Near-optimal policies for broadcasting files with unequal sizes in satellite systems

Majid Raissi-Dehkordi
Institute for Systems Research
University of Maryland at College Park
Email: majid@isr.umd.edu

John S. Baras
Institute for Systems Research
University of Maryland at College Park
Email: baras@isr.umd.edu

Abstract— **Information broadcasting is an effective method to deliver popular information pages to a large number of users in wireless and satellite networks. In a previous work, we used a dynamic optimization approach to address the problem of broadcast scheduling for a *pull* system with equal file sizes. In this paper, we address that problem in a more general setting where the file sizes are not equal and have geometric size distributions with possibly different means. The dynamic optimization approach allows us to find a near-optimal scheduling policy, which we use as a benchmark to evaluate a number of other heuristic policies. Also, we modify the resulting policy and apply it to the case with fixed (unequal) file sizes and compare the results with some other well known, as well as new, heuristic policies. Finally, we introduce a low-complexity heuristic policy to be used for practical implementations. The results show that the performance of the new policy is very close to that of the original policy.**

I. Introduction

The rapid growth in the demand for various information delivery services in recent years has sparked numerous research works for finding more efficient methods for the delivery of information. In many applications the flow of data is not symmetric. In what we call a typical *data delivery* application, there are a few information sources and a large number of users, thus, the volume of data transferred from the sources to the users is much larger than that in the reverse direction. The short information messages available in some cellular phones is an example of this type of applications. The WWW traffic, which constitutes a large portion of the Internet traffic, can be also regarded as a data delivery application. The data transferred through these applications is usually the information packages requested by many users as opposed to applications with one-to-one information content such as *email*. This property of the data delivery applications and the fact that every information package is typically requested by a large number of users at any time, makes the wireless broadcast systems a good candidate as the transport media for those applications. These systems, due to their inherent physical broadcast capability, form a one-hop structure where all the receivers share the single download link and receive the requested information at the same time. For the same reason, these systems are highly scalable and can support additional users without the need for major changes in their infrastructures. Throughout this report, we use the term *broadcast system* to refer to this type of systems with physical broadcast capability.

The two main architectures for broadcast delivery are the one-way(or *Push*) and the two-way(or *Pull*) systems. In a *push* system, the server does not actually receive the requests and schedules its transmissions based on the statistics of the user request pattern(hence the term *push*). Conversely, in a *pull* system the server receives all the requests and can schedule the transmissions based on the number of requests for different data packages. In this paper we investigate the problem of optimal scheduling of the broadcast messages in a *pull* system in order to minimize the average waiting time of the users. This situation is more general than the fixed-length setting[1] and applies to applies to a larger number of practical situations. In our formulation, we assume that the page lengths are Geometric random variables with possibly different mean values. Later in this paper, we use the obtained results to consider the fixed-length case as well.

II. Related work

Compared to the wealth of research works on the scheduling problem in *push* broadcast systems, fewer works have addressed the scheduling problem in the *pull* systems. The papers by Ammar, Dykeman and Wong[2] and [3] are probably the first papers that introduced this problem and presented both heuristic and numerical solutions for special cases. Later, Franklin and Aksoy[4] presented a heuristic policy that approximated the Longest Total Wait First (LTWF) policy and used the number of requests for each page together with the time since the last broadcast of the page to calculate the index associated with that page. The policy would then chose the page with the largest index value to be broadcasted. Su and Tassiulas[5] introduced another index policy ,which they named PIP, that used the number of requests for each page together with the request arrival rate for the page to calculate the index associated with each page. Both of these policies performed almost identical to each other and also to the LTWF performance. In [5], a Markov Decision Process(MDP) formulation of the problem was also presented. However, the complex form of the problem prevented them from going very far with that approach. The work by Raissi and Baras[1] is probably the only one to use a MDP formulation of the problem to find an analytical solution. They investigated the problem in a more general framework where distinct weights are associated with the pages and derived a near-optimal index policy using the MDP formulation. They also used that policy to propose low complexity heuristic policies that extend the PIP policy to this more general setting.

All of the above works on *pull* systems with minimum average delay objective assume equal page sizes. This restriction makes it difficult to apply any of those policies to the systems where the stored pages naturally have different sizes and this differences need to be taken into account in designing the scheduling policy. One of the important instances of this problem is in the satellite Internet delivery systems with cache broadcasting where the cache contains web pages with unequal sizes. To our knowledge, there has not been any previous work, neither heuristic nor analytical, on this more general setting and this work seems to be the first attempt to study it. In this paper we propose a MDP formulation of the problem and use the ideas from the Bandit Problems to propose index-type scheduling policies. We study the problem with random file sizes and later extend the policy to the case with deterministic file sizes as well. In section III, we present our MDP formulation of the problem and the Restless Bandit approach for solving it. After proving the necessary properties of the system, we use the Restless Bandit approach to find an index policy for this problem in section IV. Section VI is dedicated to evaluation of the policy and comparing it with some heuristic policies. We then present a low-complexity policy which performs close to our policy and also extends it to the systems with deterministic page sizes.

III. Problem formulation

We denote by $N(>1)$, the number of information pages stored in the system. We assume that the broadcasts can only start in certain time instants which are equally spaced in time. This periodic setting introduces a time unit that can be set to one without any loss of generality. The page sizes are random variables with Geometric distributions with parameter q_i for type i pages. If we denote by l_i the length of page i, we have

$$P[l_i = n] = q_i(1-q_i)^{n-1}, n \geq 1,\ 0 < q_i \leq 1,\ i = 1, \ldots, N. \tag{1}$$

Here we implicitly assume that the sizes are rounded up to the smallest integer multiple of the above time unit. We also allow preemption in the system, i.e. the broadcast of a page can be interrupted by the system, so that another page is broadcasted, and can be resumed at a later time. However, this can only happen at the beginning of every broadcast cycle. This implies that the users are capable of receiving different segments of a page separately and re-assembling them at the receiver. Therefore, every broadcast initiation time $t = 0, 1, \ldots$ is a decision time (and also a possible preemption time). The waiting time of the requests for a page is defined as the time since the arrival of the request until the end of the transmission of the last segment of that page. The new requests for each page which arrive after the beginning of the transmission of the first segment of the page, need to wait till the beginning of the next transmission of the whole page. We also assume that the

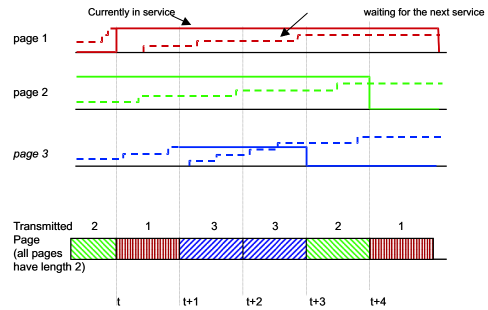

Fig. 1. Sample path of a system with three pages.

system has $K(1 \leq K < N)$ identical broadcast channels. In this *pull* broadcast system, the system has complete knowledge about the number of pending requests for each page and based on this information determines the page to broadcast in the next time unit in order to minimize the average waiting time over all users.

The request arrival process for each page $i;\ i = 1, \ldots, N$ is a discrete-time, stationary, iid process which we show by $A_i(t);\ t = 0, 1, \ldots$. We denote by $p_i(a);\ a \geq 0$ the pmf of the arrivals during every time unit and show its mean value by λ_i. The state of the system at any time instance t is $\mathbf{X}(t) = (X_1(t), Y_1(t), X_2(t), Y_2(t), \ldots, X_N(t), Y_N(t))$ where $X_i(t)$ is the number of requests for page i at time t that have received at least one segment of the requested page and $Y_i(t)$ is the number of requests for the same page which arrived after the broadcast of the first segment of the page and therefore need to wait till the next full broadcast of that page. Each $(X_i(t), Y_i(t));\ i = 1, \ldots, N$ process is a Markov process with transition probability

$$(X_i(t+1), Y_i(t+1)) = \begin{cases} (0, Y_i(t) + A_i(t)) & w.p.\ q_i & if\ i \in d(t) \\ (X_i(t), Y_i(t) + A_i(t)) & w.p.\ (1-q_i) & if\ i \in d(t) \\ (X_i(t), Y_i(t) + A_i(t)) & & if\ i \notin d(t) \end{cases} \tag{2}$$

if $X_i(t) > 0$ and

$$(X_i(t+1), Y_i(t+1)) = \begin{cases} (0, A_i(t)) & w.p.\ q_i & if\ i \in d(t) \\ (Y_i(t), A_i(t)) & w.p.\ (1-q_i) & if\ i \in d(t) \\ (0, Y_i(t) + A_i(t)) & & if\ i \notin d(t) \end{cases} \tag{3}$$

if $X_i(t) = 0$. Here $d(t) \subset \{1, \ldots, N\}$ is the set containing the indices of the K pages broadcast at time t. Figure 1 shows a sample path of the evolution of a system with three pages and a single broadcast channel.

The weighted average waiting time over all users is defined by

$$\bar{W} = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \bar{W}_i$$

where $\bar{W}_i$ is the average waiting time for all page i requests and λ is the total request arrival rate to the system. By Little's law the average waiting time can be written as

$$\bar{W} = \frac{1}{\lambda}\sum_{i=1}^{N}(\bar{X}_i + \bar{Y}_i). \tag{4}$$

where $\bar{X}_i$ and $\bar{Y}_i$ are the average numbers of the requests currently in service or waiting for service in queue i, respectively. To avoid the difficulties associated with the average cost problems, instead of minimizing (4), we use the total discounted reward criteria and try to minimize the total discounted expected number of waiting requests defined as

$$J_\beta(\pi) = E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{N}(X_i(t)+Y_i(t))\right]. \tag{5}$$

Here π is the scheduling policy resulting in $J_\beta(\pi)$ and under mild conditions[6] $(1-\beta)J_\beta(\pi)$ approaches the optimal value for problem (4) as $\beta \to 1$. Equations (5) and (2), together with the initial condition $(X(0), Y(0))$, define the minimization problem

$$J^*_\beta(\pi) = \min_\pi E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{N}(X_i(t)+Y_i(t))\right]. \tag{6}$$

It can be shown[7] that $J_\beta(\pi)$ satisfies the equation

$$\begin{aligned}&(1-\beta)J_\beta(\pi) = \\ &\quad E\left[\sum_{i=1}^{N}(X_i(0)+Y_i(0))\right] + \beta E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{N}A_i(t)\right] \\ &\quad -\beta E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i\in d(t)}q_i(X_i(t)+Y_i(t)I[X_i(t)=0])\right].\end{aligned}$$

Therefore, since the first two terms of the right-hand side are independent of the policy π, the problem of minimizing $J_\beta(\pi)$ would be equal to the maximization problem

$$\begin{aligned}&\hat{J}_\beta(\pi) = \\ &\max_\pi E\left[\sum_{t=0}^{\infty}\beta^t\sum_{i\in d(t)}q_i(X_i(t)+Y_i(t)I[X_i(t)=0])\right].\end{aligned} \tag{7}$$

This problem is in fact a DP problem with decision space $D = \{d;\ d \subset \{1,2,\ldots,N\}\ \&\ |d| = K\}$ and state vector $s = \{x_1, y_1, \ldots, x_N, y_N\}$. Let us denote by S the state space of the problem. The expected reward for broadcast of pages in $d \in D$ at any state $s \in S$ is

$$r(s,d) = \sum_{i\in d}q_i(x_i + y_i I[x_i = 0]).$$

Also, if we show the optimal value function of this problem by $V(s)$, then $V(s)$ satisfies the optimality equation

$$V(s) = \max_{d\in D}\left[r(s,d) + \beta\sum_{s'\in S}p^d(s,s')V(s')\right] \quad \forall s\in S \tag{8}$$

where $p^d(s,s')$ is the probability of going from state s to state s' with decision d as defined by equations (2) and (3). In generic terms, this problem is a scheduling problem in a queueing system with N queues and K servers with different Geometric service times for different queues. The additional property which distinguishes this problem from the similar well-known scheduling problems [8], [9] is the fact that the servers are of the bulk service type with infinite bulk size. In other words, with a single service, all of the users of the serviced queue are cleared.

In this work, we limit our search to non-idling policies. Given the fact that there is no cost associated with each service, it can be shown that a non-idling optimal policy always exists. Moreover, for practical reasons, we are only interested in *index* policies. An index policy assigns a value (*index*) to each queue, which is only a function (*index function*) of the state and other parameters of that queue. For a system with N queues, the complexity of the policy would be $O(N)$ which is crucial for implementation purposes. This problem fits in the definition of the Restless Bandit problem introduced by Whittle[10], [11]. However, Whittle's approach can be only applied if the problem has certain properties. We will briefly introduce this method in the next section in such a way to fit our problem and prove the required properties after that. More information about this formulation can be found in [10] and [11].

IV. Restless Bandit approach

The additive form of the reward in our problem allows us to use the Linear Programming(LP) formulation of the DP problems[6] and convert problem (8) into the (dual) LP problem

$$Maximize\ \sum_{i=1}^{N}\left[\sum_{s\in S_i}r_i(s)z_i(s,1)\right] \tag{9}$$

subject to

$$\sum_{d\in\{0,1\}}z_i(s',d) - \sum_{s\in S_i}\sum_{d\in\{0,1\}}\beta p_i^d(s,s')z_i(s,d) = \alpha_i(s') \tag{10}$$

for $i = 1,\ldots,N$ and $s' \in S_i$. Here, $\alpha_i(.)$ is the initial probability distribution of the states, $r_i(s)$ is the reward for activating project i while in state s,

$$z_i(s,1) = E\left[\sum_{t=0}^{\infty}\beta^t I[x_i(t)=s\ ,\ i\in d(t)]\right]$$

and

$$z_i(s,0) = E\left[\sum_{t=0}^{\infty}\beta^t I[x_i(t)=s\ ,\ i\notin d(t)]\right]$$

where $I(.)$ is the indicator function of the event defined by its argument. In other words, $z_i(s,1)$ is the discounted expected value of the number of times queue i is served while at state s.

An additional constraint implicit to this scheduling problem is that at any time t, exactly K queues should be served. Whittle's relaxation assumes that instead of having exactly K projects activated at any time, only the time average of the number of activated projects be equal to K. This assumption in the discounted case can be stated as the following additional constraint to the dual problem

$$\sum_{i=1}^{N}\sum_{s\in S_i} z_i(s,1) = K/(1-\beta) \tag{11}$$

Using the Lagrangean Relaxation[12] method, the relaxed problem can be defined as[11]

$$\begin{aligned} Maximize\ & \sum_{i=1}^{N}\left[\sum_{s\in S_i} r_i(s) z_i(s,1)\right] \\ & + \nu\left(K/(1-\beta) - \sum_{i=1}^{N}\sum_{s\in S_i} z_i(s,1)\right) \end{aligned} \tag{12}$$

where ν is the lagrange multiplier. This problem can be also written as

$$Maximize \sum_{i=1}^{N}\left[\sum_{s\in S_i}(r_i(s)-\nu) z_i(s,1)\right] + K\nu/(1-\beta) \tag{13}$$

subject to constraint (10). Using the additive form of the objective function (13) and independence of the constraint on each project in (10), this maximization problem can be broken into N independent single-project maximization problems as

$$Maximize\ \sum_{s\in S_i}(r_i(s)-\nu) z_i(s,1) \tag{14}$$

subject to constraint (10) for $i = 1, 2, \ldots, N$. The optimal policy for each problem assigns one of the two idle and active actions to every possible state of the project.

Whittle showed that an optimal index policy for the relaxed problem exists, provided that the above single-project problems have a certain monotonicity property. This property states that for each project i; $i = 1, \ldots, N$, the set of states for which it is optimal to leave the project idle, increases from $\emptyset$ to the whole state space S_i as ν is increased from $-\infty$ to ∞. In that case, the value of the index for project i while at state x_i is the amount of service cost ν that puts state x_i on the border of idle and active regions. With the index function defined, the optimal policy for the relaxed problem is characterized by a threshold value ν^* specific to the problem and is to serve those queues with their indexes greater than ν^*. Inspired by this results, Whittle proposed a heuristic policy for the original problem that uses the same index function and serves the queues with K largest index values at any time. It was later shown[13] that this policy enjoys some form of asymptotic optimality.

In the next section we first investigate the indexability of our problem and then provide a method for computing the index function.

V. Some properties of a single controlled bulk service queue

Imagine one of our bulk service queues with arrivals and service times as before. The sub-problem we would like to consider for a single queue is to find the optimal policy that results in the maximum expected value of the discounted reward given a fixed service cost ν. The optimal policy is the optimal assignment of active (serving the queue) or passive (leaving the queue idle) actions to every state. More precisely, the objective function is:

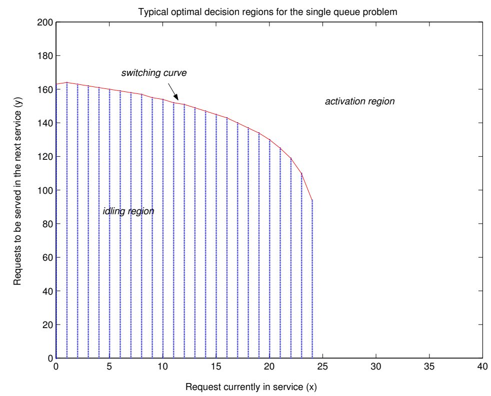

Fig. 2. Typical shapes of the idle and active regions for a single queue problem.

$$J_\beta = E\left[\sum_{t=0}^{\infty}\beta^t R(t)\right]$$

where $R(t)$ is the reward at time t, that is

$$R(t) = \begin{cases} x(t)-\nu & w.p.\ q & if\ \ d(t)=1\ \&\ x(t)>0 \\ y(t)-\nu & w.p.\ q & if\ \ d(t)=1\ \&\ x(t)=0 \\ -\nu & w.p.\ 1-q & if\ \ d(t)=1 \\ 0 & & if\ \ d(t)=0 \end{cases}$$

where $d(t)$ is the action at time t which is 1 if the queue is served and 0 otherwise and $(x(t), y(t))$ is the state of this system at time t as defined before.

We need to find if the optimal solution to this problem has the monotonicity property. Furthermore, we need to find the exact form of the policy and its switching curve to be able to calculate the index at any decision time. Figure 2 shows one example of the form of the optimal policy with the idle and active regions distinguished. All of our experimental results obtained by finding the optimal solution for this problem (via the Value Iteration method) in a large number of different settings (different values of λ,ν and q) confirm the monotonic expansion of the idling policy with increasing values of ν. Figure 3 shows a few results for this system with different ν values. The idling region for each case is the region that includes the origin and is surrounded by the x and y axes and the corresponding switching curve. The shape of the idling region is more or less the same in all results. It defines a policy which is of the threshold type in both x and y directions (except $x = 0$ points in some cases). The threshold property in the x direction can be stated as follows:

Property 1: If $d(x, y)$ is the decision defined by the optimal

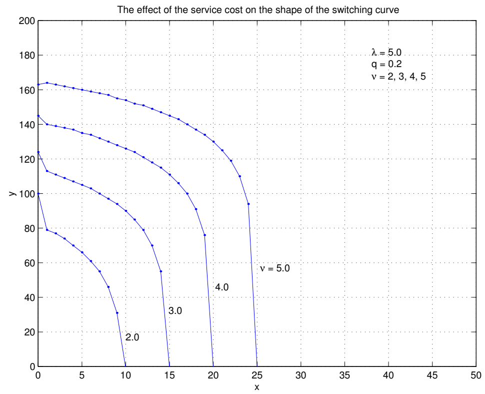

Fig. 3. Investigating the effect of the ν parameter on the optimal decision space for a typical system.

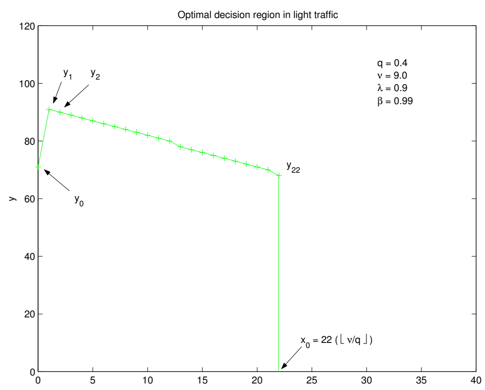

Fig. 4. Typical form of the optimal policy for a single queue problem in light traffic.

policy for state (x, y) we have

$$if\ d(x,y) = 1\ then\ d(x+i,y) = 1;\ \forall x > 0\, and\, i > 0; \quad (15)$$

Proof: [7].

Unlike the above case, it is very difficult to prove the threshold property in the y direction. Due to the bulk service property of the server, the usual techniques (e.g. [14], [15], [16], [17]) which take advantage of the sub(super)-modularity properties of the problem are not successful here. Nonetheless, the general methods for approaching this type of problems can still be used. We used induction on the Policy Iteration[6] steps to prove the properties of the switching curve for the light traffic case where the time slots are small enough so that the probability of having more than one arrival during a time slot is negligible. Briefly, the switching curve in this case has certain properties as follows. Defining $x_0 \triangleq \lfloor \frac{\nu}{q} \rfloor$, we have

1) $d(x, y) = 1\ ;\ \forall y\ ,\ \forall x > x_0$
2) $\exists y_0 > 0\ s.t.\ d(0, y) = \begin{cases} 1 & if \quad y \geq y_0 \\ 0 & if \quad y < y_0 \end{cases}$
3) $\forall\ 0 < x \leq x_0;\ \exists y_x > 0\ s.t.\ d(x, y) = \begin{cases} 1 & if \quad y \geq y_x \\ 0 & if \quad y < y_x \end{cases}$
4) $\forall\, 0 < x < x_0;\ 0 \leq y_x - y_{x+1} \leq \frac{1}{\beta a} + 1$
5) For $1 > \beta >> 0$ we have $y_0 > x_0$

where $a = \frac{q}{1-\beta(1-q)}$. Properties 1 and 4 basically describe the threshold property in the x direction in more details. Properties 2 and 3, deal with the threshold property in the y direction and property 5 is only a reasonable assumption to make the problem more tractable. Figure 4 shows the exact form of the switching curve for a typical light traffic setting which is calculated using the Value Iteration method. A general proof of these properties requires careful consideration of all cases that may happen for different values of the parameters. However, the details of the proof for a typical case can be found in [7]. The values of the different points of the switching curve can be found using a continues approximation of the curve which gives the values of the y_x for $x = 0, \ldots, x_0$ up to a rounding error. Since the top part of the curve is a straight line with slope $-1/\beta a$, we only need the y_0 and y_1 values to specify the curve. It can be shown that these values satisfy

$$y_1 - y_0 = \frac{1}{\beta a}\left[\frac{\nu}{q} - 1\right] - \frac{a p_1}{q} \quad (16)$$

$$y_0 = \frac{\nu}{q} + \frac{\beta p_1}{1-\beta}\left(1 - c^{-y_0}\right) \quad (17)$$

where $c = \frac{1-\beta p_0}{\beta p_1}$, $p_0 = 1 - p_1$ and p_1 is the rate in light traffic regime or equivalently, the probability of one arrival. It is then easy to show that the switching curve expands in both x and y directions as ν increases and therefore the monotonicity property holds for the single-queue problem. All of our numerical results (e.g. figure 3) suggest that the property also holds for higher arrival rates. However, due to the complicated form of the switching curve, we were not able to bring any analytical arguments for that. The above equations also allow us to calculate the value of the index function for every state. The index function at any state (x, y) is by definition the minimum amount of the service cost ν that puts the point (x, y) on the switching curve. for $x \leq x_0$, the corresponding y_1 value is

$$y_1 = y + (x-1)/\beta a. \quad (18)$$

Next, y_0 can be calculated by solving the following equation

$$y_1 + \frac{1}{\beta a}\left[\frac{\beta p_1}{1-\beta} + 1\right] + \frac{a p_1}{q} = y_0\left(1 + \frac{1}{\beta a}\right) + \frac{p_1}{a(1-\beta)} c^{-y_0}. \quad (19)$$

Having found the value of y_0, the corresponding ν is

$$\nu = q y_0 - \frac{q \beta p_1}{1-\beta}\left(1 - c^{-y_0}\right). \quad (20)$$

If the resulting ν turns to be smaller than qx, then x is on the right border of the idling region, i.e. $x = \nu/q$. For $x = 0$ case, the available y is in fact the y_0 value and ν is directly calculated from equation (20).

VI. Results

In this chapter we compare the results of our index policy with that of a number of other indexing policies. Unfortunately, to our knowledge, the broadcast scheduling problem with random file sizes has not been addressed before. Therefore, we don't have any immediate rival policy readily available for comparison. However, based on previous experiences, we chose a number of well-known policies used in simpler broadcast systems for comparison. In all experiments, we extended the index function defined above to general traffic cases by plugging in the actual rate in the equations. More detailed results and discussions can be found in[7].

A. *Random file sizes*

We compared our policy, which we named NOP(Near-Optimal Policy), to five other policies. We set up a system with 50 pages and simulated it under different settings with each policy. Other than the choice of the scheduling policy, every experiment had two other parameters namely, the average size of each of the pages and, the total request arrival rate of the system. In all experiments the assignment of the average size to each page was the opposite of the assignment of the request arrival rate to that page i.e., the longest page(in the average sense) was the least popular page and the shortest page the most. This rule is qualitatively consistent with many practical situations. Also, in all experiments we used the Zipf law to assign the individual request arrival rates to each queue given the total request arrival rate of the system. In order to investigate the effect of the distribution of the average file sizes on the performance of the policy, we performed our experiments for three choices of the distribution namely, Exponential, Uniform and Pareto distributions.
The policies that were used in the final set of experiments are:

- **NOP**: The light traffic approximate indexing policy derived in this paper.
- **PIP**: Arbitrary extension of the original PIP policy introduced in [5] extended for the new two dimensional setting
$$\nu_i = \frac{x_i + c_y y_i}{\sqrt{\lambda_i}}$$
- **HP1**: Heuristic policy defined as
$$\nu_i = \frac{(x_i + c_y y_i) q_i}{\sqrt{\lambda_i}}$$
- **HP2**: Heuristic policy defined as
$$\nu_i = \frac{(x_i + c_y y_i)\sqrt{q_i}}{\sqrt{\lambda_i}}$$
- **HP5**: Heuristic policy defined as
$$\nu_i = \frac{(x_i^2 + y_i^2 + x_i y_i)}{\lambda_i}$$
- **MRF**: Maximum-Request-First index defined as
$$\nu_i = (x_i + c_y y_i)$$

where c_y is a weight parameter and needs to be tuned for each policy. For a fair comparison, the c_y parameter was tuned for each policy through a large number of experiments and each policy was used with its own "optimal" c_y value.

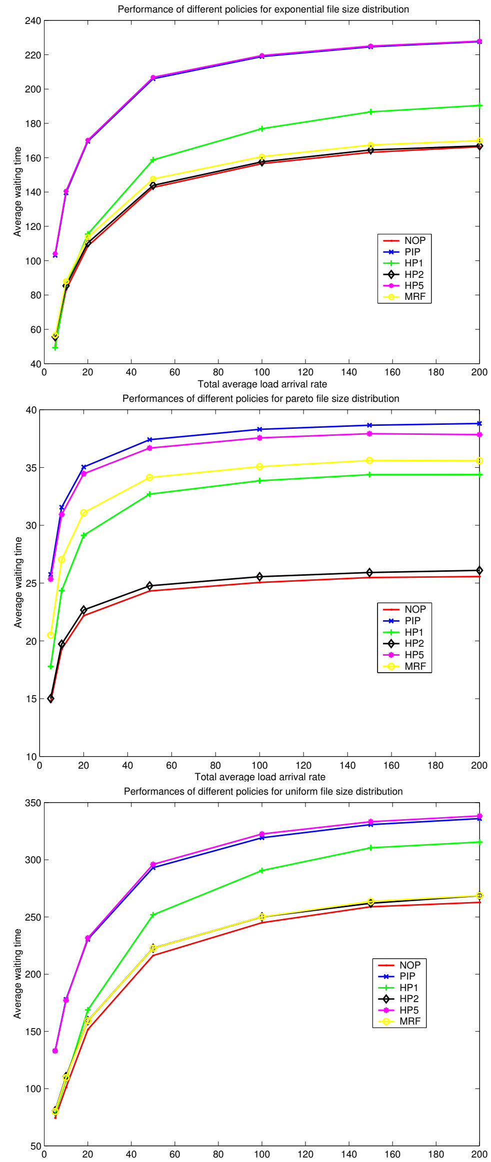

Fig. 5. Comparing the performance of different scheduling policies for different choices of the file size distribution.

Finally, the experiments were performed under seven choices of the total request arrival rates namely $\lambda = 5, 10, 20, 50, 100, 150, 200$. Figure 5 shows the results obtained from each policy under different simulation settings. The results show that the MRF and HP2 policies perform close to the NOP policy for exponential and uniform file size(average) distributions. However, only HP2 remains close to NOP for Pareto distribution with NOP slightly outperforming HP2 in all cases.
To summarize, we found that the optimized versions of all candidate policies are inferior to NOP. However, according to the results, HP2 (with $c_y = 0.5$) can be used as a low complexity alternative for NOP for practical purposes.

B. Fixed file sizes

In some broadcast systems, the files to be broadcasted are locally stored in the system and therefore the system knows their exact sizes. The cache broadcast systems are examples of this type of systems. Unfortunately, the analytical approach to this problem proved to be too complicated. However, in the absence of an analytical solution, it is possible to modify the NOP policy for this case and it is constructive to compare its performance with those of some other heuristic policies.

The NOP policy can be easily modified for use in the fixed file length case by replacing the average file size parameter $1/q_i$, with the exact file size value L_i in the previous formulas. We compared the performance of the NOP with MRF, HP1,HP2 and HP5. It can be shown[7] that the HP5 index is in fact an approximation of the Longest-Total-Wait-First policy extended for the systems with preemption. Therefore, we use the LTWF1 name for that policy in this case. We also tested another version of LTWF that take advantage of the number of transmitted segments of each file and is defined as $LTWF2 = LTWF1 - lx$, where l is the number of transmitted segments of the file. The HP1 and HP2 in this case replace the q_i factor with $1/L_i$ wherever it appears and we also used the optimum values of c_y found in the previous experiments for them. The set of experiments in this section are similar to the last section, i.e., we found the average waiting times resulted by all policies for seven different arrival rates and three choices of the file size distribution. Since the file sizes are deterministic, assuming different distributions for them has a better meaning in this case. Figure 6 shows the results. The first observation is the poor performance of the LTWF policies compared to the other policies. This can be an indication that this type of policy, unlike for the equal file size case, is not optimal (or even close to optimal) for this system. Also, in all graphs, HP1 performs poorly compared to the other policies. NOP and HP2 always perform very close to each other. As a conclusion, we can say that for practical purposes, HP2 can be used as a low complexity policy for a wide range of file size distributions.

A legitimate question is the validity of the above results for higher request arrival rates. In fact, since our index function was found by extending the index function for the light traffic regime to arbitrary rates, we expect the performance of this policy to degrade for higher rates. However, the use of the *pull* broadcast systems with their additional cost for a return channel is only justified for relatively lower traffic loads and it has been shown that in practice for higher rates, both of the *pull* and *push* systems have the same performance and therefore the latter is preferred due to its lower cost. A well-designed broadcast system(with a return channel available) uses the *pull* delivery for files with low to medium request rates and the *push* delivery for other files.

In general, the results suggest that our approximate index policy performs very close to the optimal policy. The dynamic optimization approach also allows us to address other variations of the problem. For example, an immediate extension of

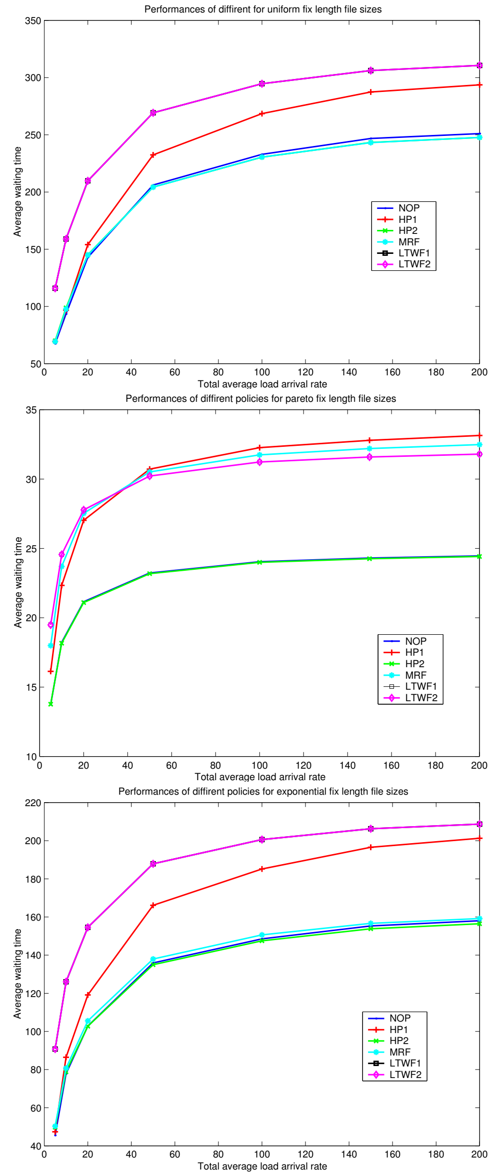

Fig. 6. Performance of different policies for different choices of the file size distribution.

this problem where distinct weights are assigned to the average waiting times of the users of different files, can be addressed in a straightforward manner. We are currently trying to address similar problems with time constraint and will publish the results in future publications.

VII. Conclusion

In this paper, we used the Restless Bandit problem formulation to address the problem of optimal scheduling in broadcast systems with random file lengths. We showed that the problem is indexable and derived an equation for the index function for the light traffic regime and extended it to be used for moderate traffic cases and fixed file size situations. At the same time, we chose several well-known, as well as new, heuristic policies and tried to optimize them for use in our experiments.

All of the results strongly suggest that our policy is a near-optimal policy. Also, one of our heuristic policies proved to perform as good as the original policy and can be used as a low complexity policy for practical applications. Nevertheless, the optimization approach remains valid as a powerful method for investigating other variations of the problem.

References

[1] Majid Raissi-Dehkordi and John S. Baras, "Broadcast scheduling in information delivery systems," *Proceedings of IEEE GLOBECOM2002*, Nov. 2002, Taipei, Taiwan.

[2] J. W. Wong and M. H. Ammar, "Analysis of broadcast delivery in a videotext system," *IEEE Trans. on computers*, Vol. C-34, No. 9, pp863-966, 1985.

[3] H. D. Dykeman, M. H. Ammar, and J. W. Wong, "Scheduling algorithms for videotex systems under broadcast delivery," *IEEE Int. Conf. on Comm. ICC86*, Vol. 3,pp1847-51, 1986.

[4] D. Aksoy and M. Franklin, "Scheduling for large-scale on-demand data broadcasting," *Proc. INFOCOM 98*, Vol. 2, pp651-9, 1998.

[5] C. Su and L. Tassiulas, "Broadcast scheduling for information distribution," *Proc. of INFOCOM 97*, 1997.

[6] M. Putterman, *Markov Decision Processes : Discrete Stochastic Dynamic Programming*, Wiley, New York., 1994.

[7] Majid Raissi-Dehkordi, "Broadcast scheduling in information delivery networks," *Doctoral dissertation, Department of Electrical and Computer Engineering*, University of Maryland at College Park, 2002.

[8] P. Varaiya, J. Walrand, and C. Buyukkoc, "Extensions of the multi-armed bandit problem," *IEEE Transactions on Automatic Control AC-30*, pp426-439, 1985.

[9] J. Baras, D. Ma, and A. Makowski, "K competing queues with geometric requirements and linear costs: the c-rule is always optimal," *J. Systems Control Lett.*, Vol. 6, pp173-180, 1985.

[10] 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.

[11] Jose Nino-Mora, "Restless bandits, partial conservation laws and indexability," *http://www.econ.upf.es/ ninomora/*.

[12] J. Beasley, "Lagrangean relaxation," *Chapter 6: Lagrangean relaxation. In Colin R. Reeves, editor, Modern Heuristic Techniques for Combinatorial Problems. Blackwell Scientific Publications*, 1993.

[13] Richard R. Weber and Gideon Weiss, "On an index policy for restless bandits," *J. Appl. Prob., Vol. 27*, pp637-648, 1990.

[14] W. Lin and P.R. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Transactions on Automatic Control, Vol.29*, pp696-703, 1984.

[15] B. Hajek, "Optimal control of two interacting service stations," *IEEE Transactions on Automatic Control, AC-29(6)*, pp491-499, 1984.

[16] G. Koole, "Structural results for the control of queueing systems using event-based dynamic programming," *Queueing Systems, Vol.30*, pp323-339, 1998.

[17] I. Viniotis and A. Ephremides, "Extension of the optimality of the threshold policy in heterogeneous multiserver queueing systems," *IEEE Transactions on Automatic Control, Vol.33*, pp104-109, 1988.