

Broadcast Scheduling for Time-Constrained Information Delivery

Majid Raissi-Dehkordi
OPNET Technologies Inc.
Email: mraissidehkordi@opnet.com

John S. Baras
Institute for Systems Research and
Electrical and Computer Engineering Department
University of Maryland
Email: baras@isr.umd.edu

Abstract—In this report, the problem of broadcast scheduling in *Push* broadcast systems is studied. We introduce an optimization approach that leads to well justified policies for *Push* broadcast systems with time constraints. In particular, we apply our results to a *Push* broadcast system with different deadlines associated to the files while allowing the files to have arbitrary demand rates and lengths. We calculate the optimal average cost for our experimental settings and show, through extensive simulation studies, that the results obtained from our scheduling policy are very close to that optimal value for each experiment.

I. INTRODUCTION

The increasing demand for content delivery applications in recent years have resulted into numerous research works on more efficient methods for the delivery of information. In a typical data delivery application, there are a few information sources and a large number of users. News, weather, traffic, music, and stock quotes are examples of the types of information that can be provided by these applications. Although the above services are already implemented over terrestrial links, it is their combination with wireless technologies that can result in very efficient information delivery systems. The inherent broadcast nature of wireless communications (including satellite technology) makes it the ideal media for delivering popular information contents from a single source to multiple users. The two main architectures for broadcast delivery are the one-way (*Push*) and the on-demand (*Pull*) systems. The two systems differ in the lack or presence of a return channel to transfer the instantaneous user requests to the server. In a *Push* system, which is the subject of this paper, the server does not actually receive the requests and schedules its transmissions based on the statistics of the user request pattern and other content-dependent parameters. Obviously, such systems benefit from a high degree of scalability since a single broadcast of an information file will serve all users for that file and the utilized downlink bandwidth is independent of the number of users.

One of the main problems in the design of *Push* broadcast systems is finding the optimum order for broadcasting different information contents over a single channel in order to achieve the "best" performance for a given bandwidth. In this report we find the *optimal* (with respect to the specific cost function defined) scheduling policy and also provide a benchmark for

evaluating other heuristic algorithms. Our main contribution is deriving a solution that allows arbitrary cost functions to be assigned to the information files. Specifically, we address the systems where *deadlines* are assigned to the information files and introduce policies that minimize the average *tardiness* over all users.

The main body of previous research on this subject has been concentrated on policies that minimize the average *waiting time* over all users. However, at least for certain types of information, pure delay can be too simplistic of a measure for cost representation. For example, for the users of the stock information, only a small amount of delay can be acceptable and the information starts to lose its value after certain delay. On the other hand, for the users of weather information, a larger delay is acceptable and the information keeps its value for a longer time. This and other similar facts are the main rationale for this research i.e., to address the scheduling problem in a general setting beyond the average delay criteria. We approach the scheduling problem from an optimization point of view and derive a lower bound on the achievable average cost by relaxing some of the constraints of the problem. We then use the results and the form of the optimal policy for that problem to derive scheduling policies for our original problem and verify the effectiveness of our solution by simulation studies and comparing the results with the lower bound cost.

This paper is organized as follows. In Section II the exact formulation of the problem is presented and the previous works on this subject are reviewed. In Section III our optimization approach to the problem and the proposed scheduling policy are explained. Section IV is dedicated to performance evaluation of our policy.

II. PUSH BROADCAST SCHEDULING, FORMULATION AND PREVIOUS WORK

In a typical *Push* broadcast system N separate information files are stored in the system. The aggregate request arrival process for each file is modeled by a Poisson process and we denote by λ_i the rate of the process for file i ; $i = 1, \dots, N$. We also denote by l_i , $i = 1, \dots, N$ the transmission time of file i over a unit bandwidth channel (*length*). In a *Push* system, the only information available for the scheduler about the requests is their arrival rates λ_i ; $i = 1, \dots, N$. We define a cost function $C_i(t)$; $i = 1, \dots, N$ which represents the cost incurred by the system for a user of file i when t seconds

This Research was partially supported by NASA cooperative agreement NCC3-528 when the first author was a research associate at the Institute for Systems Research, University of Maryland.

have been passed since the user needed to receive the file. The system is non-preemptive, meaning that ongoing transmission can't be interrupted by the system. Therefore, the broadcast of file i will take exactly l_i seconds.

The scheduling problem is defined as finding an infinite schedule for broadcasting the N files such that the total average cost incurred by the system is minimized. The waiting times are defined as the time between the user request time and the *start* of the broadcast of the file.

If we denote by \bar{C}_i the long-term average cost for file i , the overall average cost can be written as

$$C = \frac{1}{\lambda} \sum_{i=1}^N \lambda_i \bar{C}_i \quad (1)$$

where $\lambda = \sum_{i=1}^N \lambda_i$. Although the problem of broadcast scheduling has not been considered in the above general form before, there has been a number of rather interesting works on both the Push and Pull systems with average waiting time objective function ($C_i(t) = t$, $i = 1, \dots, N$). One of the earliest works on the Push broadcast scheduling is the work by Ammar and Wong [1], [2] where they studied a system with all files having a unit length and showed that the optimal scheduling policy has a periodic form. They showed that the optimal inter-broadcast periods of any two files i and j on the broadcast channel are related by $\frac{\tau_i}{\tau_j} = \sqrt{\frac{\lambda_j}{\lambda_i}}$ i.e., the files with lower request arrival rates are broadcasted less often (larger periods). They also introduced a heuristic method for designing one period of the broadcast cycle with a given cycle length L to satisfy the optimality equation as closely as possible. The final broadcast schedule is then achieved by repeating that cycle over time. In a later work [3], it was shown via optimization arguments that if the files have different sizes l_i ; $i = 1, \dots, N$, the previous relation is extended as $\frac{\tau_i}{\tau_j} = \sqrt{\frac{\lambda_j}{\lambda_i}} \sqrt{\frac{l_i}{l_j}}$. They introduced a real-time heuristic policy for achieving near-optimal results was introduced that determines the next file to broadcast based on the information available at the end of each broadcast. In another work, Su and Tassiulass [4], [5] proposed a parametric real-time policy and optimized the value of the parameter through a number of simulation experiments. The resulting policy through their approach turns out to be the same as the policy in [3]. The interested reader is also referred to other publications on this subject such as [6]–[8] and references therein for a more diverse review of the problem and its alternative settings. The common property between all of the above results is that the cost function is always of the $C_i(t) = c_i t$ form.

III. OUR APPROACH

In this section, we address the problem of broadcast scheduling in Push systems in its general form when the cost function is any monotonic non-decreasing function of time and present an optimization approach to find near-optimal scheduling policies. We will then apply our method specifically to a system where the files have different lengths and the cost function is the well-known *Tardiness* criteria used frequently in the Operations Research field. This criteria also comes up in a

slightly different broadcast system where the user's device constantly receives and stores the files and the user always accesses the most recent version stored in the device. It is not difficult to show that the scheduling problems for those types of systems will reduce to the same problem that we study. In general, to our knowledge, the broadcast scheduling problem in Push systems has not been addressed in its generality and there are no policies that address the problem beyond the average waiting time criteria.

We first consider the scheduling problem in a system similar to our system but with weaker constraints (*relaxed problem*). After finding the optimal solution for the new problem, we use that to come up with a scheduling policy for the original system. Our assumptions and notations are as follow:

- N : Total number of files stored in the system
- The request generation process for each file i is a Poisson process with known rate λ_i ; $i = 1, \dots, N$
- l_i : Length of file i
- $C_i(t)$: Cost function associated file i . For all i , $C_i(t) = 0$ if $t \leq 0$.
- It is assumed without loss of generality that the total channel bandwidth is 1
- Only one file can be in transmission at any given time (Time Division Multiplexing)

In the relaxed problem, we assume that the instantaneous bandwidth is not limited to 1 and only the long-term average of the total used bandwidth should not exceed 1. This assumption is similar to a relaxation made in [9] and [10] in a Dynamic Programming approach to the broadcast scheduling problem in Pull systems (originally introduced in [11] in the general context of Restless Bandit Problems). It is not difficult to show that the optimal policy always fully utilizes the bandwidth and does not leave the channel idle. Since our original problem with the strict constraint on the instantaneous bandwidth is a special case of the relaxed problem, the optimal cost for this new system is obviously a lower bound for the original system. This will later allow us to compare the performance of our policies with this lower bound and find out how well they perform.

Let's denote by r_i the average long-term bandwidth used for broadcast of file i . The only constraint is then to have

$$\sum_{i=1}^N r_i \leq 1. \quad (2)$$

Having r_i values fixed for each file i , $i = 1, \dots, N$, it can be shown that the average cost is minimized with a periodic broadcast schedule for each channel.

Theorem 1: For a single file with length l , average broadcast bandwidth r , and monotonic non-decreasing cost function $C(t)$, the average cost is minimized when the file is broadcasted with a fixed period.

Proof: See [12].

Given the length l_i and allocated bandwidth r_i for a typical file i , the broadcasts happen with a period $\tau_i = l_i/r_i$ (figure 1). For such a periodic schedule, the long-term average cost

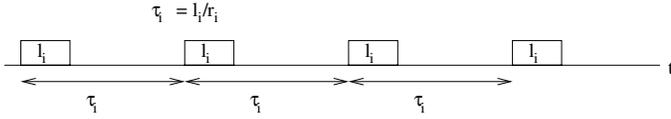


Fig. 1. optimal broadcast schedule on a single channel

for each file is equal to the average cost per period given by

$$\bar{C}_i = \frac{1}{\tau_i} \sum_{n=0}^{\infty} \frac{e^{-\lambda_i \tau_i} (\lambda_i \tau_i)^n}{n!} n c_i = \lambda_i c_i \quad (3)$$

where c_i is the average cost incurred by each user which is

$$c_i(\tau_i) = \frac{1}{\tau_i} \int_0^{\tau_i} C_i(t) dt. \quad (4)$$

With the above definitions, the relaxed scheduling problem can be formulated as a constrained optimization problem as follows

$$\min_{\tau_1, \dots, \tau_N} \frac{1}{\lambda} \sum_{i=1}^N \lambda_i c_i$$

such that $\sum_{i=1}^N \frac{l_i}{\tau_i} \leq 1$ and $\tau_i > 0, \forall i \in \{1, \dots, N\}$

where $\lambda = \sum_{i=1}^N \lambda_i$. From a practical point of view, since we have not assigned any cost for using the channel, it is obvious that the optimal policy would make full use of the channel bandwidth. Equivalently, it means that the optimal solution of the above problem occurs on the border of the constraint space i.e., when $\sum \frac{l_i}{\tau_i} = 1$. The precise statement of this fact is as follows

Theorem 2: If all $C_i(\cdot)$ s are monotonic non-decreasing, then the solution $(\tau_1^*, \dots, \tau_N^*)$ for the above optimization problem occurs on the boundary of the constraint space i.e., we have $\sum \frac{l_i}{\tau_i^*} = 1$.

Proof: See [12].

We can now use the Lagrange method to find the optimal solution for the relaxed problem. Let's introduce the *relative demand* parameters $q_i = \lambda_i / \lambda$ for $i = 1, \dots, N$. we need to find

$$\min_{\tau_1, \dots, \tau_N, \mu} L \quad (5)$$

where

$$L(\tau_1, \dots, \tau_N, \mu) = \sum_{i=1}^N q_i c_i + \mu \left(\sum_{i=1}^N \frac{l_i}{\tau_i} - 1 \right). \quad (6)$$

the optimal solution satisfies

$$q_i \frac{dc_i}{d\tau_i} - \frac{\mu l_i}{\tau_i^2} = 0; \quad i = 1, \dots, N \quad (7)$$

$$\sum_{i=1}^N \frac{l_i}{\tau_i} = 1. \quad (8)$$

The above $N + 1$ equations can be solved to find the optimal values of τ_1 to τ_N resulting in the minimum total average

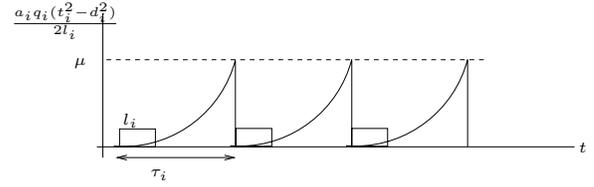


Fig. 2. optimal broadcast schedule on a single channel

cost. These equations place a requirement on the period τ_i that depends on the length and demand rate of file i

$$\frac{q_i \tau_i^2}{l_i} \frac{dc_i}{d\tau_i} = \mu. \quad (9)$$

or

$$\frac{\tau_i^2}{\tau_j^2} = \frac{q_j l_i}{q_i l_j} \frac{dc_j/d\tau_j}{dc_i/d\tau_i}. \quad (10)$$

The above constraints are our guidelines for coming up with a solution for the original system where only one file at any time can be in broadcast over the single available channel. We treat the problem as a dynamic scheduling problem despite the fact that the system is completely deterministic. We use this approach throughout this paper since it removes the problem of designing a periodic schedule at the expense of some decision overhead at the end of each broadcast. Since the decision policy turns out to have low computational complexity, this approach can be easily implemented in real systems.

Let's define t_i as the time since the last broadcast of file i . In the ideal case, file i is broadcast each time t_i hits the τ_i value which satisfies equation (9) as shown in figure (2). However, since for some files, t_i may reach the optimal value while another file is in transmission, several files may have their t_i values passed the optimal value at the end of the current broadcast. We therefore use the $\frac{q_i t_i^2}{l_i} \frac{dc_i}{d\tau_i} - \mu$ value for each file as the *eligibility* of that file (or the *index* function in Dynamic Programming terminology) for broadcast at the current decision time. Since any monotonic increasing function of our *eligibility* measure can also be used as the index function, we can instead use the policy that assigns the index function

$$\nu_i = \frac{q_i t_i^2}{l_i} \frac{dc_i}{d\tau_i} \quad (11)$$

to each file and selects the file with the largest ν_i value for broadcast. The only information required by the policy about the state of the system are the last broadcast time for each of the pages, since the t_i values would simply be the current time minus those values. Since the index functions are calculated independently for each file, This policy has a complexity $O(N)$ which confirms our previous claim about its low computational cost.

Having established a general framework for index policies for the Push broadcast systems, we can now be more specific and consider systems with specific cost functions.

A. Systems with average tardiness criteria

When the timeliness of receiving the information is of importance and each file has its own expiration period, the

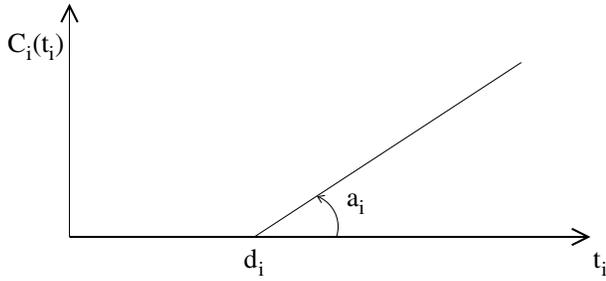


Fig. 3. Cost function for the tardiness criteria

criteria that is often used is the average tardiness criteria. The cost function representing the tardiness is shown in figure 3 where the cost starts to grow with slope a_i only after a deadline associated with the file is passed. With this definition, the average delay criteria is in fact a special case of the average tardiness criteria when all $d_i = 0$. Following (4) For the tardiness cost function, we have

$$c_i = \begin{cases} \frac{1}{2} a_i \frac{(\tau_i - d_i)^2}{\tau_i} & \tau_i \geq d_i \\ 0 & \tau_i < d_i. \end{cases} \quad (12)$$

which implies (assuming $\tau_i > d_i$)

$$\frac{dc_i}{d\tau_i} = \frac{a_i}{2} \frac{\tau_i^2 - d_i^2}{\tau_i^2}. \quad (13)$$

Using equation (9), we find

$$\tau_i = \sqrt{\frac{2l_i\mu}{a_i q_i} + d_i^2} \quad (14)$$

and the index function can be defined as

$$\nu_i = \frac{a_i q_i (t_i^2 - d_i^2)}{l_i}. \quad (15)$$

According to this function, in otherwise similar conditions, a more popular file (larger q_i) is given priority over a lesser popular file. Similarly, a more time-critical file (small d_i) has priority over another file with a larger deadline. The length however, has a negative impact on the priority and a file with a shorter length is chosen over a longer file in similar conditions. Also note that this index policy does not contain τ_i or μ terms and therefore there is no need for explicit calculation of those quantities by the system.

Here we have implicitly assumed that the average total bandwidth is smaller than that needed for periodic broadcast of all files right on their deadline expiration times i.e., some files need to be transmitted after the expiration of their deadlines. Otherwise, the problem would have been trivial. This assumption can be expressed as

$$\sum_{i=1}^N \frac{l_i}{d_i} > 1. \quad (16)$$

Combining equations (8) and (14) results in the following

equation for μ

$$\sum_{i=1}^N \frac{1}{\sqrt{\frac{2\mu}{a_i l_i q_i} + \left(\frac{d_i}{l_i}\right)^2}} = 1. \quad (17)$$

This equation can be solved to find the optimal value of μ . It can be easily shown [12] that the solution for μ is unique and is greater than zero. This guarantees that, based on equation (14), every τ_i is greater than its corresponding d_i and therefore the assumption of c_i taking the first case in equation (12) remains valid. It is also intuitively clear from figure 3, and not difficult to show, that the optimal solution should be sought in the $\tau_i \geq d_i$ region since nothing is gained by going to the $\tau_i < d_i$ region for any i .

The optimal τ_i values can now be computed from μ using equation (14) and the optimal value of the total average tardiness would be

$$C = \frac{1}{2} \sum_{i=1}^N q_i a_i \frac{(\tau_i - d_i)^2}{\tau_i}. \quad (18)$$

Again, this value is a lower bound for the original system where the bandwidth can not exceed 1, even instantaneously.

B. Multi-channel Broadcast systems

Our approach can be also easily applied to systems with more than one broadcast channels. Let's assume that the system has K parallel channels ($K < N$) each with bandwidth 1. We also assume that all users are able to receive a file from any one of the K channels. The scheduling problem for the relaxed system in this case can be written as

$$\min_{\tau_1, \dots, \tau_N} \frac{1}{\lambda} \sum_{i=1}^N \lambda_i c_i \quad \text{such that} \quad \sum_{i=1}^N \frac{l_i}{\tau_i} \leq K$$

with the assumption that $\sum_{i=1}^N \frac{l_i}{d_i} > K$. Following the same approach, the optimality equations for the τ_i values will still be the same as equation (14) but with a different value for μ which is, for the average tardiness criteria, the solution of the following equation

$$\sum_{i=1}^N \frac{1}{\sqrt{\frac{2\mu}{a_i l_i q_i} + \left(\frac{d_i}{l_i}\right)^2}} = K. \quad (19)$$

Using similar discussions about the properties of this equation, it can be shown that the index policy for this case is to calculate the *same* index function for each file as before and broadcast the files with the K largest index values.

Having derived our scheduling policy, we can now discuss its performance through a number of simulation studies.

IV. PERFORMANCE RESULTS

In order to evaluate the performance of our policy, we set up a Push broadcast system with 100 files. We set the total demand rate λ as a variable and pick q_i values according to a Zipf distribution [13] with unit exponent i.e., $\forall i, j : q_i/q_j = j/i$ and $\sum q_i = 1$.

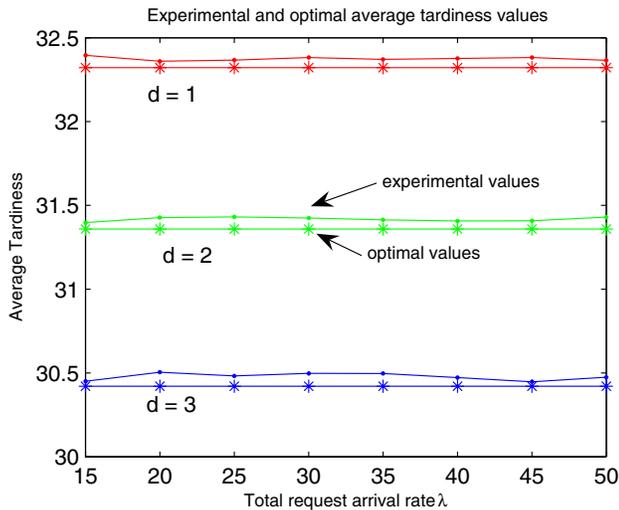


Fig. 4. Optimal and experimental average tardiness for a system with equal deadlines assigned to all files

Since the policy allows the files to have different q_i , d_i , l_i and, a_i values, our experiments are broken into several sets. As an initial study, we eliminate the effect of file lengths and weights by setting all l_i and a_i values to one. We also set all deadlines to a common value d and run the experiments by changing the total request arrival rate λ from 15 to 50 with a step of 5 and setting the common deadline value to $d = 1$, $d = 2$ and $d = 3$. Figure 4 shows the total average tardiness obtained from the experiments along with the lower bound values calculated for each experiment. Our first observation is that the average cost is independent of the total request arrival rate. This is expected since the contribution of the arrival rates λ_i to our policy is only through the normalized q_i values. Based on this observation, the total arrival rate in all of our following experiments is set to a fixed value of 50 and is not changed. The second observation in this experiment is the small difference between the experimental results and the corresponding optimal values. As we expect, for a fixed total bandwidth, shorter deadlines result in larger average costs.

Since the optimal value of the average tardiness for the relaxed problem is a lower bound for our real problem, the *goodness* measure in our experiments is defined as how close we get to that optimal value for each experiment. If we denote by \hat{C} the average tardiness resulting from our heuristic policy and by C the lower bound average tardiness given by equation (18), the *goodness* measure G is defined as

$$G = 100 * \frac{\hat{C} - C}{C}. \quad (20)$$

Calculating the above goodness measure for experiments in Figure 4 showed a maximum of 0.3% difference between the optimal values and the results of our policy. In the second set of experiments, both the deadlines and file lengths were varied and the performance of the policy under different assignment methods for both quantities was evaluated. The total arrival rate was fixed at 50 and all weights were set to one. The deadlines were assigned to the files in three different ways. In the first

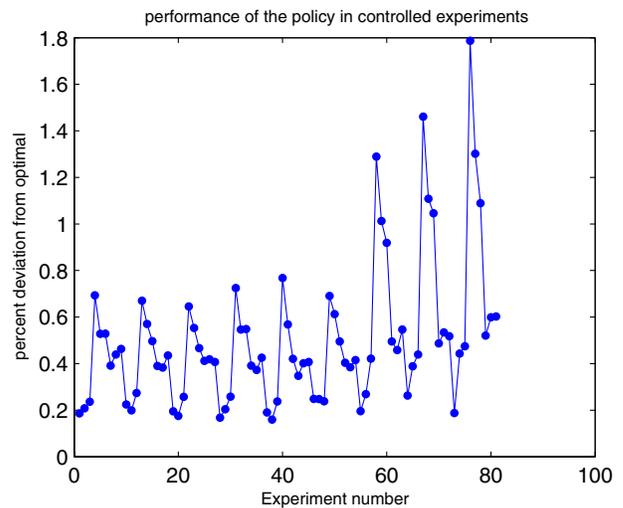


Fig. 5. Relative deviation of the results from the optimal values for controlled assignment of deadlines and lengths

set of experiments, all deadlines were equal to a fixed number d . In the second set, the deadline assignment was a linear function of the file number with a positive slope such ($d_i = (i-1)/10+d$) and in the third set of experiments, the deadlines were assigned in the reverse order ($d_i = (99-i)/10+d$). The same set of options was applied to the file lengths as well and the file lengths took a constant value l , as well as increasing and decreasing values with offset l . The goal was to find out how the policy performs in the above cases and to discover worst case conditions for its performance. For these experiments, both d and l took values from $\{1, 2, 3\}$ resulting in a total number of 81 experiments. Figure 5 shows the G values for those experiments. The worst case scenarios can be easily located as the nine points on the right half of the graph. Those points represent the cases where the most popular file had the longest deadline and shortest length while the least popular file had the shortest deadline and the longest length. The three peak points among these nine experiments are those where the file length offset is minimum ($l = 1$) and the highest peak is when the deadline offset is also at its maximum value of $d = 3$. Overall, even in the worst case, the relative deviation from optimal is reasonably small.

Our final test aimed at evaluating the policy in the presence of unequal deadlines, unequal lengths, and unequal weights being assigned to the files. This set consisted of 100 simulations in which the values of file lengths, deadlines and weights for each file were assigned by separately sampling a *uniform*(1,10) distribution. Due to space constraints the graph is not shown here (See [12]) but the main observation was that the difference between the experimental results and the optimal values was smaller than 1% for all cases.

Although our main goal has been to achieve the minimum average cost in each problem, it is constructive to study the treatment of the individual files by the optimal policy and see how close the heuristic policy approximates the optimal policy for each file. The first question can be answered by looking at

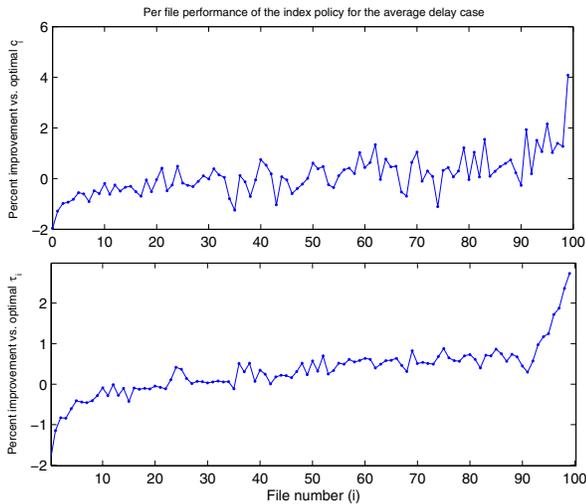


Fig. 6. (top) Relative difference between the experimental and optimal values of the average delay for each file. Positive values indicate smaller (better) than optimal. (bottom) Relative difference between the experimental and optimal values of the average broadcast period for each file. Positive values indicate smaller (better) than optimal.

our previous derivations and plugging in equation (14) in (12) to find all individual average tardiness values as functions of μ . Since the resulting functions are not easy to investigate, we consider the case with all $d_i = 0$. In that case, we have for all i and j ,

$$\frac{c_i}{c_j} = \frac{a_i \tau_i}{a_j \tau_j} = \frac{\sqrt{q_j} \sqrt{l_i} \sqrt{a_i}}{\sqrt{q_i} \sqrt{l_j} \sqrt{a_j}}.$$

Non-zero d_i values introduce skewness to the above relations and should be mainly investigated separately for each specific scenario.

The second question deals directly with how well our proposed policy approximates the optimal policy. Although at this point we are not able to make a general statement about this subject, we focus on a simplified setting and try to answer this question for that case. We consider a system with all l_i and a_i values being set to one and $d_i = 0$ for all i i.e., the *average delay* case. The top graph in Figure 6 shows the relative difference between experimental and optimal values of the average delay for individual files. It can be seen that the index policy favors the files with lower demands by providing them with a smaller-than-optimal average delay. Obviously, this effect results in a larger overall average delay. For the same setting, we can record the individual values of the time differences between all successive broadcasts of each file i and observe how close to τ_i those value are. The bottom graph in the same figure plots the percentage difference between the average inter-broadcast time for each file i and its optimal τ_i value. As we expect, this graph is similar to the previous experiment and the heuristic policy favors the files with smaller demand.

In our policy, only the current value of the index function for each file is considered. However, in reality, after the broadcast of a file starts, the index functions of all files and their

corresponding costs keep increasing. Ideally, one may also think of a look-ahead approach that takes this fact into account and come up with a more complicated policy. Although it is not known if such policies may result in a better performance, our experimental results are reasonably close to the optimal values and we do not see a strong motivation for studying such policies. Overall, our results indicate that the proposed scheduling policy with an $O(N)$ complexity performs very close to optimal at least for the family of cost functions defined by figure 3.

V. CONCLUSION

This paper presents a general formulation of the scheduling policy in Push broadcast systems. Our formulation addresses the systems with generalized cost functions and provides index policies for them. We introduced an auxiliary problem and finding the optimal solution for that system enables us to propose a scheduling policy for the original system. Our results are based on the comparison between the performance of the policy and the theoretical lower bounds for the average tardiness criteria. In all of our experiments, even the worst case result was very close to the optimal result. A similar technique is also applicable to Pull systems and this problem is the subject of ongoing research.

ACKNOWLEDGMENT

The authors would like to thank Dr. Samy Abbes for his helpful suggestions and comments about this research.

REFERENCES

- [1] M. Ammar and J. Wong, "On the optimality of cyclic transmission in teletext systems," *IEEE Trans. Comm.*, pp. Vol. 35, pp68–73, Jan. 1987.
- [2] —, "The desgning of teletext broadcast cycles," *Perf. Eval.*, pp. Vol. 5, No. 4, pp235–242, Nov. 1985.
- [3] N. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," *Wirel. Netw.*, vol. 5, no. 3, pp. 171–182, 1999.
- [4] C. Su and L. Tassiulas, "Broadcast scheduling for information distribution," *Proc. of INFOCOM 97*, 1997.
- [5] C. J. Su and L. Tassiulas, "Broadcast scheduling for the distribution of information items with unequal length," *Proc. of the 31st Conference on Information Science and Systems (CISS'97)*, Mar. 1997, Baltimore, Maryland.
- [6] M. Ammar, "Response time in a teletext system: an individual user's perspective," *IEEE Trans. Comm.*, pp. vol. 35, pp1159–1170, Nov. 1985.
- [7] N. Vaidya and H. Jiang, "Data broadcast in asymmetric wireless environments," *Proc. 1st Int. Wrkshp Sat.-based Inf. Serv.(WOSBIS)*, NY, Nov. 1996.
- [8] J. Xu, D. Lee, Q. Hu, and W. Lee, "Data broadcast," in *Handbook of Wireless Networks and Mobile Computing*, edited by I. Stojmenovic, John Wiley & Sons Inc., 2002.
- [9] M. Raissi-Dehkordi and J. S. Baras, "Broadcast scheduling in information delivery systems," *Proceedings of IEEE GLOBECOM2002*, Nov. 2002, Taipei, Taiwan.
- [10] —, "Near-optimal scheduling policies for broadcast of files with unequal sizes in satellite systems," *Allerton conference on communications, control, and computing*, Oct. 2003, Allerton, Illinois.
- [11] P. Whittle, "Restless bandits: activity allocation in a changing world," *A Celebration of Applied Probability*, ed. J. Gani, *J. Appl. Prob.*, 25A, pp287–298, 1988.
- [12] M. Raissi-Dehkordi and J. S. Baras, "Near-optimal scheduling policies for time-sensitive broadcast systems," *Technical Report, Institute for Systems Research*, University of Maryland at College Park, <http://www.isr.umd.edu>, 2007.
- [13] G. Zipf, *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, Cambridge, Massachusetts, 1949.