systems.

The problem of minimizing the response time is formalized as follows: There is a collection of pages $P = \{p_1, \ldots, p_n\}$. We assume that time is slotted and any page can be broadcast in a *single* time slot. At any time t, the broadcast server receives $n_p(t)$ requests for page p for each $p \in P$. We say that a request $\rho \in \Psi = \{1, \ldots, m\}$ for page $p_{i_\rho} \in P$ that arrives at time t_ρ is satisfied at time $c_{i_\rho}(t_\rho)$, if $c_{i_\rho}(t_\rho)$ is the first time after t_ρ when page p_{i_ρ} is transmitted by the broadcast server. The *response time* of the request $\rho \in \Psi$ is defined to be the time that elapses from its arrival till the time it is satisfied, i.e. $c_{i_\rho}(t_\rho) - t_\rho$. We assume that request ρ arrives in the end of the time slot t_ρ and therefore, it cannot be satisfied in the time-slot in which it arrived, i.e. the response time for any request is at least 1. We want to find a broadcast schedule that minimizes the average response time, defined to be $(\sum_{p\in P}\sum_{t=1}^{T} n_p(t)(c_p(t)-t))/(\sum_{p\in P}\sum_{t=1}^{T} n_p(t))$.

In the second problem every request has a deadline; if a request is not served before its deadline it is lost. The goal is to maximize the total throughput, i.e. the total number of satisfied requests. More formally, every request $\rho \in \Psi$ has three parameters associated with it. The arrival time t_ρ, the page $p_{i_\rho} \in P$ which it requests and the deadline d_ρ, i.e. if page p_ρ is not transmitted during the time interval $(t_\rho, d_\rho]$ the request is lost, otherwise it is satisfied. The goal is to assign one page per time unit during the planning horizon $[0, d_{max}]$ and maximize the total number of satisfied requests.

Previous Work and Our results: We first discuss the average response time problem. Our paper focuses on the offline version of the problem where the request sequence is known in advance to the scheduling algorithm. This problem was shown to be NP hard by Erlebach and Hall [8].

Most of the previous algorithmic work on the problem has focused on resource augmentation where the server is given extra speed compared to the optimal algorithm. These results can be viewed as bicriteria approximation algorithms that compare k-speed approximation algorithm against the performance of an

*IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. E-mail: {bansal,dcopper,sviri}@us.ibm.com.

optimal 1-speed algorithm, where a k-speed algorithm is one that allows a server to broadcast k pages in each time slot. Kalyanasundaram et al. [11] gave a $\frac{1}{\alpha}$-speed, $\frac{1}{1-2\alpha}$-approximation algorithm for any fixed α, for $0 \leq \alpha \leq 1/3$. Gandhi et al. [9] have given a $\frac{1}{\alpha}$-speed, $\frac{1}{1-\alpha}$-approximation algorithm for any $\alpha \in (0, 1/2]$. Erlebach and Hall [8] gave a 6-speed 1-approximation algorithm for the problem, which was improved to a 4-speed 1-approximation algorithm by [9]. Later, Gandhi et al. [10] give a 2-speed, 2-approximation, (which they improved further in their journal version to a 2-speed, 1-approximation). Very recently, Bansal et al. [3] gave an algorithm that works for arbitrary small speed up factor (such algorithms are referred to as fully-scalable algorithms in the scheduling literature [13]). Their algorithm achieved an approximation ratio of $O(1/\epsilon)$ with a $(1+\epsilon)$-speed. Here $(1+\epsilon)$-speed means that the algorithm is allowed to transmit one extra page every $1/\epsilon$ time steps.

In the case where no extra speed is allowed, observe that transmitting the pages in the cyclic order $p_1, \ldots, p_n$ repeatedly is a trivial $O(n)$ approximation. This follows because every request has response time of no more than n in the cyclic schedule above, whereas it has response time of at least 1 in any optimum schedule. For a while this was essentially the best known guarantee, and only very recently Bansal et al. [3] gave an algorithm that achieves an $O(\sqrt{n})$ approximation.

We give here an improved algorithm for the problem in the absence of extra speed that achieves $O(\log^2(T+n))$ approximation, where T is the time at which the last request arrives. In fact our algorithm achieves a somewhat stronger guarantee in that it gives a schedule with average response time of $3 \cdot \text{OPT} + O(\log^2(T+n))$, where OPT denote the value of the optimum solution. We also show that the above algorithm can be modified to yield a slightly better approximation ratio of $O(\log^2(T+n)/\log\log(T+n))$.

The minimum average response time problem has also been studied in the online setting. A lower bound of $\Omega(\sqrt{n})$ without speedup and a lower bound of $\Omega(1/\epsilon)$ with a speedup factor of $(1+\epsilon)$, on the competitive ratio of any randomized online algorithm is known [3]. In [11, Lemma 7], an $\Omega(n)$ lower bound on the competitive ratio of deterministic algorithms is given. Edmonds and Pruhs [6] gave a $(4+\epsilon)$-speed, $O(1+1/\epsilon)$-competitive online algorithm. Later, they [7] showed that a natural algorithm, Longest Wait First, is 6-speed, $O(1)$-competitive. Another measure that has been studied in the literature is minimizing the maximum response time (of a request). For this problem, Bartal and Muthukrishnan [4], gave an $O(1)$-competitive algorithm.

The throughput maximization problem was first studied in [2] in a much more general setting. Bar-Noy et al. [2] designed a $1/2$-approximation algorithm for the problem. Gandhi et al. [10] designed a $3/4$-approximation algorithm for the problem considered in this paper by using depended randomized rounding. By using the deterministic rounding approach from [1] we obtain a $5/6$-approximation algorithm for the throughput maximization version of the broadcast scheduling problem.

2 Preliminaries

We begin by considering an LP relaxation of a natural integer linear program (ILP) for the problem, which is also the starting point of all previously known approximation algorithms for this problem [11, 9, 10, 3][1].

Let $y_{pt'} = 1$ iff page p is broadcast at time t', and let $x_{ptt'} = 1$ iff a request for page p arrived at time t is satisfied at time $t' > t$. Let n_{pt} denote the number of requests for page p arrived at time t. Relaxing the

[1]Strictly speaking, [11] considered a different flow-based LP formulation, however Khuller and Kim [12] showed that this formulation is infact identical to the one considered by [9, 10, 3] and others.

integrality constraints on $x_{ptt'}$ and y_{pt} gives the following linear program.

$$\min \sum_p \sum_t \sum_{t'=t+1}^{T+n} (t'-t) \cdot n_{pt} \cdot x_{ptt'} \tag{1}$$

$$\text{subject to } \; x_{ptt'} \le y_{pt'}, \qquad \forall p, t, t' > t, \tag{2}$$

$$\sum_{t'=t+1}^{T+n} x_{ptt'} \ge 1, \qquad \forall p, t, \tag{3}$$

$$\sum_p y_{pt'} = 1, \qquad \forall t', \tag{4}$$

$$x_{ptt'} \ge 0, \qquad \forall p, t, t', \tag{5}$$

$$y_{pt'} \ge 0, \qquad \forall p, t'. \tag{6}$$

Here T refers to the last time when any request arrives. Observe that it suffices to define variables until time $T+n$, as all requests can be satisfied by time $T+n$ by transmitting page p_i at time $T+i$ for $i = 1, \ldots, n$.

We solve the fractional relaxation of this ILP. The relaxed solution may be viewed as broadcasting pages fractionally at each unit of time such that total fraction of all the pages broadcast in any unit of time is 1. A request for a page p arriving at a time t is considered completely satisfied at time t' if t' is the earliest time such that the total amount of page p broadcast during the interval $(t, t']$ is at least 1.

Given an optimal solution of (1)-(6), we first define some useful notation and properties of this LP solution. We will use $r(p,t)$ to denote the LP cost for the response time for a request for page p that arrives at time t. That is $r(p,t) = \sum_{t'>t} (t'-t) x_{ptt'}$.

It is easy to see that in any optimum solution, $x_{ptt'}$ is completely determined by the values of $y_{pt''}$. In particular, for any p, t and t', if $\sum_{t''=t+1}^{t'} y_{pt''} < 1$, then $x_{ptt'} = y_{pt}$, and if $\sum_{t''=t+1}^{t'} y_{pt''} \ge 1$, then $x_{ptt'} = \max\{0, 1 - \sum_{t''=t+1}^{t'-1} y_{pt''}\}$. Otherwise, the solution to the LP can be improved trivially. This implies that if we define $z_{ptt'} = \max\{0, 1 - \sum_{t''=t+1}^{t'-1} y_{pt''}\}$, then $r(p,t)$ can be expressed as $\sum_{t'>t} z_{ptt'}$. Finally, it is trivial to see that in any optimum solution, each request is completely satisfied by time $T+n$ and moreover we can assume that for any time t, the total amount of page transmitted is exactly 1, that is $\sum_p y_{pt} = 1$ for all $1 \le t \le T+n$.

The following observation about response time is crucial and will be used repeatedly:

Lemma 2.1. *Consider a request for a page p that arrives at time t. For $\alpha \in (0,1]$, let $t(\alpha)$ denote the earliest time after t such that a total of α fraction or more of page p is broadcast during $[t, t(\alpha)]$. Then $\int_0^1 (t(\alpha) - t) d\alpha = r(p,t)$. Equivalently if we choose α uniformly at random in $(0,1]$ and transmit page p at $t(\alpha)$, then the expected response time for this request is LP cost for this request.*

Proof. Since α is chosen uniformly at random in $(0,1]$, the probability that $t(\alpha) \ge t'$, is exactly equal to the probability that $\alpha > \sum_{t''=t+1}^{t'-1} y_{pt''}$ which is exactly equal to $\max\{1 - \sum_{t''=t+1}^{t'-1} y_{pt''}, 0\} = z_{ptt'}$. As, $\int_0^1 (t(\alpha) - t) d\alpha = E[t(\alpha) - t] = \sum_{t'>t} Pr[t(\alpha) \ge t'] = \sum_{t'>t} z_{ptt'} = r(p,t)$, the desired result follows. □

The following lemma is straightforward and will allow us later to consider a discrete version of Lemma 2.1. The proof is deferred to the appendix.

Lemma 2.2. *We can assume that $x_{ptt'}$ and y_{pt} are integral multiples of $\delta = 1/(T+n)^2$. This adds at most 1 to the response time of each request.*

3 Overview of Techniques

The structure of our algorithm is similar to that of the $O(\sqrt{n})$ approximation algorithm of [3].

We first obtain a *tentative* schedule which may be an invalid schedule in the sense that it is allowed to violate the capacity constraints (4) in the ILP. In particular, this schedule may transmit multiple pages at a time step. Suppose this tentative schedule has the following additional properties:

1. The total response time for this schedule is at most $c = O(1)$ times the cost of ILP.

2. The capacity constraints are satisfied approximately in the following sense. For any interval of time $(t, t']$, the number of pages broadcast by the tentative schedule in this interval is no more than $t - t' + b$, for some fixed b. We refer to this b as the *backlog* of the tentative schedule.

In this case, the tentative schedule can be transformed into a valid schedule as follows: We transmit pages in the same order as the tentative schedule while ensuring that no page is transmitted at an earlier time than in the tentative schedule. It is not hard to see that the backlog property ensures that no page is transmitted more than b steps later than in the tentative schedule (See Lemma 4.6 for a formal proof). This implies a solution with average response time $c \cdot \text{OPT} + b$. In the algorithm of [3], c was 1 and b was $O(\sqrt{n})$.

Our improved approximation is based on two new ideas:

First, we use some global information of the LP solution to construct a tentative solution. Prior to our work, all algorithms were local in the following sense: Given a solution to the LP above, they produce a schedule (or a tentative schedule) that ensures that for each request r_p^t, it is satisfied at some time slot t' such that the cumulative amount of page p transmitted by the LP solution during the interval $(t, t']$ is no more than 1. The main reason for this is that the response time for each request in the integral schedule can be charged to its response time in the LP solution.

Note that one consequence of having a local schedule is that if a page p is broadcast at time t, then the next broadcast of page p must be at a time t' such that cumulative amount of page p transmitted by the LP solution during $(t, t']$ is no more than 1. Interestingly, the example described in detail in the Appendix (Section 7) shows that it is unlikely an algorithm based on a local schedule can achieve an approximation ratio better than $O(\sqrt{n})$.

We get around this problem by adopting a more global approach. We show that for each page p, we can partition the time horizon, $1, \ldots, T + n$, into intervals $B(p, i)$, such that the cumulative amount of page p transmitted in $B(p, i)$ is $O(\log(T + n))$, and that the tentative schedule for each $B(p, i)$ can be chosen independently of the other blocks. While this could lead to requests not being satisfied by a transmission locally, we show that our choice of $B(p, i)$ allows us to charge their response time to $O(1)$ times their cost in ILP.

The second part of the algorithm is to give a scheme to choose the tentative schedule for each $B(p, i)$ in such a way that when all the tentative schedules for each $B(p, i)$ are combined together, the backlog at any time step is bounded by $O(\log^2(T + n))$. To do this, we define a linear program where the variables correspond to the possible tentative schedules that can be chosen for each $B(p, i)$. This program has the property that there is feasible (fractional) solution where the backlog is 0 and the response time is only $O(1)$ times that of ILP. The main idea then is to solve a sequence of linear programs, where we successively relax the constraints in this linear program such that the number of fractionally set variables decrease geometrically at each step, the cost of the objective function does not increase and the increase in backlog is bounded during each iteration of relaxing the constraints. In the end we obtain a tentative schedule with backlog $O(\log^2(T + n))$.

4 Minimizing Average Response Time

Our algorithm begins by solving the LP (1)-(6). Let $r(p,t)$ denote the response time according to the LP solution for a request for page p that at time t. Let $c(p,t,t')$ denote the cumulative amount of page transmitted by the LP solution during the time interval $(t,t']$, that is $c(p,t,t') = \sum_{t''=t+1}^{t'} y_{pt''}$. We now define the key concepts of blocks and p-good time points.

4.1 Blocks and p-good time points

Definition 1. *Let $r(p,t)$ denote the response times as determined by the solution to the linear program (1)-(6). For a page p we call a time point t to be p-good if $r(p,t) \leq 2r(p,\tau)$ for all $\tau < t$ such that $c(p,\tau,t) \leq 1$.*

The following lemma shows that a p-good point can be found in any interval of time that broadcasts a sufficient amount of page p.

Lemma 4.1. *Any time interval $(t,t']$ such that $c(p,t,t') > \log(T+n)$ contains a p-good point.*

Proof. Suppose all the points in $(t,t']$ are not p-good. Since, t' is not p-good, there exists a $t_1 < t'$ such that $c(p,t_1,t') \leq 1$ and $r(p,t_1) > 2r(p,t')$. Note that t_1 lies in the interval (t,t'), and hence is not p-good by our assumption. Thus there exists t_2 such that $c(p,t_2,t_1) \leq 1$ and $r(p,t_2) > 2r(p,t')$. Repeating the argument for $\log(T+n)$ steps, we obtain a sequence of points $t < t_{\log(T+n)} < \cdots < t_2 < t_1 < t'$ such that $c(p,t_{\log(T+n)},t') \leq \log(T+n)$ and $r(p,t_{\log(T+n)}) > 2^{\log(T+n)} r(p,t')$, which is impossible as the response time for any request is bounded between 1 and $(T+n)$. $\square$

Lemma 4.1 implies that if t is a p-good point, and such that $c(p,t,T+n) > \log(T+n)$, then there is another p-good point $t' > t$ such that $c(p,t,t') \leq \log(T+n)$. Thus, for each p we can form a collection $G(p) = \{0 = t(p,0), t(p,1), t(p,2), \ldots, t(p,b_p) = (T+n)\}$ of time points such that the $t(p,i)$ is p-good for $1 \leq i \leq b_p - 1$ and $1 \leq c(p,t(p,i-1),t(p,i)) \leq 1 + \log(T+n)$ for all $1 \leq i \leq b_p - 1$. The last interval must be such that $1 \leq c(p,t(p,b_p-1),t(p,b_p)) \leq 2 + \log(T+n)$ and such a collection of points can be formed by a simple greedy strategy. If the last interval produced by this greedy strategy has length smaller than 1, we merge this interval and the second-last interval (this is why we have slack for the size of the last interval).

We call the time intervals $(0 = t(p,0), t(p,1)], (t(p,1), t(p,2)], \ldots, (t(p,b_p-1), t(p,b_p)]$, blocks for page p. Note that there are b_p blocks for page p. We will use $B(p,i)$ to denote the i^{th} block for page p. Let $\mathcal{B}_p$ denote the set of all blocks for page p and let $\mathcal{B} = \cup_p \mathcal{B}_p$ denote the set of all $blocks$. For a block $B(p,i)$, we define its tail to be the time slots t such that $c(p,t,t(p,i)) < 1$. That is, the cumulative amount of page p transmitted after time t until the end of the block in which t lies, is less than 1.

Let us focus on a particular block, say $B(p,i)$. For $\alpha \in (0,1]$, and $l = 0,1,2,\ldots$, and block $B(p,i)$, let $t(p,i,l,\alpha)$ denote the time such that $t(p,i,l,\alpha) \in B(p,i)$ and it is the earliest time when $l+\alpha$ units of page have been broadcast during $(t(p,i), t(p,i,l,\alpha)]$, i.e. since the start of the block $B(p,i)$ until $t(p,i,l,\alpha)$. For a given block $B(p,i)$ and $\alpha \in (0,1]$, let $C(p,i,\alpha)$ denote the set of all time slots $t(p,i,l,\alpha)$ for $l = 0,1,2,\ldots$.

We will only be interested in tentative schedules that are obtained by choosing an offset $\alpha(p,i)$ for each block $B(p,i)$ and transmitting page p at all time slots in $C(p,i,\alpha)$. We use Γ to denote the class of all such possible tentative schedules. Observe that any tentative schedule in Γ satisfies all the requests. This follows as each request for page p that arrives in $B(p,i)$ for $1 \leq i \leq b_p - 1$ (i.e. except for the last block for page p) is served within $B(p,i)$ or in $B(p,i+1)$ by any tentative schedule in Γ. Finally, as there is at least unit of page p broadcast in the LP solution after the arrival of the last request for page p, all requests for page p that arrive during the last block $B(p,b_p)$ are served within $B(p,b_p)$.

The following lemma shows that there is a convex combination of tentative schedules in Γ, such that the total response time is not too high and all the capacity constraints are satisfied at each time step.

Lemma 4.2. *Suppose for each block $B(p,i) \in \mathcal{B}$, we choose the offset $\alpha(p,i)$ uniformly at random in $[0,1]$. Then the tentative schedule satisfies the following properties:*

1. *The expected number of pages transmitted at any time step t is exactly 1.*

2. *For each request, its expected response time is at most 3 times the cost incurred by it in the LP solution of (1)-(6).*

Proof. 1. For any time $t \in B(p,i)$, since we choose $\alpha(p,i)$ uniformly at random in $(0,1]$, then the probability that page p is transmitted at time t is exactly y_{pt}.

For any p, as the blocks $B(p,i)$ partition the entire time interval $(0, T+n]$, for each time t there is exactly one block for page p that contains t. Thus, the probability that page p is transmitted at time t in the tentative schedule is exactly y_{pt}. Summing up over all the pages we have that the expected number of pages transmitted at time t is exactly $\sum_p y_{pt}$ which is exactly 1 by (4).

2. Consider a particular block $B(p,i)$. We say that request ρ is *early* if it does not arrive in the tail of $B(p,i)$, or equivalently that $c(p,t_\rho,t(p,i)) \geq 1$. Note that if a request ρ is early this implies that irrespective of the choice of $\alpha(p,i)$, it will always be served at some time within $B(p,i)$ in the tentative schedule. Since $\alpha(p,i)$ is chosen uniformly at random in $(0,1]$, by Lemma 2.1 the expected response time for early requests is the same as the cost in the LP solution.

Thus we focus on the contribution of requests that are not early. We call such requests late requests. For late requests, it could be the case that they arrive during the block $B(p,i)$ but they are only served during $B(p,i+1)$ in the tentative schedule. Consider a late request ρ for page p that arrives at time $t_\rho \in B(p,i)$ for some i such that $1 \leq i \leq b_p - 1$. Since this request is late, we have that $c(p,t_\rho,t(p,i)) < 1$. With probability $c(p,t_\rho,t(p,i)))$ this request is served in $B(p,i)$ and with probability $1 - c(p,t_\rho,t(p,i))$ it is served in $B(p,i+1)$. Conditioned on the event that this request is served in $B(p,i+1)$, as $\alpha(p,i+1)$ is chosen uniformly at random in $(0,1]$, by Lemma 2.1, its expected response time is $(t(p,i)-t_\rho) + r(p,t(p,i))$. Thus the overall expected response time of ρ is

$$\begin{aligned}
& \sum_{t''=t_\rho+1}^{t(p,i)} (t''-t_\rho)\cdot x_{pt_\rho t''} + (1-c(p,t_\rho,t(p,i)))\cdot(t(p,i)-t_\rho+r(p,t(p,i))) \\
= & \sum_{t''=t_\rho+1}^{t(p,i)} (t''-t_\rho)\cdot x_{pt_\rho t''} + \sum_{t''=t(p,i)+1}^{\infty} (t(p,i)-t_\rho)\cdot x_{pt_\rho t''} + (1-c(p,t_\rho,t(p,i)))\cdot r(p,t(p,i)) \\
\leq & \sum_{t''=t_\rho+1}^{\infty} (t''-t_\rho)\cdot x_{pt_\rho t''} + r(p,t(p,i)) \\
\leq & \; r(p,t_\rho) + 2r(p,t_\rho) = 3r(p,t_\rho)
\end{aligned}$$

The first step follows as $\sum_{t''=t(p,i)+1}^{\infty} x_{pt_\rho t''} = 1 - c(p,t_\rho,t(p,i))$; the second step follows by upper bounding $\sum_{t''=t_\rho+1}^{t(p,i)}(t''-t_\rho)\cdot x_{pt_\rho t''} + (t(p,i)-t_\rho)\sum_{t''=t(p,i)+1}^{\infty} x_{pt_\rho t''}$ by $\sum_{t''=t_\rho+1}^{\infty}(t''-t_\rho)\cdot x_{pt_\rho t''}$; and the last step follows as $t(p,i)$ is a p-good point and hence by definition $r(p,t(p,i)) \leq 2r(p,t_\rho)$. □

Finally, by Lemma 2.2 we can assume that all the offsets α are integral multiples of δ. For ease of notation, we will use $B(p, i, j)$ to denote the time slots in $C(p, i, \alpha = \delta j)$. We will call $B(p, i, j)$ a *block-offset*.

4.2 Auxiliary LP

Recall that our goal is to choose the offsets for each block in such a way that the total response time of the tentative schedule is not too high, and secondly the backlog is small at all times. For this purpose we consider the linear program defined by (7)-(10) below.

We have variables z_{pij}. These correspond to choosing $\alpha(p, i) = j/(T + n)^2$ for block $B(p, i)$. We will have a set of constraints that the total amount of offsets chosen for each block is exactly 1. We will also have the capacity constraints at for each time step, that is the total amount of page transmitted at any time is at most 1. Finally, express the objective function of minimizing the total response time in terms of the variables z_{pij} as follows: For each block $B(p, i)$ we associate a *block-offset response time* $R(B(p, i, j))$ which essentially accounts for the contribution of the block-offset $B(p, i, j)$ to the total response time. Observe that choosing an offset for block $B(p, i)$ can affect the response time of requests for page p that arrive in $B(p, i)$ and possibly the late requests in $B(p, i - 1)$. The block-offset response time $R(B(p, i, j))$ is computed as follows:

1. Let t' denote the earliest time in $B(p, i, j)$. Each request for page p in the tail of the previous block $B(p, i - 1)$ contributes $t' - t(p, i - 1)$ to $R(B(p, i, j))$. Note that this is time (restricted to time units in $B(p, i)$), that any late request in $B(p, i - 1)$ might possibly have to wait.
2. For a request r_p^t for page p that arrive at time t, where $t \in B(p, i)$, we do the following. Let t' denote the earliest time such that $t' > t$ and $t' \in B(p, i, j)$, if such a t' does not exist, then we set $t' = t(p, i)$. Then, the request r_p^t contributes $t' - t$ to $R(B(p, i, j))$. Note that this quantity is the contribution to the response time of r_p^t restricted to the time units in $B(p, i)$.

It is easy to verify by the definition of $R(B(p, i, j))$ that they have the following property.

Lemma 4.3. *If we construct a tentative schedule where for each block $B(p, i)$, we choose a block-offset $B(p, i, j(p, i))$, then the total response time for this tentative schedule is no more than $\sum_{p,i} R(B(p, i, j(p, i)))$.*

We define the following auxiliary linear program:

$$\min \sum_p \sum_i \sum_j R(B(p, i, j)) \cdot z_{pij} \tag{7}$$

$$\text{subject to } \sum_j z_{pij} = 1, \qquad \forall p, i, \tag{8}$$

$$\sum_p \sum_i \sum_j w(B(p, i, j), t) \cdot z_{pij} \leq 1, \qquad \forall t, \tag{9}$$

$$z_{pij} \geq 0, \qquad \forall p, i, j. \tag{10}$$

Here $w(B(p, i, j), t)$ is an indicator function: $w(B(p, i, j), t)$ is 1 if $t \in B(p, i, j)$ and 0 otherwise.

The following lemma essentially follows from Lemma 4.2.

Lemma 4.4. *There is a feasible solution to the LP (7)-(10) with cost no more than 3 times the cost of the LP (1)-(6).*

Proof. Consider the solution where $z_{pij} = \delta$, for all p, i and $0 \leq j \leq 1/\delta - 1$. This corresponds to choosing an offset uniformly at random for each block $B(p, i)$. By Lemma 4.2 (part 1) implies that constraints (10) are satisfied.

We now show that the cost of this solution is no more than 3 times the cost of 1-6. Consider an early request in $B(p, i)$. Since this request is completely served within $B(p, i)$ by Lemma 2.1, the contribution of this request to the objective function is exactly its response time. For a late request for page p that arrives at time t in $B(p, i-1)$, its contribution to the objective function corresponding to $\sum_j z_{p,i-1,j} \cdot R(B(p, i-1, j))$ is exactly $\sum_{t''=t+1}^{t(p,i)} (t'' - t)x_{ptt''} + (1 - c(p, t, t(p, i))) \cdot (t(p, i) - t)$ which is at most $r(p, t)$. Similarly, the contribution to $\sum_j z_{p,i,j} \cdot R(B(p, i - 1, j))$ is exactly $r(p, t(p, i - 1))$ which is at most $2r(p, t)$. $\square$

Observe that the LP above in weak in that z_{pij} could have fractional values, however on the other hand it is tight in the sense that the capacity constraints (9) are satisfied exactly. The rest of algorithm for will deal with obtaining an integral solution to the above LP by relaxing the constraints (9), but still ensuring that the solution thus obtained is useful enough to imply our desired result.

4.3 The algorithm

The idea for our algorithm is the following: We relax the constraints (9) in the auxiliary LP such that we only require these to hold for certain time intervals of rather than for each time unit. We will show that this gives a basic solution where a constant fraction of the variables z_{pij} are assigned integrally. We then redefine the intervals and solve another LP. We repeat this for $O(\log(T + n))$ steps, until all but $O(1)$ variables z_{pij} are non-zero. These are simply set to either 0 or 1 to obtain a valid integral solution, which is our tentative schedule.

To complete the proof we show that the total response of the tentative schedule obtained above is not too high, and that the backlog of this tentative schedule is always $O(\log^2(T + n))$.

Before we can describe the algorithm to compute the tentative schedule formally, we need some notation. Let $I = (t_1, t_2]$ be an interval of time. The size of I denoted by $Size(I)$ is defined as $t_2 - t_1$. The weight of an interval with respect $B(p, i, j)$, which we denote by $w(B(p, i, j), I)$, is the cardinality of the set $B(p, i, j) \cap I$. That is, $w(B(p, i, j), I)$ is the number of time slots in I that belong to $B(p, i, j)$.

Our algorithm will solve a sequence of LP's. At each step k, some variables $x(p, i, j)$ that were fractional at the end of step $k - 1$ get assigned to 1. A partial solution is an assignment where some $x(p, i, j)$ are set to 1. For a partial solution obtained at the end of step k, and an interval I, let $Used(I, k)$ denote the number of time slots in this interval used up by $x(p, i, j)$ that are assigned integrally to 1 by the end of step k, i.e. $Used(I, k) = \sum_{p,i,j} w(B(p, i, j), I)$ such that $x(p, i, j) = 1$. We will use $Free(I, k)$ to denote $Size(I) - Used(I, k)$.

We now describe the algorithm to compute the tentative schedule.

1. Initialize: We divide the time horizon from $1, \ldots, T+n$ into consecutive intervals of size $5\log(T + n)$. We call this collection of intervals $\mathcal{I}_0$. For all $I \in \mathcal{I}_0$, we define $Used(I, 0) = 0$ and $Free(I, 0) = Size(I) - Used(I, 0) = Size(I)$. Let $\mathcal{B}_0$ be the set of all blocks $B(p, i)$ and let $\mathcal{S}_0 = \emptyset$.

2. Repeat the following for $k = 1, \ldots,$.

   - Consider the following linear program defined iteratively based on $\mathcal{B}_{k-1}$, $\mathcal{I}_{k-1}$ and $Free(I, k - 1)$. (This can viewed as a relaxation of the auxiliary linear program (7)-(10) restricted to particular variables).

$$\min \sum_{B(p,i)\in\mathcal{B}_{k-1}} \sum_j R(B(p,i,j)) \cdot z_{pij} \tag{11}$$

$$\text{subject to} \sum_j z_{pij} = 1, \qquad \forall p, i \text{ such that } B(p,i) \in \mathcal{B}_{k-1}, \tag{12}$$

$$\sum_p \sum_i \sum_j w(B(p,i,j), I) \cdot z_{pij} = \mathit{Free}(I, k-1), \qquad \forall I \in \mathcal{I}_{k-1}, \tag{13}$$

$$z_{pij} \geq 0, \qquad \forall p, i, j. \tag{14}$$

- Solve this LP. Let $\mathcal{P}$ denote the set of blocks $B(p,i)$ such that $z_{pij} = 1$ for some j. Let $\mathcal{S}$ denote the set of block-offset pairs $B(p,i,j)$ such that $z_{pij} = 1$.
- Set $\mathcal{B}_k = \mathcal{B}_{k-1} \setminus \mathcal{P}$. These are precisely the blocks $B(p,i)$ for which z_{pij} is not equal to 1 for any j at the end of step k. Set $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \mathcal{S}$. These are precisely the variables z_{pij} that are integrally set to 1 thus far by the end of step k. For each interval $I \in \mathcal{I}_{k-1}$, recompute

  $$\mathit{Used}(I,k) = \mathit{Used}(I, k-1) + \sum_{p,i,j:B(p,i,j)\in\mathcal{S}} w(B(p,i,j), I)$$

  Note that $\sum_{p,i,j:B(p,i,j)\in\mathcal{S}} w(B(p,i,j), I)$ is exactly the number of pages that are assigned to be transmitted during interval I in step k. Set $\mathit{Free}(I,k) = \mathit{Size}(I) - \mathit{Used}(I,k)$. Essentially, $\mathit{Free}(I,k)$ denotes the number of free time slots in interval I, at the end of step k.
- Finally, we compute the set of intervals $\mathcal{I}_k$ by merging the intervals in $\mathcal{I}_{k-1}$ as follows: Initially $\mathcal{I}_k = \emptyset$. Starting from the leftmost interval in $\mathcal{I}_{k-1}$, merge intervals $I_1, I_2 ... I_l \in \mathcal{I}_{k-1}$ greedily to form I until $\mathit{Free}(I_1,k) + \mathit{Free}(I_2,k) + \ldots, \mathit{Free}(I_l,k)$ first exceeds $5\log(T+n)$. We set $\mathit{Free}(I,k) = \mathit{Free}(I_1,k) + \mathit{Free}(I_2,k) + \ldots, \mathit{Free}(I_l,k)$ and $\mathit{Used}(I,k) = \mathit{Used}(I_1,k) + \mathit{Used}(I_2,k) + \ldots, \mathit{Used}(I_l,k)$. By construction, we have that $5\log(T+n) \leq \mathit{Free}(I,k) \leq 10\log(T+n)$. Add I to $\mathcal{I}_k$ and remove $I_1, \ldots, I_l$ from $\mathcal{I}_{k-1}$ and repeat the process until the total free space in the intervals in $\mathcal{I}_{k-1}$ is less than $5\log(T+n)$ and hence we cannot form new intervals. In this case we just merge all the remaining intervals in $\mathcal{I}_{k-1}$ into one interval and add this final interval to $\mathcal{I}_k$.
- If $|\mathcal{I}_k| = 1$ then the algorithm makes one more iteration and then stops. On this last iteration there is just one constraint of type (13) in the relaxed auxiliary LP. The optimal solution is integral and very easy to define. We choose the best offset for every remaining block, i.e. we define $z_{pij} = 1$ if $R(B(p,i,j)) = \min_s R(B(p,i,s))$ for block $B(p,i)$.

4.4 Analysis

Lemma 4.5. *At each iteration of step 2 in the above algorithm, the number of blocks $B(p,i)$ that do not have any z_{pij} set to 1 decreases (almost) by a constant factor. In particular*

$$|\mathcal{B}_k| \leq 0.2 \cdot |\mathcal{B}_{k-1}| + 2$$

Proof. The total number of non-trivial constraints (of type (12) and (13)) in the LP at step k is $|\mathcal{I}_{k-1}| + |\mathcal{B}_{k-1}|$. Consider a basic optimal solution of the LP at stage k. Let f_k be the number of non-zero variables that are set fractionally (strictly between 0 and 1) and let g_k denote the number of variables set to 1. Then, since we have a basic solution, we have that $f_k + g_k \leq |\mathcal{I}_{k-1}| + |\mathcal{B}_{k-1}|$. Now, consider the constraints of type 12, if in some block $B(p,i)$ there is no z_{pij} that is set to 1, then there must be at least 2 variables z_{pij} set fractionally, which implies that $f_k/2 + g_k \geq |\mathcal{B}_{k-1}|$. Combining these two facts implies that $g_k \geq |\mathcal{B}_{k-1}| - |\mathcal{I}_{k-1}|$. By definition, as $|\mathcal{B}_k| = |\mathcal{B}_{k-1}| - g_k$, this implies that $|\mathcal{B}_k| \leq |\mathcal{I}_{k-1}|$.

We now upper bound $|\mathcal{I}_{k-1}|$. Let Free_{k-1} denote the total free space at the end of iteration $k-1$, that is, $\sum_{I \in \mathcal{I}_{k-1}} \mathit{Free}(I, k-1)$. Since each interval except probably the last has at least $5\log(T+n)$ free spaces, we have that $|\mathcal{I}_{k-1}| \leq \lceil \mathit{Free}_{k-1}/(5\log(T+n)) \rceil$. As for any block-offset $B(p,i,j)$ and interval I, the number of time slots $w(B(p,i,j), I)$ is most $\log(T+n)+2$ time slots, it follows from constraints (12) and (13) that Free_{k-1} is at most $(\log(T+n)+2)|\mathcal{B}_{k-1}|$. This implies that

$$|\mathcal{I}_{k-1}| \leq \lceil 0.2 \cdot |\mathcal{B}_{k-1}| + \frac{0.4}{\log(T+n)} \rceil \leq 0.2 \cdot |\mathcal{B}_{k-1}| + 2$$

□

Combining Lemma 4.5 with the fact that $|\mathcal{B}_0| \leq T+n$ we obtain that the algorithm stops after $\log(T+n) + \Theta(1)$ iterations.

In the end of our algorithm we obtain an assignment of zero-one values to variables z_{pij}. Since on every step of our algorithm we relaxed the LP from the previous step, the cost of this final integral solution is upper bounded by the optimal value of (7)-(10), which is at most 3 times the optimal value of (1)-(6). This solution also provides us with integral tentative schedule since it gives us an assignment of pages to the time slots.

To actually obtain a proper schedule from this tentative schedule, we look at the pages transmitted in the tentative schedule at time 0 and greedily assign it to the next free slot after time t. Formally, we can view the process of constructing the feasible schedule from the tentative schedule as follows: There is a queue Q, whenever a page p is tentatively scheduled at time t, we add p to the tail of Q at time t. At every time step, if Q is non-empty, we broadcast the page at the head of Q.

To complete the proof, we show that no page is delayed more than $O(\log^2(T+n))$ than its position in the tentative schedule. Thus it suffices to show that the queue length $Q(t)$ at time t, in the above description is always bounded by $O(\log^2(T+n))$ at all times t.

Lemma 4.6. *Let $\mathit{Used}(t_1,t_2)$ denote the pages transmitted during $(t_1,t_2]$ in the tentative schedule. The maximum queue length at any time is bounded by $\max_{t_1<t_2}(\mathit{Used}(t_1,t_2) - (t_2 - t_1))$.*

Proof. Let t_2 be the time when the backlog in the queue is maximum, and let b denote this backlog. Consider the last time t_1 before t_2 the queue was empty. Since t_1 was the last time when the queue was empty, it must be the case that exactly $t_2 - t_1$ pages were transmitted during the interval $(t_1,t_2]$, and hence b is exactly $\mathit{Used}(t_1,t_2) + t_2 - t_1$. This implies the desired result. □

Lemma 4.7. *For every t_1, t_2, $\mathit{Used}(t_1,t_2) - (t_2 - t_1) \leq 20\log^2(T+n) + O(\log(T+n))$*

Proof. Consider time interval $(t_1,t_2]$. If $\Lambda \leq \log(T+n) + \Theta(1)$ is the number of iterations of our algorithm then there are at most 2Λ intervals $[a,b]$ generated by the algorithm which strictly overlap with t_1, i.e. $a < t_1 < b$ or strictly overlap t_2, i.e. $a < t_2 < b$. The total number of pages assigned to this intervals by the tentative schedule is at most $2\Lambda \cdot 10\log n \leq 20\log^2(T+n) + O(\log(T+n))$.

All other intervals generated by our algorithm do not strictly overlap with t_1 and t_2. They are either completely inside or completely outside $(t_1,t_2]$. We claim that by constraints (13) the total number of pages assigned to the remained intervals overlapping with $(t_1,t_2]$ is upper bounded by $t_2 - t_1$. It follows from the facts that on each iteration we are allowed to use only time slots which were not occupied by the integral assignments from previous iterations and the total number of pages transmitted in every interval on each iteration is exactly length of this interval minus the amount of free space which could be used on the next iteration. The lemma follows. □

Thus we have that

Theorem 1. *The above algorithm produces a broadcast schedule with average response time at most* $3 \cdot OPT + O(\log^2(T+n))$*, where OPT denote the average response time of the optimum schedule.*

4.5 Improving the approximation ratio to $\log^2(T+n)/\log\log(T+n)$

The main idea is to trade-off the constant multiplier term with the additive $O(\log^2(T+n))$ term in the approximation ratio in Theorem 1. We modify the definition (see Definition 1) of a p-good such that, we call a time t to be p-good if $r(p,t) \le \log(T+n) r(p,\tau)$ for all τ, t such that $c(p,\tau,t) < 1$. With this modification, imitating Lemma 4.1, we can form blocks $B(p,i)$ where the amount of page p transmitted in a block is at most $(\log(T+n)/\log\log(T+n)) + 2$. Also, Lemma 4.2 now gives us that the expect response time of the tentative schedule obtained is $O(\log(T+n))$ times that of the optimum cost. Now repeat the algorithm in Section 4.3 with intervals of size $5\log(T+n)/(\log\log(T+n))$ (instead of intervals of size $5\log(T+n)$). As there are $O(\log(T+n))$ iterations of step 2 of the algorithm, it follows directly that the backlog at any time step is at most $O(\log^2(T+n)/\log\log(T+n)$.

Thus we have that

Theorem 2. *The above algorithm is an* $O(log^2(T+n)/\log\log(T+n))$ *approximation algorithm for minimizing the average response time.*

5 Throughput maximization version

LP relaxation. We use the same LP formulation as in the paper [10]. The boolean variable $x_{pt} = 1$ if the page p is transmitted at time t. The boolean variable $y_\rho = 1$ if request $\rho \in \Psi$ is satisfied.

$$\max \sum_{\rho \in \Psi} y_\rho \tag{15}$$

$$\sum_{t=t_\rho+1}^{d_\rho} x_{p_\rho t} \ge y_\rho, \qquad \rho \in \Psi, \tag{16}$$

$$\sum_{p \in P} x_{pt} \le 1, \qquad 1 \le t \le T, \tag{17}$$

$$0 \le x_{pt} \le 1, \qquad 1 \le t \le T, p \in P, \tag{18}$$

$$0 \le y_\rho \le 1, \qquad \rho \in \Psi. \tag{19}$$

As usual on the first step we solve LP (15)-(19) optimally. Let (y^*, x^*) be an optimal solution of that LP.

Defining bipartite graph. We construct the edge weighted bipartite graph $G = (U, V, E, \lambda)$ exactly as in [10]. The vertices in $U = \{1, \ldots, T\}$ represent time slots.

To construct vertices in V and edges E we first choose $z \in [0,1]$ uniformly at random. The vertices in V consist of groups corresponding to each page. For page p we define time intervals $I_{1p}, \ldots, I_{m_p(z)p}$ where $m_p(z)$ is either $\lceil \sum_{t=1}^T x^*_{pt} \rceil$ or $\lceil \sum_{t=1}^T x^*_{pt} \rceil + 1$ depending on random variable z and solution (y^*, x^*). All intervals except first and last are defined that exactly one unit of page p is broadcast during its duration. The intervals are defined iteratively, $I_{1p} = (0, \tau_{1p}]$ where τ_{1p} is the earliest time slot such that $\sum_{t=1}^{\tau_{1p}} x^*_{pt} \ge z$. We define vertex $v_{1p} \in V$ corresponding to this interval and connect it to all vertices in U corresponding to time slots $t \in I_{1p}$ with $x^*_{pt} > 0$. The weight $\lambda(v_{1p}, t)$ of each edge (v_{1p}, t) is exactly x^*_{pt} except the last one which is $z - \sum_{t=1}^{\tau_{1p}-1} x^*_{pt}$.

The interval I_{jp} for $j \geq 2$ is defined in a similar way, $I_{jp} = (\tau_{j-1,p} - 1, \tau_{j,p}]$ where τ_{jp} is the first time slot such that $\sum_{t=1}^{\tau_{jp}} x^*_{pt} \geq j - 1 + z$. We define vertex $v_{jp} \in V$ and connect it by edges to vertices in U corresponding to time slots in I_{jp} with nonzero variables x^*_{pt}. The weights on edges are again equal to x^*_{pt} for all edges (v_{jp}, t) except two edges $(v_{jp}, \tau_{j-1,p})$ and (v_{jp}, τ_{jp}). The weight of the edge $(v_{jp}, \tau_{j-1,p})$ is $x^*_{p\tau_{j-1,p}}$ minus weight of the edge $(v_{j-1,p}, \tau_{j-1,p})$, i.e. leftover of the fractional value of page p located at that slot. The weight of the edge (v_{jp}, τ_{jp}) is $j - 1 + z - \sum_{t=1}^{\tau_{jp}-1} x^*_{pt}$.

The last interval is defined analogously. The only difference is that the total fractional amount of page located in this interval can be any number in the interval $(0, 1]$ depending on solution and random variable z.

Pipage rounding. For every request $\rho \in \Psi$ we define the following nonlinear function on edge weights of graph G. Let $I_{j_\rho p_\rho}$ be the first interval where the request ρ for page p_ρ was released, i.e. $t_\rho + 1 \in I_{j_\rho p_\rho}$ and $t_\rho + 1 \notin I_{j_\rho - 1, p_\rho}$. Then

$$F_\rho(\lambda) = 1 - \left(1 - \sum_{t=t_\rho+1}^{\min\{d_\rho, \tau_{j_\rho p_\rho}\}} \lambda(v_{j_\rho p_\rho}, t)\right)\left(1 - \sum_{t=\tau_{j_\rho p_\rho}}^{\min\{d_\rho, \tau_{j_\rho+1, p_\rho}\}} \lambda(v_{j_\rho+1 p_\rho}, t)\right).$$

The following two lemmata on properties of the function $\sum_{\rho \in \Psi} F_\rho(\lambda)$ combined with pipage rounding yield the desired result.

Lemma 5.1. *Let LP^* be the value of the optimal solution (y^*, x^*). Then $E(\sum_{\rho \in \Psi} F_\rho(\lambda)) \geq \frac{5}{6} LP^*$.*

Proof. If $d_\rho < \tau_{j_\rho p_\rho}$ then $F_\rho(\lambda) = \sum_{t=t_\rho+1}^{d_\rho} \lambda(v_{j_\rho p_\rho}, t) = y^*_\rho$ and this event occurs with probability $1 - y^*_\rho$. Otherwise, the value of $\sum_{t=t_\rho+1}^{\tau_{j_\rho p_\rho}} \lambda(v_{j_\rho p_\rho}, t)$ is distributed uniformly at random in $[0, y^*_\rho]$ by the uniformity of z and the way we defined variables λ. In this case we also can estimate the value of the second sum $\sum_{t=\tau_{j_\rho p_\rho}}^{\min\{d_\rho, \tau_{j_\rho+1,p_\rho}\}} \lambda(v_{j_\rho+1 p_\rho}, t) \geq y^*_\rho - \sum_{t=t_\rho+1}^{\tau_{j_\rho p_\rho}} \lambda(v_{j_\rho p_\rho}, t)$ since y^*_ρ is the lower bound on amount of page p transmitted in the interval $(t_\rho, \min\{d_\rho, \tau_{j_\rho+1,p_\rho}\}]$. Therefore,

$$E(F_\rho(\lambda)) \geq (1 - y^*_\rho) y^*_\rho + \int_0^{y^*_\rho} (1 - (1 - \xi)(1 - (y^*_\rho - \xi))) d\xi =$$
$$\left(1 - \frac{(y^*_\rho)^2}{6}\right) y^*_\rho \geq \frac{5}{6} y^*_\rho.$$

Using the linearity of expectation we get the claim of the lemma. □

Lemma 5.2. *Consider an arbitrary integral feasible solution $(\tilde{x}, \tilde{y})$ of the linear program (15)-(19) with the additional property that at most one page p is transmitted during any time interval $I_{jp}, j = 1, \ldots, m_p(z)$ and $IP(\tilde{x}, \tilde{y})$ is the value of $(\tilde{x}, \tilde{y})$. Let $\tilde{\lambda}$ be the weight function defined on graph G such that $\lambda(v_{jp}, t) = 1$ iff page p is transmitted during time slot $t \in I_{jp}$ in the integral solution $(\tilde{x}, \tilde{y})$. Then $IP(\tilde{x}, \tilde{y}) \geq \sum_{\rho \in \Psi} F_\rho(\tilde{\lambda})$.*

Proof. The fact that page p is transmitted only once during any time interval $I_{jp}, j = 1, \ldots, m_p(z)$ guarantees that $F_\rho(\tilde{\lambda}) \in \{0, 1\}$. If $F_\rho(\tilde{\lambda}) = 1$ then either $\sum_{t=t_\rho+1}^{\min\{d_\rho, \tau_{j_\rho p_\rho}\}} \lambda(v_{j_\rho p_\rho}, t) = 1$ or $\sum_{t=\tau_{j_\rho p_\rho}}^{\min\{d_\rho, \tau_{j_\rho+1,p_\rho}\}} \lambda(v_{j_\rho+1 p_\rho}, t) = 1$. Each case guarantees that $\tilde{x}_\rho = 1$ which implies the statement of the lemma. □

Theorem 3. *There exists an integral solution of linear program (15)-(19) of value at least $\frac{5}{6} LP^*$ and such a solution can be found in polynomial time.*

Proof. We first show that there exists an integral feasible solution $(\tilde{x}, \tilde{y})$ of the linear program (15)-(19) with additional property that at most one page p is transmitted during any time interval $I_{jp}, j = 1, \ldots, m_p(z)$ such that $\sum_{\rho \in \Psi} F_\rho(\tilde{\lambda}) \geq \sum_{\rho \in \Psi} F_\rho(\lambda)$ and that such a solution can be found in polynomial time.

We construct integral assignment $(\tilde{\lambda})$ of pages into time slots iteratively starting with fractional assignment λ and graph G. On every step we keep only edges with fractional weights λ. We find a cycle in a current graph G or any maximal path if G is acyclic. This cycle or a path can be represented as a union of two matchings M_1 and M_2 since G is bipartite. We modify edge weights by adding ϵ to weight of edges in M_1 and subtract ϵ from weight of edges in M_2. Let $\lambda(\epsilon)$ be new weight function. The function $F_\rho(\lambda(\epsilon))$ is a convex function of ϵ since it is a quadratic polynomial with nonnegative main coefficient. Therefore it is maximized on endpoints of the interval from where ϵ is chosen. We chose the largest or smallest ϵ whichever maximizes $F_\rho(\lambda(\epsilon))$ and makes one of the weights in $M_1 \cup M_2$ integral (0 or 1). The detailed description of pipage rounding and its applications is contained in [1].

By repeating this procedure at most $|E|$ times we get an integral feasible assignment $(\tilde{\lambda})$ of pages into intervals such that at most one page p is transmitted during any time interval $I_{jp}, j = 1, \ldots, m_p(z)$. The $(\tilde{x}, \tilde{y})$ is a corresponding solution of LP (15)-(19).

Combining Lemmata 5.1 and 5.2 we obtain that expected value of the solution $(\tilde{x}, \tilde{y})$ is at least $\frac{5}{6} LP^*$. To find such a solution deterministically we can as usual discretize the probability space for random variable z and chose the outcome with best value. Details of this procedure are omitted. $\square$

6 Acknowledgments

We thank Moses Charikar, Sanjeev Khanna, Tomasz Nowicki, Tracy Kimbrel, Seffi Naor, Kirk Pruhs, Baruch Schieber and Grzegorz Swirscsz for helpful discussions.

References

[1] A. Ageev and M. Sviridenko, *Pipage Rounding: a New Method of Constructing Algorithms with Proven Performance Guarantee.* Journal of Combinatorial Optimization, 8, 2004.

[2] A. Bar-noy, S. Guha, Y. Katz, J. Naor, B. Schieber and H. Shachnai, Throughput Maximization of Real-Time Scheduling with Batching, In Proc. of Soda 2002, pp.742-751.

[3] N. Bansal and M. Charikar and S. Khanna and J. Naor. *Approximating the average response time in broadcast scheduling.* Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005.

[4] Y. Bartal and S. Muthukrishnan. *Minimizing maximum response time in scheduling broadcasts.* Proc. of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 558-559, 2000.

[5] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, First Edition, 2000.

[6] J. Edmonds and K. Pruhs. *Broadcast scheduling: when fairness is fine.* Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 421-430, 2002.

[7] J. Edmonds and K. Pruhs. *A maiden analysis of Longest Wait First.* Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 818-827, 2004.

[8] T. Erlebach and A. Hall. *NP-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow.* Proc. 13th ACM-SIAM Symp. on Disc. Algorithms, 2002, pp. 194-202.

[9] R. Gandhi, S. Khuller, Y. Kim and Y-C. Wan. *Approximation algorithms for broadcast scheduling.* Proc. of the 9th Conference on Integer Programming and Combinatorial Optimization (IPCO), 2002.

[10] R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan. *Dependent Rounding in Bipartite Graphs* submitted for publication, preliminary version in Proc. of the Forty-Third IEEE Symposium on Foundations of Computer Science (FOCS'02), pages 323-332, Nov. 2002.

[11] B. Kalyanasundaram, K. Pruhs, and M. Velauthapillai, *Scheduling broadcasts in wireless networks*, Proc. of the 8th Annual European Symposium on Algorithms, 2000, pp. 290-301.

[12] S. Khuller and Y. Kim, *Equivalence of Two Linear Programming Relaxations for Broadcast Scheduling*, Operations Research Letters, 32(5), 473–478 2004.

[13] K. Pruhs, J. Sgall, E. Torng, *Online Scheduling.* Handbook of Scheduling: Algorithms, Models, and Performance Analysis, editor Joseph Y-T. Leung, CRC Press, 2004.

7 Appendix

Proof of Lemma 2.2.

Lemma 7.1. *We can assume that $x_{ptt'}$ and y_{pt} are integral multiples of $\delta = 1/(T+n)^2$. This at most adds 1 to the response time of each request.*

Proof. Given an arbitrary LP solution, we simply round down the values of y_{pt} to the closest multiple of δ and modify $x_{ptt'}$ accordingly. We also transmit $\delta \cdot T \leq 1/n$ units of each page p at time $T+n+1$ to ensure that each request remains completely satisfied.

Observe that each $x_{ptt'}$ is reduced by at most δ. As the response time for a request for page p at time t is $\sum_{t'>t}(t'-t)x(p,t,t')$, the rounding adds at most $T \cdot T \cdot \delta \leq 1$ to the response time of each request. $\square$

Bad example:

We give an example of an LP solution for which every tentative schedule that is local has a backlog of $\Omega(\sqrt{n})$ at some time. This will imply that algorithmic techniques based on local tentative schedules are unlikely to yield an approximation guarantee better than $\Theta(\sqrt{n})$.

We construct a half integral solution to the LP as follows: Let H be the Hadamard matrix of order n, and J be the matrix of order n with all entries equal to 1. Consider the matrix $A = \frac{1}{2}(H+J)$. The matrix A is a $\{0,1\}$ matrix where each row A_i (expect for the one will all 1's contains contains exactly $n/2$ 1's. It is well known that for any vector $x \in \{-1,+1\}^n$, there is row with discrepancy at least $\sqrt{n}/2$ (See for example, Page 17, [5]). That is, for each vector $x \in \{-1,+1\}^n$, $|A_i x| >= \sqrt{n}/2$ for some $1 \leq i \leq n$.

We view these A_i as subsets of $\{1,\ldots,n\}$. We also assume that n is a multiple of 4. Let n_i denote the number of $1's$ in A_i. Let $S_i = n_1 + \ldots + n_i$, and let $S_0 = 0$. The LP schedule is constructed as follows: The schedule transmits 1/2 unit of each page in A_i during the interval $(S_{i-1}, S_{i-1}+n_i/2]$, and again during the interval $(S_{i-1}+n_i/2, S_i]$. For $j \in A_i$, we call its (half unit) transmission time during $(S_{i-1}, S_{i-1}+n_i/2]$ its *odd* slot and its transmission during $(S_{i-1}+n_i/2, S_i]$ its *even* slot.

Note that one consequence of having a local tentative schedule is that if a page is transmitted in its odd slot at some time t, then the next transmission of this page must be no later than next odd slot for this page. We first consider strictly local schedules where each page in transmitted in the tentative schedule only during the odd slots or only during the even slots. This associates a vector $x \in \{-1,+1\}^n$ where is the i^{th} entry is -1 if page i is transmitted during odd slots and is 1 otherwise. As the number of pages transmitted by the tentative schedule during $(S_{i-1}, S_i]$ is exactly n_i, it is easy to see that $|A_i x|$ is exactly equal to the backlog at time $S_{i-1}+n_i/2$ or S_i, which implies the desired claim for strict tentative schedules. If the tentative schedule is not strict, the tentative schedule will end up transmitting a higher cumulative amount of pages than transmitted by the LP. By repeating the above instance say k times, it is easy to see that this will create a backlog of at least k.