TECHNICAL RESEARCH REPORT

A Dynamic Optimization Approach to the Scheduling Problem in Satellite and Wireless Broadcast Systems

by M. Raissi-Dehkordi, J.S. Baras

CSHCN TR 2002-34
(ISR TR 2002-68)

A dynamic optimization approach to the scheduling problem in satellite and wireless broadcast systems*

Majid Raissi-Dehkordi, John S. Baras
Institute for Systems Research
University of Maryland at College Park

Abstract

The continuous growth in the demand for access to information and the increasing number of users of the information delivery systems have sparked the need for highly scalable systems with more efficient usage of the bandwidth. One of the effective methods for efficient use of the bandwidth is to provide the information to a group of users simultaneously via broadcast delivery. Generally, all applications that deliver the popular data packages (traffic information, weather, stocks, web pages) are suitable candidates for broadcast delivery and satellite or wireless networks with their inherent broadcast capability are the natural choices for implementing such applications.

In this report, we investigate one of the most important problems in broadcast delivery i.e., the broadcast scheduling problem. This problem arises in broadcast systems with a large number of data packages and limited broadcast channels and the goal is to find the best sequence of broadcasts in order to minimize the average waiting time of the users.

We first formulate the problem as a dynamic optimization problem and investigate the properties of the optimal solution. Later, we use the bandit problem formulation to address a version of the problem where all packages have equal lengths. We find an asymptotically optimal index policy for that problem and compare the results with some well-known heuristic methods.

Keywords: broadcast scheduling, satellite data delivery, restless bandit problem.

*Research partially supported by NASA cooperative agreement NCC3-528, by MIPS grant with Hughes Network Systems, and by, Lockheed Martin Networking Fellowship all with the Center for Satellite and Hybrid Communication Networks at the University of Maryland at College Park.

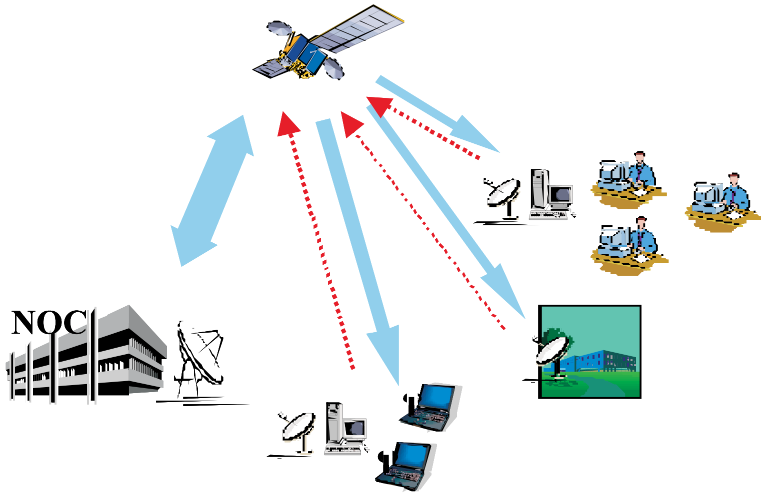

Figure 1: Typical architecture of a satellite information delivery system.

1 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 on some cellular phones is an example of this type of applications. The WWW traffic, which constitutes about 50% to 70% of the Internet traffic [12, 28], 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 like *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. In fact, the broadcast transmission via either wired or wireless media, makes a more efficient use of the bandwidth by not sending the information through any path more than once. However, the wireless media, due to their inherent physical broadcast capability, have the additional advantage of forming a one-hup structure where all the receivers share the single download link and receive the requested information at the same time. Throughout this report, we use the term *broadcast system* to refer to this type of system with physical broadcast capability. Figures 1 and 2 show two examples of these type of systems. In both systems, we assume that all the users who are waiting for a specific

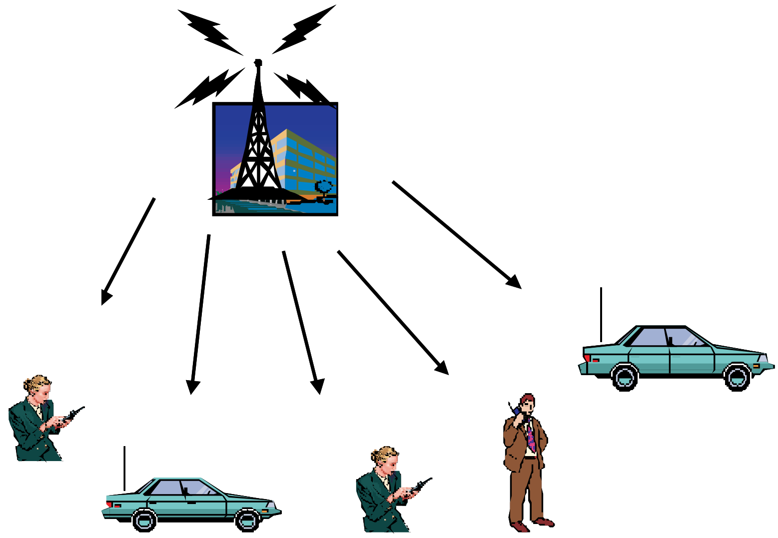

Figure 2: Typical architecture of a wireless information delivery system.

package will be directly served with a single transmission of that package over the broadcast channel. This property, solves one of the major problems in the design of any information delivery system, which is the scalability problem. The scalability of a system depends on the relation between the resources of the system and its number of users. In a satellite information delivery system, or any other system with broadcast capability, the main resource of the system, which is the downlink, is insensitive to the number of users and the number of users can be increased without any need for an increase in the bandwidth of the downlink[1]. Therefore, the satellite and wireless environments provide highly scalable systems for data delivery applications. Some of the popular data delivery applications are as follows:

- **Delivery of popular information packages:** In this type of service, certain number of time-sensitive information pages like stocks, weather or traffic information are broadcast by the system to the users upon their request. In this application, the packages usually have a short fixed length. Also, in some cases, deadlines may be introduced for some of the time-sensitive packages. The main concern of the provider is to schedule the broadcast of the information packages in order to minimize a measure of the delay experienced by the users of different packages. Many cellular phones are currently capable of receiving the information like news, weather and so on and the use and variety of these systems is expected

[1]There are of course other practical issues like the uplink bandwidth, geographical coverage, ... that need to be taken into account. However, considering the highly asymmetric nature of the data and assuming that the users are within the coverage area of the system, the downlink becomes the main bottleneck of the system

to grow with the advancements in the broadband wireless systems and mobile computing field.

- **Cache broadcast:** This application is a method for fast delivery of the WWW service from certain web servers to their users. The use of local caches to locally store the popular information in various parts of the network is a common practice for reducing the response time of the system. A cache enables the system to locally respond to the requests for popular web pages without the need for accessing the main server on the Internet. In satellite networks, the cache is usually located at the Network Operation Center(NOC). Having a cache installed in a satellite Internet delivery system, the performance can be further increased by broadcasting the cached pages to all users who need the pages at the same time or within certain time interval instead of serving all of them individually.
- **Webcasting:** This type of system is in fact quite similar to cache broadcasting. The users of this system are the information providers or companies who already have the necessary information for their users or employees on the WWW and like to provide them with fast access to those pages. The difference between this service and the regular Internet service is the fact that this service does not necessarily provide access to the Internet, and the web site contents are locally stored in the ground station. This system, like other WWW applications, works as a client/server application and there is some type of uplink to transfer the requests to the server(ground station) but the transmitted data is available to all members of the group.

There are already a number of companies (e.g. Hughes Networks System [30], Cidera [5]) that offer various data delivery services using satellite links and their number is expected to grow with the advances in the information technology and the increasing number of users. The advances in wireless networks and the advent of mobile computing applications suggest that there will be more room for taking advantage of the potential benefits of broadcast systems for making more efficient networks.
The two main architectures for broadcast delivery are the one-way(or *Push*) and the two-way(or *Pull*) systems. The two systems differ in the lack or presence of a return channel to transfer the user requests to the server. 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. A *pull* system is potentially able to achieve a better performance than a *push* system but the cost of a return channel can generally overshadow this performance improvement. For this reason *hybrid* architectures, those that combine *push* and *pull* systems, are commonly suggested in the literature [17, 13, 9]. The main problem to address in both of the above broadcast methods is the

scheduling of data transmissions. As we will mention in the next section, the problem of scheduling in a *push* system is solved to a large extent. However, to our knowledge, the problem of finding the optimal broadcast scheduling policy for a *pull* system has not been solved yet.

Based on the nature of the applications supported by a data delivery system, different performance metrics can be used to evaluate the performance of the system. However, in most cases, the average waiting time is the parameter that is usually chosen. Other parameters like the worst-case waiting time can also be of interest when strict deadlines are assigned to the packages. In this work, we try to minimize the *weighted* average waiting time of the users to allow more flexibility in assigning *soft* priorities to the packages.

This report is organized as follows. Following the literature review in section 2, the mathematical formulation of the problem is introduced in section 3. section 6 is dedicated to reviewing the principles of our approach that is based on the *Restless Bandit* problem formulation [33]. After proving the required properties, we find both the exact and approximate index policies for that problem. Finally, in section 9, a detailed investigation of the performance of our policy compared to other well-known heuristic policies is presented.

It is worth mentioning that although our work on broadcast scheduling is motivated by the problems in broadcast communication systems, our results are not limited to communication applications. This work can be considered as the generic problem of finding the optimal scheduling policy in a queueing system with a bulk server of infinite capacity. It is easy to think of some applications of this problem in transportation industry which has been the origin of many queueing and scheduling problems.

2 Related work

The series of works by Ammar and Wong are probably the first papers addressing the broadcast scheduling problem in detail. In [21, 22] they consider various aspects of the *push* systems by analyzing the problems associated with a Teletext system. They derive a tight lower bound for the average waiting time of the users of a Teletext system with equal-sized packages of data. They also showed that the optimal scheduling policy is of the cyclic type where the frequency of appearance of every page in every broadcast cycle is directly related to the square root of the arrival rate of the requests for that page. They presented a heuristic algorithm for designing the broadcast cycle based on the arrival rates. Vaidya and Hameed [31, 23] extended the so called square root formula to cover *push* systems with unequal page sizes and also considered the systems with multiple broadcast channels. They showed that the appearance frequency of a page in the broadcast cycle is inversely related to the square root of its length and proposed an on-line algorithm for transmitting the requested pages. Moreover, they investigated the role of channel errors and made provisions for the error probability in their algorithm. Su and Tassiulas [29] proposed an MDP formulation of both the *push* and *pull* delivery systems. They showed that the optimal policy for a *push* system with two pages is of the cyclic type and derived an equation for the optimal content of every cycle. They also proposed a heuristic indexing policy for the *push* broadcast scheduling that dynamically chooses the page to be broadcasted at the beginning of every broadcast period. In a separate work, Bar-Noy [1] finds the optimal broadcast schedule for a *push* system with two pages under different choices of the request arrival processes while allowing the page lengths to be different. There are also other papers [10, 11] which address the scheduling problem for more complicated variations of a *push* system by proposing different data delivery schemes.

Despite the wealth of resources about the *push* systems, the number of works addressing the *pull* broadcast systems is limited. However, none of those papers (except [8] to our knowledge) have tried to find the *optimal* scheduling policy and most of them have suggested heuristic algorithms which despite their good performances in some cases [7, 29] do not contain the notion of optimality. In [8], the problem of finding the optimal scheduling policy for a *pull* system is formulated as a dynamic programming problem. They attempted to numerically solve the problem for small systems and made a number of conjectures about the properties of the optimal policy based on the results. This work might be the first analytic approach for solving the *pull* scheduling problem. However, the problem of finding the optimal policy still remains open. In [17], a number of heuristic policies for a *pull* system are proposed and their resulting average waiting times are compared. Valuable observations about the performances of both *push* and *pull* systems are also made in that paper. In [29], an index policy called the *Performance Index Policy* (PIP) was introduced. The PIP index associated with each page is a function of both the arrival rate and the number of pending requests for that page. After experimental tuning of the parameter of that function for the case with Zipf distribution of the arrival rates, the PIP policy produced satisfying results in a number of experiments. The work by Aksoy and Franklin [7] proposed another index policy named RxW and reported a performance comparable to PIP in different experiments. The two above works are probably the best known scheduling methods for a *pull* system. However, the distance between their performances and that of an optimal policy is unknown. All of the above works only consider the case where all pages are of equal importance and have equal sizes and do not apply to cases like cache broadcasting where the pages can have unequal lengths. Moreover, due to the complete heuristic nature of the algorithms, it is difficult to extend them to other possible scenarios. This is the main motivation behind our work. In this thesis, we address the scheduling problem in a *pull* system. We aim to find a near-optimal (with respect to the weighted average waiting time) scheduling policy via optimization methods and also provide a benchmark for evaluating current and possibly future heuristic algorithms. We have approached the scheduling problem from a dynamic optimization point of view. This formulation is similar to the formulation in [8] and [29] but instead of using numerical methods for very simplified versions of the problem or using this formulation in its initial form to find a few properties of the unknown optimal policy, our goal is to reach an analytic solution and present an index policy through optimization arguments. Using the *Restless Bandit* [33] formulation, our approach naturally addresses the systems with multiple broadcast channels, or prioritized pages and also provides guidelines for the case with unequal page sizes.

3 Problem formulation

In our formulation, we denote by $N(>1)$, the number of information packages stored in the system. In this section we present the formulation of the case where all packages have equal sizes. This assumption is also made in [29, 8, 7] and most of the other works on this subject and is a reasonable assumption for many applications. Throughout this thesis, we will use the terms *page*, *package*, and *information package* interchangeably to simplify the notation. The fixed page size assumption naturally introduces a time unit that is equal to the time required to broadcast a page on a channel and it can be set to one without loss of generality. All of the broadcast times therefore, start at integer times denoted by t; $t = 0, 1, \ldots$. Here we assume that the system has $K(1 \leq K < N)$ identical broadcast channels. In a *pull* broadcast system, the system receives the requests for all packages from the users and based on this information the scheduler decides which pages to transmit in the next time unit in order to minimize the average waiting time over all users.

For the systems with a large number of users it is reasonable to assume that the requests for each page i; $i = 1, \ldots, N$ arrive as a Poisson process and denote by λ_i the rate of that process. The waiting time for every request is the time since the arrival of the request to the system until the end of the broadcast of the requested page. Due to the Poisson assumption for the request arrival process and given that a request arrives in the interval $[t, t+1)$, its exact arrival time would have a uniform distribution over this interval. Therefore, the waiting time from the time of arrival till the start of the next broadcast cycle $(t+1)$ has a mean of $1/2$ which, together with an integer part (i.e. number of time units till the beginning of the broadcast of the requested page) make the actual waiting time of the request. This constant value can be omitted from our calculations without loss of generality and we can assume that the requests for every page i arrive at discrete time instants t as batches with random sizes having $Poisson(\lambda_i)$ distribution. The system therefore, can be shown by a set of N queues where each queue corresponds to one of the packages and holds all the

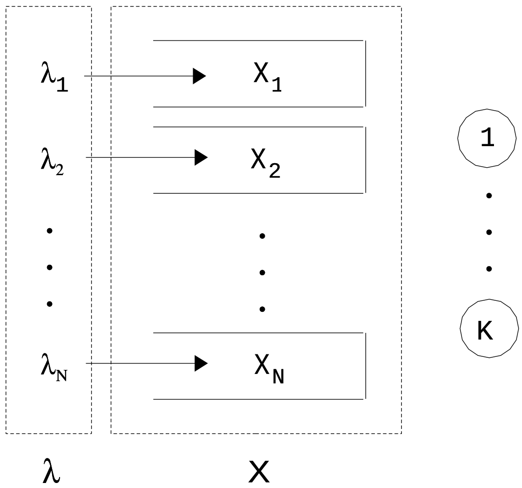

Figure 3: The pull type broadcast as a queuing system.

pending requests for that package, and K servers as in figure 3. Due to the broadcast nature of the system, the queues are of the bulk service type [4] with infinite bulk size i.e. the requests waiting in a queue will be served altogether once the queue is serviced. The state of this system at each time t is shown by $\mathbf{X}(t) = (X_1(t), X_2(t), \ldots, X_N(t))$: where $X_i(t)$ is the number of pending requests for page i at time t. Also, let's denote by $\mathbf{A}(t) = (A_1(t), A_2(t), \ldots, A_N(t))$ the discrete-time request arrival process for all pages where $A_i(t)$ represents the number of requests for page i during $[t, t+1)$ time interval. $X_i(t)$; $i = 1, \ldots, N$ is a Markov chain which evolves as

$$X_i(t+1) = X_i(t) - X_i(t)\mathbf{1}[i \in d(t)] + A_i(t) \tag{1}$$

where $d(t) \subset \{1, \ldots, N\}$ is the set containing the indices of the K pages broadcasted at time t. Figure 4 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

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

where $\bar{W}_i$ is the average waiting time for page i requests and λ is the total request arrival rate to the system. The c_i coefficients are the weights associated with the pages to allow more flexibility in assigning soft priorities to the pages. By Little's law the average waiting time can be written as

$$\bar{W} = \frac{1}{\lambda} \sum_{i=1}^{N} c_i \bar{X}_i. \tag{2}$$

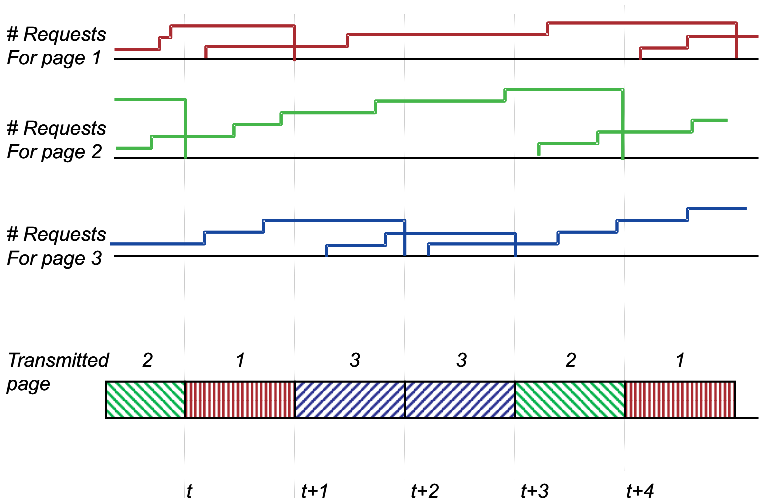

Figure 4: Sample path of a system with three pages.

where $\bar{X}_i$ is the average number of requests in queue i and the constant λ term can be omitted in the minimization problem. Due to the discrete-time nature of the system, and to avoid technical difficulties associated with the DP problems with average reward criteria, instead of minimizing (2), we use the expected discounted reward criteria defined as

$$J_\beta(\pi) = E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i X_i(t)\right] \tag{3}$$

where π is the scheduling policy resulting in $J_\beta(\pi)$. Equations (3) and (1), with the initial condition $X(0)$, define the following DP problem with J^* denoting the optimal value defined as

$$J^*_\beta(\pi) = \min_\pi E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i X_i(t)\right]. \tag{4}$$

We have shown in appendix A that $J_\beta(\pi)$ satisfies the equation

$$(1-\beta)J_\beta(\pi) = E\left[\sum_{i=1}^{N} c_i X_i(0)\right] + \beta E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i A_i(t)\right] - E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i\in d(t)} c_i X_i(t)\right] \tag{5}$$

where $d(t)$ is the set of the pages broadcasted at time t. 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

$$\hat{J}^*_\beta(\pi) = \max_\pi E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i\in d(t)} c_i X_i(t)\right]. \tag{6}$$

This problem is in fact a DP problem with state space $\mathbf{S} = (S_1, \ldots, S_N)$ where $S_i = 0, 1, \ldots;\ i = 1, \ldots, N$ and decision space $D = \{d;\ d \subset \{1, 2, \ldots, N\}\ \&\ |d| = K\}$ with

$|d|$ denoting the cardinality of set d. The decision space D is the set of all possible K tuples of the indices 1 through N. The reward function for broadcasting of pages in $d \in D$ at state $s = \{x_1, \ldots, x_N\} \in S$ is

$$r(s,d) = \sum_{l \in d} c_l x_l \tag{7}$$

and a stationary policy is a function $\pi : S \mapsto D$ that maps every state to a decision value. It can be shown (Theorem 6.10.4 [26]) that under mild conditions on the reward function (which includes our linear function) and given the assumption of finite arrival rates, the L operator defined as

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

is a contraction mapping and therefore this DP problem with unbounded rewards has an optimal solution. Here, $p^d(s,s')$ is the stationary transition probability of going from state s to state s' under decision d and $V(s)$; $s \in S$ is the value function associated with the optimal solution. This function satisfies the optimality equation

$$V(s) = L(V(s)) \quad \forall s \in S. \tag{9}$$

This maximization problem is the problem we will address in the sequel to find a non-idling, stationary optimal policy for the *pull* broadcast environment. What we are specially interested in is an index-type policy that assigns an index $\nu_i(x_i)$ to queue i; $i = 1, \ldots, K$ and the optimal decision is to service the queue(s) with the largest index value(s). If the index for each queue only depends on the state of that queue, the computation load for every decision would be of order N which is important from a practical point of view for systems with a large number of stored pages.

Since in our formulation there is no cost for serving a queue, we expect the optimal policy to serve exactly K non-empty queues at each time. This can be better seen via a sample path argument. Suppose that $\{d_1, d_2, \ldots\}$ is the decision sequence dictated by policy π when the system starts from initial state $\mathbf{x}$ and the arrivals occur according to sequence $A = \{a_1, a_2, \ldots\}$. Suppose that, at some time instant t, there are $M > K$ non-empty queues in the system and π opts to serve $K' < K$ of them. We can construct a new policy π^* which serves the same queues as π plus $K - K'$ additional non-empty queues. Let's suppose one of the additional queues j have x_j requests at time t, and $t' > t$ is the earliest time policy π will serve that queue. In this system, the reward function is linear and the arrivals are independent of the state of the system. Hence, if $S_A^{\pi}(\mathbf{x})$ is the total discounted reward generated by policy π with initial state $\mathbf{x}$ and arrival sequence A, then we will have $S_A^{\pi^*}(\mathbf{x}) \geq S_A^{\pi}(\mathbf{x}) + c_j x_j (\beta^t - \beta^{t'})$. This argument shows that for every idling policy π, we can construct a non-idling policy π^* which will result in a greater total discounted reward for every sample path and therefore, in a

greater expected discounted reward. Henceforth, from now on, we only focus on the set of non-idling policies for finding the optimal policy.
Also, it should be mentioned at this point that in the discrete-time setting of the above problem, the arrival process is only modelled as an i.i.d. sequence with a specific pmf. Although our initial Poisson assumption for the request arrival process implies that the corresponding pmf would be that of a Poisson distribution, our analysis below is quite general and holds for other distributions as well.

3.1 Properties of the optimal policy

As in many other problems, the DP formulation of our problem provides a mathematical characterization of the optimal solution but does not necessarily lead to a closed-form or analytical expression for it. The range of the results that we can get by working with equation (9) is limited to a few properties of the optimal solution. However, since the methods for proving those properties are similar to what we will use in the following sections where we introduce our main approach for solving this problem, it is constructive to point to some of the results in this section.

The properties we tried to prove show that the optimal policy is of the threshold type and the decision surfaces (in the N-dimensional space with each dimension representing the length of one queue) are non-decreasing with respect to all coordinates. This approach has a limited range and only gives us ideas about the form of the optimal policy. We first need the following lemma to prove the properties.

Lemma 1 *Let $S_p^d(\mathbf{x})$ denote the resulting discounted reward sum when the initial condition is $\mathbf{x}$ and arrivals occur as sample path p and the fixed(independent of state) decision sequence d is applied to the system. Then we have*

$$S_p^d(\mathbf{x}) \leq S_p^d(\mathbf{x} + \mathbf{e}_i) \leq c_i + S_p^d(\mathbf{x}). \tag{10}$$

where $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0)$ is the unit vector in R^N with ith element equal to one.

Proof: Consider two identical systems one with initial condition $\mathbf{x}$ and the other with initial condition $\mathbf{x} + \mathbf{e}_i$ defined as above. If the same fixed policy is applied to these two systems, the reward would be the same before the first broadcast of i. At that point, the second system receives a reward that is $1c_i$ units more than that received by the first system. Since the dynamics of the system forces the length of the serviced queue to zero, it in fact erases the memory of the queue after each service. Therefore, the resulting rewards even for queue i in both systems would be the same afterwards. Therefore, the first inequality holds($c_i > 0$). The presence of the discount factor $0 < \beta < 1$ causes the additional instantaneous reward in the second system to result in at most a c_i unit difference between the two discounted sum of the rewards(if i is served at time $t = 0$), hence the second inequality holds.

The first property can be proved using the above lemma. suppose

- $\mathbf{x} = (x_1, x_2, \ldots, , x_N)$
- $\mathbf{y} = \mathbf{x} + \mathbf{e}_i;\;\; i \in 1, 2, \ldots, N$

then

Theorem 1 *For* $\mathbf{x}$ *and* $\mathbf{y}$ *defined as above and function* $V(.)$ *being the value function of the optimal policy of our maximization problem, we have*
(a) $V(\mathbf{y}) \leq V(\mathbf{x}) + c_i$.
(b) $V(\mathbf{x}) \leq V(\mathbf{y})$

Proof: Let d^* be the optimal policy and denote by $d_p^{\mathbf{x}}$ the deterministic sequence of decisions dictated by d^* when the arrivals occur according to a deterministic sample path p and the initial condition is $\mathbf{x}$. According to lemma 1 we have

$$S_p^{d_p^{\mathbf{y}}}(\mathbf{y}) \leq S_p^{d_p^{\mathbf{y}}}(\mathbf{x}) + c_i \tag{11}$$

If we take the expectation of both sides with respect to the sample path probability $P(p)$, we get

$$V(\mathbf{y}) \leq c_i + \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{x}). \tag{12}$$

Also, according to the definition of optimality of policy d^* we have

$$V(\mathbf{x}) = \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{x}) \geq \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{x}). \tag{13}$$

inequality (a) follows by combining the two above results.
Also, according to lemma 1 we have

$$S_p^{d_p^{\mathbf{x}}}(\mathbf{x}) \leq S_p^{d_p^{\mathbf{x}}}(\mathbf{y}) \tag{14}$$

If we take the expectation of both sides with respect to the sample path probability $P(p)$, we get

$$V(\mathbf{x}) \leq \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{y}). \tag{15}$$

Also, according to the definition of optimality of policy d^* we have

$$V(\mathbf{y}) = \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{y}) \geq \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{y}). \tag{16}$$

Hence inequality (b) follows.

The second property can also be proved using the following discussion.

Theorem 2 *If d^* is the optimal policy and $d^*(\mathbf{x}) = i$ then $d^*(\mathbf{y}) = i$ with $\mathbf{x}$ and $\mathbf{y}$ defined as above.*

Proof: Since i is the optimal policy for state $\mathbf{x}$, we have

$$c_i x_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_j x_j + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N. \tag{17}$$

We need to show

$$c_i y_i + \beta \sum_A P(A) V(\mathbf{y} + A - y_i \mathbf{e}_i) \geq c_j y_j + \beta \sum_A P(A) V(\mathbf{y} + A - y_j \mathbf{e}_j) \quad j = 1, \ldots, N \tag{18}$$

or since $\mathbf{y}$ is different from $\mathbf{x}$ just in the ith element,

$$c_i x_i + c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_j x_j + \beta \sum_A P(A) V(\mathbf{y} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N \; j \neq i. \tag{19}$$

From 17 we have

$$c_i x_i + c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_i + c_j x_j + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N. \tag{20}$$

Also , from lemma 1 part(a) we have

$$V(\mathbf{y} + A - x_j \mathbf{e}_j) \leq c_i + V(\mathbf{x} + A - x_j \mathbf{e}_j) \tag{21}$$

or

$$\beta \sum_A P(A) V(\mathbf{y} + A - y_j \mathbf{e}_j) \leq c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j). \tag{22}$$

From (20) and (22), equation (19) follows, that proves the theorem.

The last property shows that the optimal policy is of the threshold type. In other words, once i becomes the optimal decision for an state $\mathbf{x}$, it remains the optimal decision for all states $\mathbf{x} + k\mathbf{e}_i; \;\; k = 1, 2, \ldots$.
As it was mentioned before, equation (9) only reveals limited properties of the optimal policy. Our main approach in this report requires some background from the Bandit problems and also a few properties of the bulk service queueing systems. Therefore, we explain our main approach in section 6 after providing the necessary material in the following section.

4 Some properties of a single controlled bulk service queue

Queues with infinite bulk service capability posses a number of interesting properties. A generic single-server bulk service queue with Poisson arrivals and general service distribution is shown by the $M/G_a^b/1$ notation [4] where the subscript a is the minimum number of customers in the queue needed by the server to start a service and superscript b is the bulk size i.e., the number of customers which will be served by each service. Since here we deal with controlled queues in a dynamic programming setting, we do not present our results about the regular continuously serviced bulk service queues. However, some results are included in appendix D for the interested reader. Imagine one of our bulk service queues with Poisson arrivals and constant service times as before. If we assume that all the arrivals that arrive during a service period are counted only at the end of that period, the system would be a pure discrete-time system. 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 that the reward obtained by serving the queue at any time is equal to the number of customers in the queue and there is also a fixed cost ν for each service. The optimal policy is the optimal assignment of active or passive actions to every state. More precisely, the objective function is:

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

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

$$R(t) = \begin{cases} cs(t) - \nu & if \;\; d(t) = 1 \\ 0 & if \;\; d(t) = 0 \end{cases}$$

and $d(t)$ is the indicator function that is 1 if the queue is served and 0 otherwise. $s(t)$ is the state of this system at time t and is the number of customers in the queue waiting to be serviced. A property that is crucial in later discussions is as follows

Property 1 *The optimal policy is of the threshold type with respect to the state space ordering. In other words, if it is optimal to serve the queue at state x, then it is also optimal to serve the queue if it is at any state $y > x$.*

The proof of this property can be found in appendix B where we use an induction argument. This property shows that for every fixed value of the service cost ν^*, the set of states where it is optimal not to serve the queue ($S^0(\nu^*)$) contains all the states less than or equal to a threshold state $s_{th}(\nu^*)$. The optimal policy is so far to compare the state of the queue at each decision instant with the threshold state and serve the queue if the state is larger then the threshold. The threshold state $s_{th}(\nu^*)$ is the largest state for which it is still optimal to leave the queue idle. The threshold state also has the following property

Property 2 *For the single bulk service queue discussed in this section, the threshold state $s_{th}(\nu)$ is a non-decreasing function of the service cost ν (figure 5).*

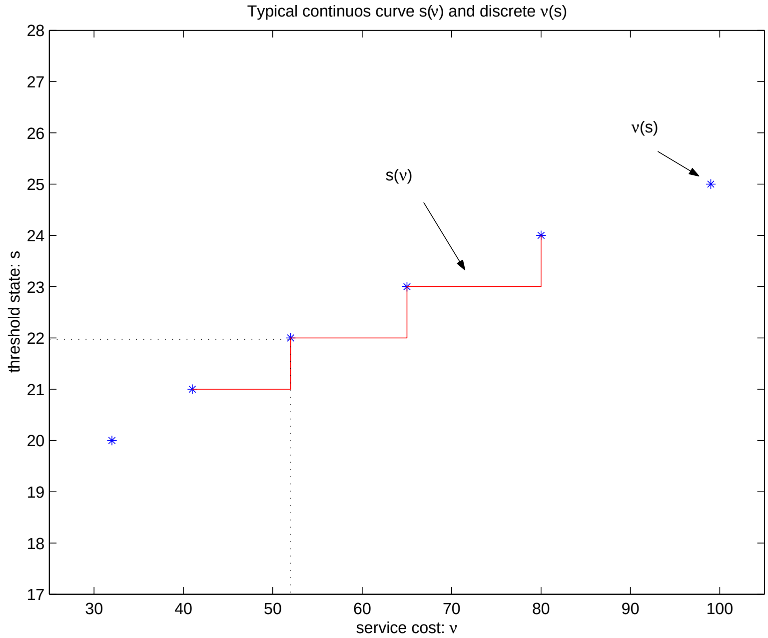

Figure 5: Typical $s(\nu)$ and $\nu(s)$ curves.

Proof: Appendix C.
Instead of finding the threshold state for every value of ν, we can assign to every state s(figure 5), a corresponding service cost value $\nu(s)$ that is the minimum service cost needed to keep state s in the idling set. Therefore, if the system is in state s, it is optimal to serve the queue if the service cost is smaller than $\nu(s)$, leave the queue idle if it is larger and equally optimal to serve or to remain idle if it is equal to $\nu(s)$. The function $\nu(s)$ can be considered as the index associated with state s which, when compared to the actual value of the service cost ν^*, determines the optimal action. This is the characteristics of an *index policy* in the dynamic optimization context as will be discussed below.

5 Stochastic scheduling and Bandit problems, review

In a typical stochastic scheduling problem there is a system that is composed of a number of controllable stochastic processes and a limited amount of available control should be distributed between the projects during the operation of the system in such way to maximize the total reward generated by them. Manufacturing and computer communication systems might be the most important examples of such systems. There is no unified and practical method to find the optimal solution to all the problems that fit into the above general definition. However, many such problems can be formulated in the framework of dynamic programming. Although the straightforward numerical application of this method does not necessarily result into useful results (due to the usually large size of the formulations), its framework sometimes helps to reveal structural properties of the optimal policy. One of the well known models for the dynamic programming formulation of the stochastic scheduling problems with known structural results for the solution is the *Multiarmed Bandit* model. In the basic discrete-time version of the Multiarmed Bandit problem, there are N independent reward processes (called *projects*) and a single server. At each discrete decision instant $t = 1, 2, \ldots$, the server can be allocated to *activate* only one of the projects and the other projects remain *idle*. Each project i; $i = 1, 2, \ldots, N$, when activated, changes its state $s_i(t)$ according to its stationary state transition probability matrix. Also the activated project generates an immediate reward $R(t) = r_i(s_i(t))$ which is a function of its state. The idle projects neither change their states nor produce any rewards. The optimization problem is to maximize the expected discounted value of the total reward defined as

$$E\left[\sum_{t=1}^{\infty} \beta^t R(t)\right] \tag{23}$$

where $0 < \beta < 1$ is a constant discount factor and the initial state is known. This problem has received considerable attention since it was formulated about 60 years ago. The most important result appeared in 1970s where Gittins and Jones [16, 15] found that the optimal policy is of the *index* type. More specifically, they showed that at each decision instant t, there is a function called the *index* associated with each of the projects defined as

$$\nu_i(s_i(t)) = \max_{\tau \geq t} \frac{E\left[\sum_{l=t}^{\tau} \beta^l r_i(s_i(l))\right]}{E\left[\sum_{l=t}^{\tau} \beta^l\right]} \tag{24}$$

and the optimal policy at time t is independent of the previous decisions and is to activate the project with the largest index value. The significance of this results is in exploring the indexing structure of the optimal policy which converts the original N dimensional problem into N one-dimensional problems, a property that is crucial to the applicability of this method for practical applications with large values of N. All of our effort throughout this work is also focused on finding policies of the index type even if they do not result in *the optimal* solution. The interpretation of the above index function is simple. It is the maximum expected discounted reward per unit of the discounted time for each project and intuitively it makes sense to activate the project which can potentially produce the maximum reward. In a number of other significant works on this problem, other interpretations of the index function [14, 35, 18] as well as extensions to the original problem [34, 32, 20, 19] were introduced and studied by

other researchers.
The main restriction of the Multiarmed Bandit problem is the one that requires the passive projects to remain frozen and do not change their states which is not necessarily the case for many problems and particularly our problem. If we consider the N queues in our problems as the N projects in the above formulation, the state of the projects will be the length of each queue and the reward function for serving a queue will be the number of serviced customers. Obviously, the idle queues keep receiving new arrivals and their state keeps changing even during the idle state. This restriction is somehow alleviated in the Multiarmed Bandit formulation of the scheduling problem in regular single-service queueing systems [34, 32] which resulted in the so-called $c\mu$ rule as the optimal policy (also through other approaches e.g. [2, 6]). However, we were not able to use any of those formulations for our bulk-service scheduling problem. We therefore use what Whittle [33] introduced as an extension to this problem which is called the *Restless Bandit* problem and allows the passive projects to produce rewards and change their states too. Unfortunately, with this generalization, the existence of an index-type solution is no longer guaranteed. However, as Whittle showed, in some cases an index-type solution can be found for a relaxed version of this problem that results into reasonable conclusions about the optimal policy for the original problem.

6 Restless Bandits formulation

In this section we explain the Whittle's method for use in the discrete time version of the dynamic optimization problem and will give the formulation of the β-discounted version of the Restless Bandit problem in a way to match our problem and refer the reader to [33, 24, 3, 25] for more detailed information.
In this formulation, the dynamic optimization problem is treated as a linear optimization problem using the linear programming formulation of the MDPs. Let us call the state space of queue i by S_i and the total N dimensional state space of the problem by S. Also, let us show the decision space of the problem with D and suppose that $\alpha(j)$ is the probability distribution of the initial state of the system.
The linear programming(LP) formulation of the MDP [26] converts the original dynamic programming problem

$$V(s) = \max_{d \in D} \left[r(s,d) + \beta \sum_{j \in S} p^d(j|s) V(j) \right] \quad \forall s \in S \tag{25}$$

to the (dual) LP problem

$$Maximize \ \sum_{s \in S} \sum_{d \in D} r(s,d) z(s,d)$$

subject to

$$\sum_{d\in D} z(j,d) - \sum_{s\in S}\sum_{d\in D} \beta p^d(j|s) z(s,d) = \alpha(j) \ \ \forall j \in S$$

and $z(s,d) \geq 0$ for $d \in D$ and $s \in S$.
Here, $\alpha(.)$ is the initial probability distribution of the states and

$$z(s,d) = E\left[\sum_{t=0}^{\infty} \beta^t I[x(t) = s \ \&\ d(t) = d]\right] \tag{26}$$

where $I(.)$ is the indicator function of the event defined by its argument. In other words, $z(s,d)$ is the discounted expected value of the number of times that action d is taken at state s.
For our scheduling problem, the state space S is the product of the N state spaces $S_1, S_2, \ldots, S_N$. Therefore, the objective function of the dual problem can be written as

$$Maximize \ \sum_{n=1}^{N}\left[\sum_{s\in S_n} r_n(s,0) z_n(s,0) + \sum_{s\in S_n} r_n(s,1) z_n(s,1)\right] \tag{27}$$

subject to

$$\sum_{l\in\{0,1\}} z_n(j,l) - \sum_{s\in S_n}\sum_{l\in\{0,1\}} \beta p_n^l(j|s) z_n(s,l) = \alpha_n(j) \ \ for\ n = 1,\ldots,N\ and\ j \in S_n. \tag{28}$$

where

$$z_n(s,1) = E\left[\sum_{t=0}^{\infty} \beta^t I[x_n(t) = s \ \&\ n \in d(t)]\right] \tag{29}$$

and

$$z_n(s,0) = E\left[\sum_{t=0}^{\infty} \beta^t I[x_n(t) = s \ \&\ n \notin d(t)]\right] \tag{30}$$

and $p_n^1(j|s)$ $\left(p_n^0(j|s)\right)$ is the probability of queue n going from state s to state j when it is activated(idle). Obviously, in our problem we have $r_n(s,0) = 0$ and $r_n(s,1) = c_n s$. An additional constraint implicit to this scheduling problem is that at any time t, exactly K queues should be serviced. This constraint is in fact the only constraint that ruins the decoupled structure of the dual problem and the following relaxation removes this limitation. This relaxation assumes that instead of having exactly K of the 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

$$\sum_{n=1}^{N}\sum_{s\in S_n} z_n(s,1) = K/(1-\beta). \tag{31}$$

To exploit the structure of the solution to the new problem, Whittle used the Lagrangian Relaxation method to define a relaxed problem which, in our case, is

$$Maximize \sum_{n=1}^{N}\left[\sum_{s\in S_n} r_n(s,1)z_n(s,1)\right] + \nu\left(K/(1-\beta) - \sum_{n=1}^{N}\sum_{s\in S_n} z_n(s,1)\right) \tag{32}$$

subject to

$$\sum_{l\in\{0,1\}} z_n(j,l) - \sum_{s\in S_n}\sum_{l\in\{0,1\}} \beta p_n^l(j|s)z_n(s,l) = \alpha(j) \quad for\ n=1,\ldots,N\ and\ j\in S_n.$$

the above problem can be stated as

$$Maximize \sum_{n=1}^{N}\left[\sum_{s\in S_n}(r_n(s,1)-\nu)z_n(s,1)\right] + K\nu/(1-\beta) \tag{33}$$

subject to

$$\sum_{l\in\{0,1\}} z_n(j,l) - \sum_{s\in S_n}\sum_{l\in\{0,1\}} \beta p_n^l(j|s)z_n(s,l) = \alpha(j) \quad for\ n=1,\ldots,N\ and\ j\in S_n.$$

Therefore, multiplier ν works as a constant cost for activating a project. Whittle termed ν as a constant subsidy for not activating a project, but in the queuing theory problems the service cost interpretation seems more familiar. Problem (33) can be decoupled into N separate problems

$$Maximize \sum_{s\in S_n}(r_n(s,1)-\nu)z_n(s,1) \tag{34}$$

subject to

$$\sum_{l\in\{0,1\}} z_n(j,l) - \sum_{s\in S_n}\sum_{l\in\{0,1\}} \beta p_n^l(j|s)z_n(s,l) = \alpha(j) \quad for\ n=1,\ldots,N\ and\ j\in S_n. \tag{35}$$

The solution to the Lagrangian Relaxation problem (33) is a function of the parameter ν and is an upper bound to the solution of problem (27) and for a specific value ν^* the solutions to both problems are equal. Suppose that ν^* is known, then, the problem becomes finding the optimal policies for each of the N problems in (34). Here for each queue n we have the problem of serving or not serving the queue at each state $s\in S_n$, given that the reward for serving a queue is $c_n s-\nu^*$ and the reward for not serving it is zero, so that the total discounted expected reward is maximized. This is the problem we studied in section 4 and found that the optimal policy, for a fixed value of the service cost ν, is an index policy with the index being a function of the current state of the system and it is optimal to serve the queue if the index is larger than ν. the optimal policy for each queue is therefore to calculate the value of index for that queue and activate the queue if it is larger than the service cost.

Whittle used this idea and gave a logical heuristic to address the original problem with the strict constraint on the number of active projects. The heuristic policy is to find the critical cost value(index) $\nu_n(s_n(t))$ for each queue n at decision time t and serve the queues with K largest index values. He conjuctured that this policy is asymptotically optimal and approaches the real optimal point as K and N increase. Weber and Weiss [27] showed that this conjecture is not necessarily true in all cases and presented a sufficient condition for it to hold. They also presented a counterexample for this conjecture. However, based on their results, they argued that such counterexamples are extremely rare and the deviation from optimality is negligible. We remind that the above heuristic would not have been meaningful if our projects did not have the monotonicity property that resulted in an index type optimal solution for the single queue problem.

The significance of this result is in the fact that it reduces the original problem to the simpler problem of finding the optimal policy for a single-queue system, which is potentially much easier to solve and to get either an analytical or experimental solution for it. So far, we have shown that our problem have certain properties that make the above heuristic an acceptable indexing policy. The complexity of this indexing policy is hidden in the form of the $\nu(s)$ function for each queue and in the following section we present a recursive method to calculate $\nu(s)$ for each queue.

7 Calculation of the index function

The index ν associated with state $s \in S_n$ is the amount of the service cost that makes both the active and idle actions equally favorable at that state under the optimal policy. Using the results of appendix B, it can be easily shown that for that value of ν, the optimal policy would be to serve the queue for states larger than s and to remain idle for states smaller than s. Therefore, the following set of equations characterizes the value function $V^s(.)$ for $\nu(s)$.

$$V^s(0) = \beta \sum_{i=0}^{\infty} p(i) V^s(0+i) \quad (36)$$

$$V^s(1) = \beta \sum_{i=0}^{\infty} p(i) V^s(1+i) \quad (37)$$

$$\vdots$$

$$V^s(s) = \beta \sum_{i=0}^{\infty} p(i) V^s(s+i) \quad (38)$$

$$
\begin{aligned}
V^s(s) &= -\nu(s) + cs + \beta \sum_{i=0}^{\infty} p(i) V^s(i) &(39)\\
V^s(s+1) &= -\nu(s) + c(s+1) + \beta \sum_{i=0}^{\infty} p(i) V^s(i) &(40)\\
&\vdots
\end{aligned}
$$

This system of equations has all the $V^s(.)$ values plus $\nu = \nu(s)$ as unknowns. In other words we fix the border state to s and need to find the corresponding $\nu(s)$. To find the complete $\nu(s)$ index function, this set of equations should be solved for every s. In the following, we will try to exploit the properties of this system of equations to find an easier method for calculating the $\nu(s)$ function. Due to the special form of the $V^s(.)$ function we have

$$V^s(s+i) = V^s(s) + ci; \quad i = 0, 1, \ldots$$

Therefore the set of unknowns reduces to $V^s(0), \ldots, V^s(s), s, \nu(s)$. The last term in equation (39) is equal to $V^s(0)$ therefore we have

$$\nu(s) = cs + V^s(0) - V^s(s). \tag{41}$$

Equation (38) can be written as

$$
\begin{aligned}
V^s(s) &= \beta \sum_{i=0}^{\infty} p(i) V^s(s+i)\\
&= \beta \sum_{i=0}^{\infty} p(i)(V^s(s) + ci)\\
&= \beta V^s(s) + \beta c \lambda.
\end{aligned}
$$

Therefore we have

$$V^s(s) = \frac{\beta c \lambda}{1-\beta} \tag{42}$$

and

$$V^s(s+i) = ci + \frac{\beta c \lambda}{1-\beta}. \tag{43}$$

Substituting (42) in (41) gives

$$\nu(s) = cs + V^s(0) - \frac{\beta c \lambda}{1-\beta}. \tag{44}$$

According to this equation we only need to find $V^s(0)$ in order to calculate $\nu(s)$. The reduced set of equations for finding $V^s(x);\ x = 0, \ldots, s-1$ is therefore

$$
\begin{aligned}
V^s(x) &= \beta \sum_{i=0}^{\infty} p(i) V^s(x+i)\\
&= \beta \sum_{i=0}^{s-x-1} p(i) V^s(x+i) + \beta \sum_{i=s-x}^{\infty} p(i) V^s(x+i)
\end{aligned}
$$

$$
\begin{aligned}
&= \beta \sum_{i=0}^{s-x-1} p(i)V^s(x+i) + \beta \sum_{i=0}^{\infty} p(s-x+i)V^s(s+i) \\
&= \beta \sum_{i=0}^{s-x-1} p(i)V^s(x+i) + \beta V^s(s) \sum_{i=0}^{\infty} p(s-x+i) \\
&+ \beta c \sum_{i=0}^{\infty} ip(s-x+i) \\
&= \beta \sum_{i=0}^{s-x-1} p(i)V^s(x+i) \\
&+ \beta V^s(s)h(s-x-1) + \beta(\lambda - m(s-x-1) - (s-x)h(s-x-1) \\
&= \beta \sum_{i=0}^{s-x-1} p(i)V^s(x+i) \\
&+ \beta h(s-x-1)(V^s(s) - c(s-x)) + \beta c(\lambda - m(s-x-1))
\end{aligned}
$$

where $h(.)$ and $m(.)$ are functions of the Poisson distribution defined as

$$
h(n) = \sum_{i=n+1}^{\infty} p(i) \tag{45}
$$

and

$$
m(n) = \sum_{i=0}^{n} ip(i). \tag{46}
$$

Defining $\mathrm{W}_s = (V^s(0), \ldots, V^s(s-1))$, we can write the above system as

$$
A_s \mathrm{W}_s = B_s \tag{47}
$$

where

$$
A_s = \begin{bmatrix}
1-\beta p(0) & -\beta p(1) & \ldots & -\beta p(x) & \ldots & -\beta p(s-1) \\
0 & 1-\beta p(0) & \ldots & -\beta p(x-1) & \ldots & -\beta p(s-2) \\
\vdots & \vdots & & \vdots & & \vdots \\
0 & 0 & \ldots & 1-\beta p(0) & \ldots & -\beta p(s-x-1) \\
\vdots & \vdots & & \vdots & & \vdots \\
0 & 0 & \ldots & 0 & \ldots & 1-\beta p(0)
\end{bmatrix} \tag{48}
$$

and

$$
B_s = \beta \begin{bmatrix}
h(s-1)\left[\frac{\beta c\lambda}{1-\beta} - cs\right] + c\lambda - cm(s-1) \\
h(s-2)\left[\frac{\beta c\lambda}{1-\beta} - c(s-1)\right] + c\lambda - cm(s-2) \\
\vdots \\
h(s-x-1)\left[\frac{\beta c\lambda}{1-\beta} - c(s-x)\right] + c\lambda - cm(s-x-1) \\
\vdots \\
h(0)\left[\frac{\beta c\lambda}{1-\beta} - c\right] + c\lambda - cm(0).
\end{bmatrix} \tag{49}
$$

An immediate observation of the role of the weight coefficient c in the above equations shows that it only results in a solution W_s(and so $V^s(0)$) which is c times larger than the solution for $c = 1$ case. Taking this observation into account, from equation (44) it can be seen that the index function satisfies

Property 3 *If $\nu_c(s)$ is the index function for a bulk service queue with the reward function at state x defines as $R(x) = cx$, we have $\nu_c(s) = c\nu_1(s)$; $\forall s \in S$.*

Hence, without any loss of generality we will continue our analysis for $c = 1$.
By solving equation (47) the value of $\nu(s)$ is found and for every value of s a similar $s \times s$ system needs to be solved. However, a closer look at the structure of A_s and B_s matrices shows that A_{s+1} is formed by adding an additional first row and first column to A_s and also B_{s+1} is formed by adding an additional first row to B_s. The new system has of course $s+1$ unknowns shown as $\mathrm{W}_{s+1} = (V^{s+1}(0), V^{s+1}(1), \ldots, V^{s+1}(s))$. Since matrix A is upper triangular, the subsystem defining the $V^{s+1}(1), \ldots, V^{s+1}(s)$ values is the same as the previous system defining the $V^s(0), \ldots, V^s(s-1)$ values. Therefore, we have

Property 4 *If $V^s(.)$ is the value function of the optimal policy for the case where s is the border state and $V^{s+1}(.)$ the similar function for $s+1$ being the border state, then $V^{s+1}(x+1) = V^s(x);\ \ x = 0, \ldots, s-1$.*

Also using the above property it is easy to show that

Property 5 *For $V(0)$ values we have*
$V^{s+1}(0) = \frac{\beta}{1-\beta p(0)} \Big[p(1)V^s(0) + \ldots + p(s)V^s(s-1) + \lambda + h(s)(\frac{\beta\lambda}{1-\beta} - s - 1) - m(s)\Big].$

Therefore, once the values of the $V^s(x)$; $x = 0, \ldots, s-1$ are found, the values of $V^{s+1}(x)$; $x = 0, \ldots, s$ can be easily calculated using the $V^s(.)$ values. The index function can therefore be efficiently computed using this recursive method.

We can also use the above relations to prove a number of properties of the index function. The results are for $c = 1$ and the extension to $c \neq 1$ is trivial.

Theorem 3 *The index function $\nu(s)$ is a non-decreasing function of s such that*
(a) $\nu(s) \leq \nu(s+1)$.
(b) $\nu(s+1) \leq \nu(s) + 1$

Proof: Based on equation (44) we have

$$\nu(s+1) - \nu(s) = 1 + V^{s+1}(0) - V^s(0) \tag{50}$$

but

$$\begin{aligned}
V^{s+1}(0) - V^s(0) &= \beta \sum_{i=0}^{\infty} p(i)[V^{s+1}(i) - V^s(i)] \\
&= \beta p(0)[V^{s+1}(0) - V^s(0)] + \beta \sum_{i=1}^{\infty} p(i)[V^{s+1}(i) - V^s(i)] \\
&= \beta p(0)[V^{s+1}(0) - V^s(0)] + \beta \sum_{i=1}^{\infty} p(i)[V^s(i-1) - V^s(i)]
\end{aligned}$$

Therefore

$$V^{s+1}(0) - V^s(0) = \frac{\beta}{1-\beta p(0)} \sum_{i=1}^{\infty} p(i)[V^s(i-1) - V^s(i)]. \tag{51}$$

Since $V^s(i-1) \le V^s(i) \le V^s(i-1)+1;\ \ i = 1, 2, \ldots$(Lemma 1), we have

$$\frac{-\beta(1-p(0))}{1-\beta p(0)} \le V^{s+1}(0) - V^s(0) \le 0. \tag{52}$$

Using equation (50), we have

$$1 - \frac{\beta(1-p(0))}{1-\beta p(0)} \le \nu(s+1) - \nu(s) \le 1 \tag{53}$$

and since $\beta < 1$ the left hand term is always greater than 0 which completes the proof. This property was used in section 6 to establish the indexing argument and tells that the $\nu(s)$ curve is monotonic increasing with a maximum slope of 1(c in the general case).

8 Index function in light traffic regime

In the previous section we calculated the index function via a recursive method for a Poisson arrival with arbitrary rate but we failed to present a closed form formula for that function due to the complexity of the equations. An interesting case to consider is when the arrival rate is low so that we can model the arrivals in every period to be according to an iid Bernoulli sequence with p the probability of having one arrival and $1-p$ the probability of zero arrivals. It is worth noticing that this assumption is not as restrictive as its name may imply. It only needs the arrival rate to be enough low with respect to our time unit which is the distance between successive decision instances. Therefore for a system with small page sizes or equivalently large download bandwidth, this can be a reasonable assumption. Consider again the bulk service queuing system with infinite capacity for the server and assume that we have the option of serving or not serving the queue at equally spaced decision instances of time $t = 0, 1, \ldots$ where the service time of the server is a constant 1. Using the same method as the last section, if

ν is the amount of service cost that makes state s equally favorable for both idle and active decisions, then the value function of the optimal policy satisfies the following system of linear equations

$$\begin{aligned} V(0) &= \beta(1-p)V(0) + \beta p V(1) & (54)\\ V(1) &= \beta(1-p)V(1) + \beta p V(2) & (55)\\ &\vdots \\ V(s) &= \beta(1-p)V(s) + \beta p V(s+i) & (56)\\ V(s) &= -\nu + cs + \beta(1-p)V(0) + \beta p V(1) & (57)\\ V(s+1) &= -\nu + c(s+1) + \beta(1-p)V(0) + \beta p V(1) & (58)\\ &\vdots \end{aligned}$$

where $V(x)$ is the expected reward of the optimal policy given the initial state x. Here again we have

$$V(s+i) = V(s) + ci; \quad i = 0, 1, \ldots$$

and we can verify that the following equations hold

$$\begin{aligned} \nu(s) &= cs + V(0) - V(s) \\ V(0) &= \left(\frac{\beta p}{1-\beta(1-p)}\right)^s V(s) \\ V(s) &= \frac{\beta c p}{1-\beta}. \end{aligned}$$

Therefore

$$\nu(s) = cs + \frac{\beta c p}{1-\beta}\left[\left(\frac{\beta p}{1-\beta(1-p)}\right)^s - 1\right]. \tag{59}$$

It can be shown that this function is monotonic increasing with a slope between 0 and 1. Since p is the probability of a single arrival for a Poisson process with a low rate, it is in fact the rate of the process and can be replaced with λ keeping in mind that the formula is only valid for small values of $\lambda < 1$. We expect the new index function to be very close to the original index function for small rates and deviate from that as the rate increases. To observe the degree of match between the two functions, we plotted the functions for several rate values in figure 6. According to these results, as we expect, the two functions are very close for small values of λ and their difference increases with λ. However, there is an acceptable match even for a range of $\lambda > 1$ values. Therefore, for practical purposes, the closed form function might be used for small rates.

9 Results

In this section some of the results we obtained from simulation studies about the performance of different broadcast scheduling policies are presented. We have compared

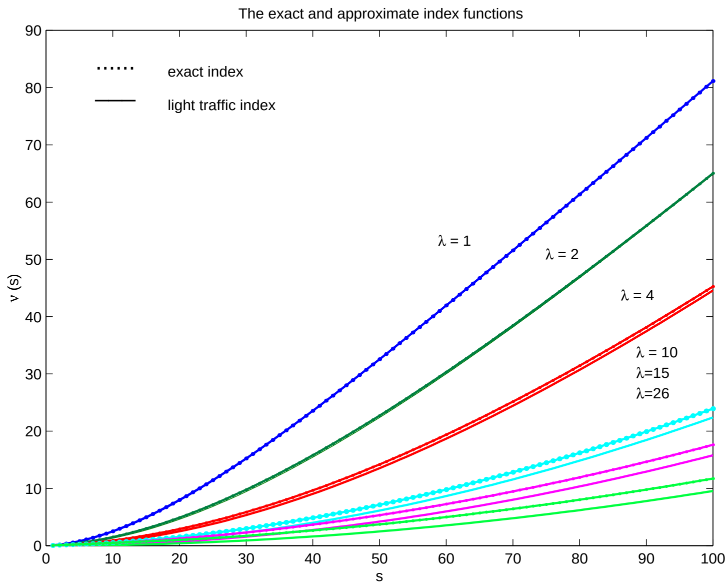

Figure 6: The exact index function and the light traffic approximation.

the performances of the following policies in different experiments.

- MRF or Maximum Requested First, This policy serves the queue with the largest number of pending requests.
- FCFS , this policy is the simple First Come, First Serve policy where the queue with the oldest request is served first.
- PIP or Priority Index Policy, this policy introduced in [29] is the best known indexing broadcast scheduling policy. The index function is defined as x/λ^γ where x is the queue length, λ the arrival rate and γ is a constant. It is found by trial and error that a value of γ around 0.5 is the optimum value. Therefore, in the following simulations we have used $x/\sqrt{\lambda}$ as the PIP index function.
- NOP or Near-Optimal Policy which is the index function defined by our method.

In the first set of simulations, we used 100 queues with the arrival rates distributed according to the Zipf distribution. The total rate was varied from very low to very high values to show the performances of the policies for a wide range of the input rate. The service times were set to one time unit and the total average waiting times were calculated for each simulation. Figure 7 shows the results of these experiments. As we observe, the performance of our policy is much better than MRF and FCFS and identical to PIP. Also in figure 8 the performances of PIP and NOP policies are compared with the light traffic approximation of the NOP index which we call NOPL. It

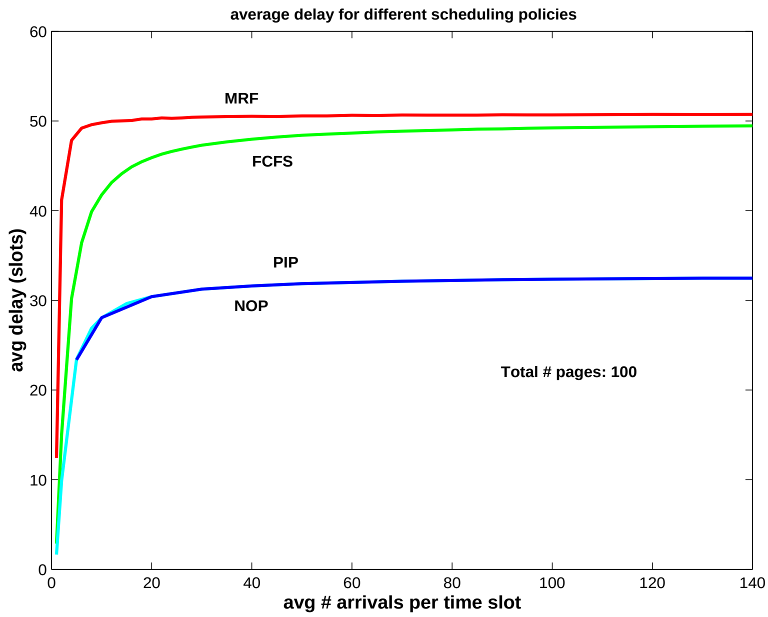

Figure 7: Comparison of the total average waiting time for different scheduling policies with the distribution of the arrival rates having a Zipf distribution.

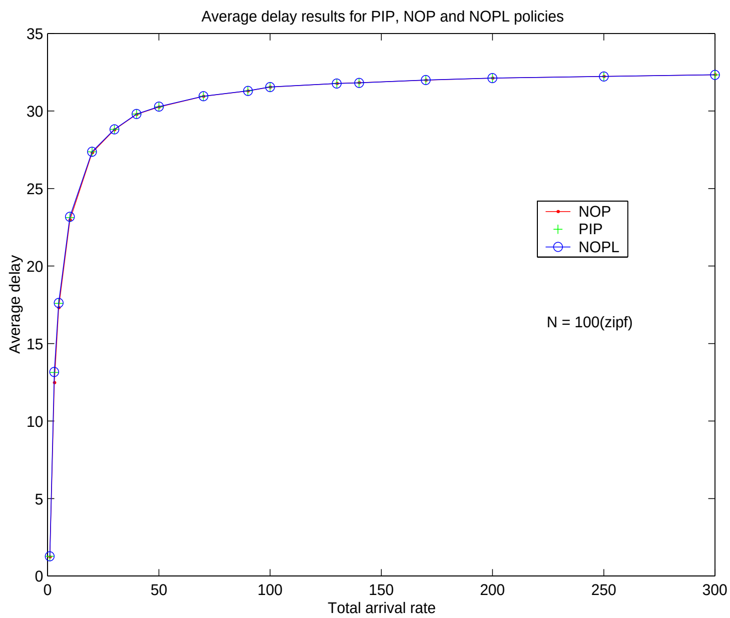

Figure 8: Performance comparison of the PIP, NOP and NOPL policies.

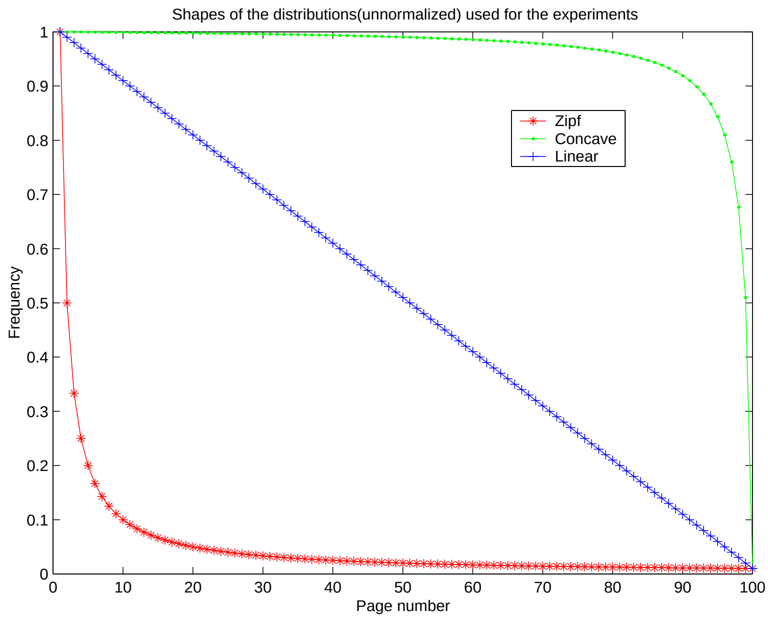

Figure 9: The Zipf distribution and the other two distributions used as the probability of the different pages in the experiments.

can be seen that, for the range of arrival rates tested in these experiments, there is no difference in the performances of NOP and NOPL policies and the closed form index can be used in practical purposes. In order to further compare the performances of PIP and NOP policies, we performed two other sets of simulations each with different distribution of the input rates among the queues. Since the Zipf distribution defines a convex distribution, we used a linearly decreasing distribution in one group of experiments and a concave shaped distribution in another group and ran the simulations for different values of the total input rate(figure 9). Figures 10 and 11 show the results of these experiments only for the PIP and NOP policies. We can see that the results are extremely similar. It was mentioned before that once we find a proper index function, any monotonic increasing function of that index can be used as an index as well. The results suggest that what Su [29] found as an index by trial an error is in fact very close to a monotonic increasing function of the index we have calculated using optimization arguments. Figure 12 shows the individual average waiting times experienced by the requests for each page under PIP and NOP policies for an specific arrival rate. The close matching of the two results confirms the close relation between PIP and NOP policies. In another set of experiments we compared the performances of PIP and NOP policies for the case where the pages have different weights. We showed in previous sections that the effect of weight C in the index function $\nu(s)$ is in the form of a simple multiplicative factor. PIP, in its original form, does not address the case with weights.

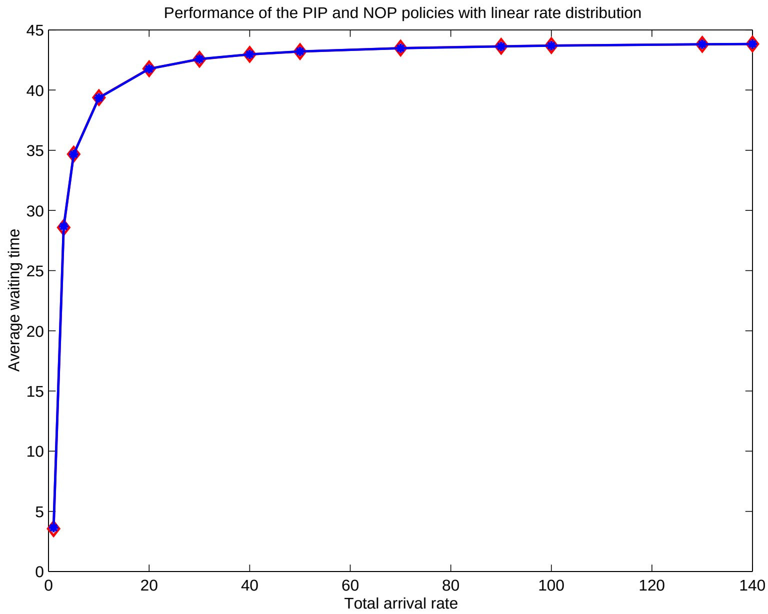

Figure 10: Comparison of the total average waiting time for PIP and NOP scheduling policies with the distribution of the arrival rates having a linear shape.

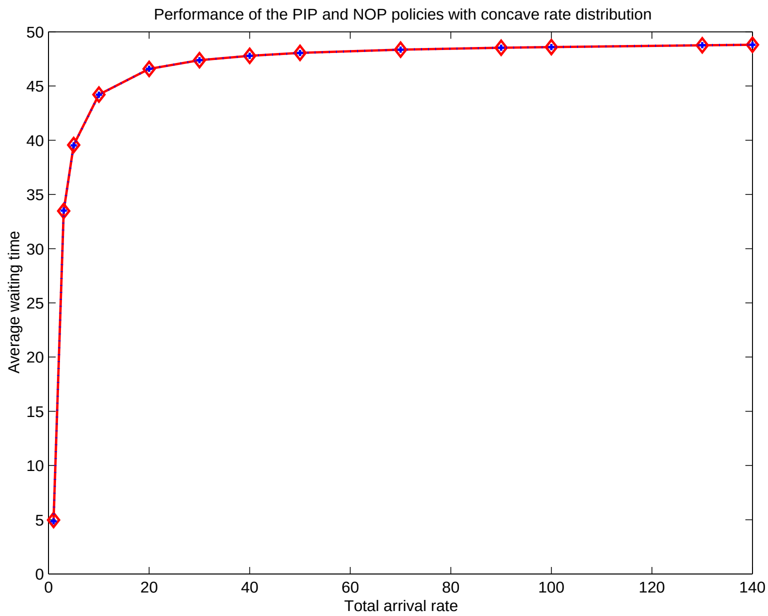

Figure 11: Comparison of the total average waiting time for PIP and NOP scheduling policies with the distribution of the arrival rates having a concave shape.

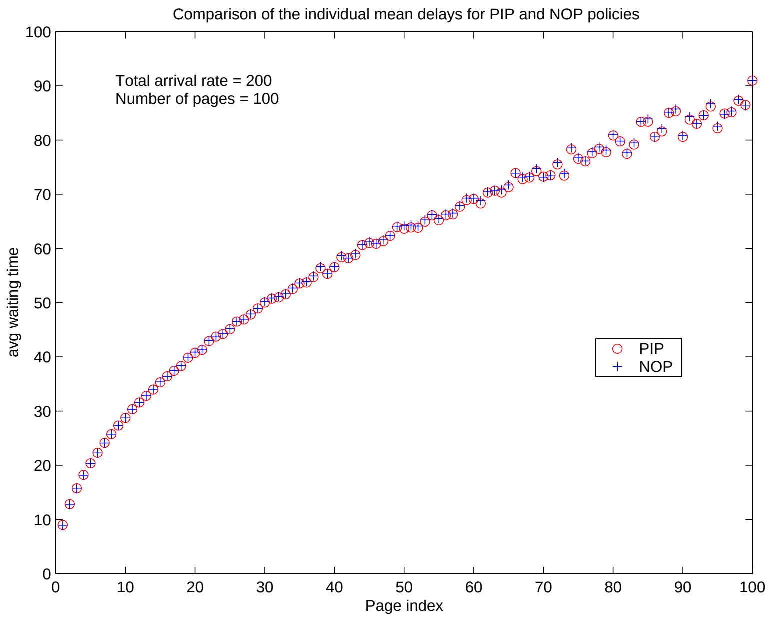

Figure 12: Average waiting times for the requests for each of the 400 pages under different policies.

Therefore, we tried to use the same analogy and extend its definition so that the weight coefficient appears in the index function as well. In the first extension, which we call EPIP1 for notational convenience, we define the index function as $\nu(x) = \frac{cx}{\sqrt{\lambda}}$ and in the second extension(EPIP2) we define it as $\nu(x) = \frac{\sqrt{c}x}{\sqrt{\lambda}}$. We performed the experiments on a system with 100 pages with Zipf distribution of the arrival rates and assigned a weight of 5 to the first 10 pages. The weights of the other pages were set to 1. Figure 13 shows the performances of all four policies under different arrival rates. As we can see, PIP by itself does not perform very well which is not unexpected. EPIP1, which uses the same multiplicative form as NOP to incorporate the effect of weights, also does not perform as good as NOP. However, EPIP2 have exactly the same performance as NOP and suggests that the effect of weight in the PIP index should be through a square root multiplicative factor. The NOP policy and the method we used for its derivation, in addition to having the notion of optimality, has the advantage of being more flexible because this method allows us to define the index function for the general case of weighted priorities assigned to the packages and moreover, we are currently using it for dealing with the unequal file size case which is not studied yet.

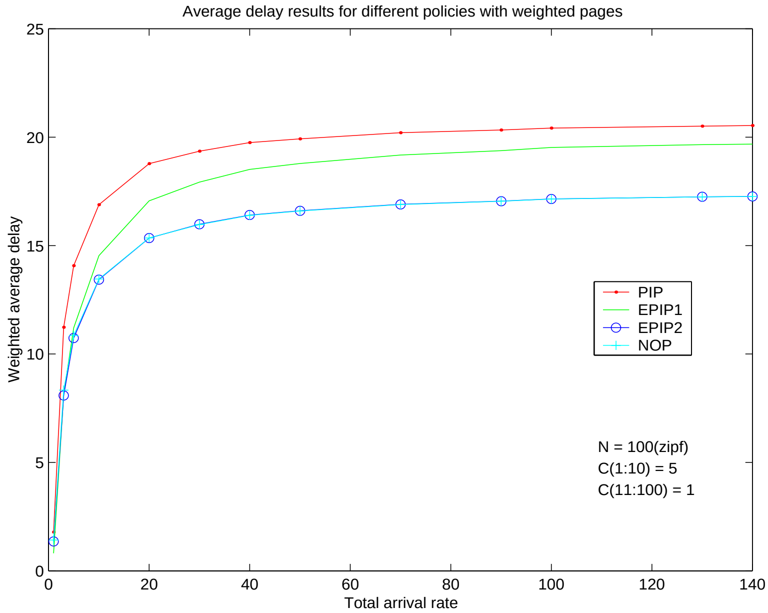

Figure 13: Performance comparison of NOP and different versions of the PIP policy for the weighted average delay case.

10 Conclusion

In this report, the problem of optimal broadcast scheduling was addressed. We presented a MDP formulation of this problem for the case where all files have equal sizes. This formulation allowed us to use the restless bandit problem approach after proving the necessary properties of the single-queue problem. We also derived a recursive method for the calculation of the index function for our near-optimal policy and a closed form formula for the light traffic case. The experimental results show that our policy outperforms or matches the results of all other heuristic policies in all the experiments. Moreover, our approach naturally allows for the assignment of distinct weights to different pages as a form of soft priority assignment in the system. The above method has also been used to address the extended version of this problem where the restriction of equal file sizes is removed and the results will be published in future reports.

A Derivation of the maximization problem

Assuming initial condition $X(0)$, the objective function of the minimization problem can be written as

$$J_\beta = E\left[\sum_{t=0}^{\infty} \beta^t \left[\sum_{i=1}^{N} c_i X_i(t)\right]\right]$$

$$= E\left[\sum_{i=1}^{N} c_i X_i(0)\right] + E\left[\sum_{t=0}^{\infty} \beta\beta^t \left[\sum_{i=1}^{N} c_i X_i(t+1)\right]\right]$$

but if $d(t)$ is the set of pages transmitted at time t we have

$$X_i(t+1) = \begin{cases} X_i(t) + A_i(t) & i \notin d(t) \\ A_i(t) & i \in d(t) \end{cases} \tag{60}$$

with $A_i(t)$ being the number of new requests for page i. therefore J_β can be written as

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

We also have

$$\beta J_\beta = \beta E\left[\sum_{t=0}^{\infty} \beta^t [\sum_{i=1}^{N} c_i X_i(t)]\right] \tag{61}$$

therefore

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

The first two terms of the right hand side of the equation are independent of the policy. Therefore, since $1 - \beta > 0$, minimizing J_β is equal to maximizing

$$\hat{J}_\beta = E[\sum_{t=0}^{\infty} \beta^t \sum_{i \in d(t)} c_i X_i(t)] \tag{62}$$

which completes the derivation.

B optimality of the threshold policy

In this part we prove that the optimal policy is of the threshold type and moreover, the idling region is a convex set on the state space of the queue containing the origin. We show the value function of the optimal policy π^* by $V(.)$ and we first prove some properties of this function.

We need the following lemma:

Lemma 2 *Let $S_p^d(x)$ denote the resulting discounted reward sum when the initial condition is x and arrivals occur as sample path p and the fixed(independent of state) decision sequence d is applied to the system. Then we have*

$$S_p^d(x) \leq S_p^d(x+1) \leq c + S_p^d(x). \tag{63}$$

Consider two identical queues one with initial condition x and the other with initial condition $x+1$ defined as above. If the same fixed policy is applied to these two systems, the reward would be the same before the first service epoch. At that point, the second system receives a reward that is c units more than that received by the first system. Since the dynamics of the system forces the length of the serviced queues to zero, it in fact erases the memory of the queues after each service. Therefore, the resulting rewards even for both queues would be the same afterwards. Therefore, the left hand inequality holds($c > 0$). The presence of the discount factor $0 < \beta < 1$ causes the additional instantaneous reward in the second queue to result in at most a c unit difference between the two discounted sum of the rewards(if queues are served at time $t = 0$), hence the right inequality holds.

The first part of the theorem can be proved using the above lemma.

Theorem 4 *For the value function $V(.)$ of the optimal policy of our maximization problem, we have*
(a) $V(x+1) \leq V(x) + c$.
(b) $V(x) \leq V(x+1)$

Proof: Let $d\pi^*$ be the optimal policy and denote by π_p^x the deterministic sequence of decisions dictated by π^* when the arrivals occur according to a deterministic sample path p and the initial condition is x. According to lemma 2 we have

$$S_p^{\pi_p^{x+1}}(x+1) \leq S_p^{\pi_p^{x+1}}(x) + c \tag{64}$$

If we take the expectation of both sides with respect to the sample path probability $P(p)$, we get

$$V(x+1) \leq c + \sum_p P(p) S_p^{\pi_p^{x+1}}(x). \tag{65}$$

Also, according to the definition of optimality of policy π^* we have

$$V(x) = \sum_p P(p) S_p^{\pi_p^x}(x) \geq \sum_p P(p) S_p^{d_p^Y}(X). \tag{66}$$

inequality (a) follows from combining the two above results.
Also, according to lemma 2 we have

$$S_p^{\pi_p^x}(x) \leq S_p^{\pi_p^x}(x+1) \tag{67}$$

If we take the expectation of both sides with respect to the sample path probability $P(p)$, we get

$$V(x) \leq \sum_p P(p) S_p^{\pi_p^x}(x+1). \tag{68}$$

Also, according to the definition of optimality of policy d^* we have

$$V(x+1) = \sum_p P(p) S_p^{\pi_p^{x+1}}(x+1) \geq \sum_p P(p) S_p^{\pi_p^x}(x+1). \tag{69}$$

Hence inequality (b) follows.

Now, we can prove the following property:

Theorem 5 *If $\pi^*(x) = 0$, i.e. it is optimal to remain idle at state x, then it is also optimal to remain idle at state $x-1$ i.e. $\pi^*(x-1) = 0$.*

Proof: since $\pi^*(x) = 0$, we have:

$$cx - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq \beta \sum_{i=1}^{\infty} p(i) V(x+i) \tag{70}$$

Starting with the above property, we have

$$V(x+i) \leq c + V(x-1+i) \tag{71}$$

or

$$\beta \sum_{i=1}^{\infty} p(i) V(x+i) \leq c + \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{72}$$

Using the hypothesis, we have

$$cx - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq c + \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{73}$$

or

$$cx - c - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{74}$$

that is, $d(x-1) = 0$ which completes the proof.

C Relation between the threshold state and the service cost

We showed that for every value of the service cost ν there exist a threshold state $s(\nu)$ with the set of idling states under the optimal policy being $S_0 = 0, \ldots, s$. Here we will show that $s(\nu)$ is a non-decreasing function.
Let us assume that u is the stationary optimal policy for service cost ν with threshold state s and denote by $V^{\nu}(.)$ the value function associated with that policy. Based on the optimality principle, function $V^{\nu}(.)$ satisfies:

$$\begin{aligned} V^{\nu}(0) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(0+i) \geq -\nu + 0 + V^{\nu}(0) \\ V^{\nu}(1) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(1+i) \geq -\nu + c + V^{\nu}(0) \\ &\vdots \\ V^{\nu}(x) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \geq -\nu + cx + V^{\nu}(0) \\ &\vdots \\ V^{\nu}(s) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(s+i) \geq -\nu + cs + V^{\nu}(0) \\ V^{\nu}(s+1) &= -\nu + c(s+1) + V^{\nu}(0) \geq \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(s+1+i) \\ &\vdots \end{aligned} \tag{75}$$

Now, take a new value for the service cost $\nu' > \nu$ and show by $V^{\nu'}(.)$ the value function obtained by applying policy u(with threshold s) with this new value of the service cost. Function $V^{\nu'}(.)$ satisfies:

$$\begin{aligned} V^{\nu'}(0) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(0+i) \\ V^{\nu'}(1) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(1+i) \\ &\vdots \\ V^{\nu'}(x) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(x+i) \\ &\vdots \\ V^{\nu'}(s) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(s+i) \\ V^{\nu'}(s+1) &= -\nu + c(s+1) + V^{\nu'}(0) \end{aligned} \tag{76}$$

$$\vdots$$

Let's denote the difference between the two value functions by $\Delta(.)$ i.e. $\Delta(x) = V^{\nu}(x) - V^{\nu'}(x) \quad x = 0, 1, \ldots$. It is easy to show that function $\Delta(.)$ satisfies the following equations:

$$\Delta(x) = \beta \sum_{i=0}^{\infty} p(i)\Delta(x+i) \;\; for \;\; x \leq s \tag{77}$$

and

$$\Delta(x) = \beta \sum_{i=0}^{\infty} p(i)\Delta(i) + \Delta\nu = \Delta(0) + \Delta\nu \;\; for \;\; x > s \tag{78}$$

where $\Delta\nu = \nu' - \nu > 0$. After some simplifications we have

$$\Delta(x) = \beta \sum_{i=0}^{s-x} p(i)\Delta(x+i) + \beta(\Delta(0) + \Delta\nu)h(s+1-x) \;\; for \;\; x \leq s \tag{79}$$

where $h(x) = \sum_{i=x}^{\infty} p(i)$. The following simple lemma asserts that all $\Delta(0), \ldots, \Delta(s)$ values are positive.

Lemma 3 *All $\Delta(i)$ $i = 0, 1, \ldots$ values defined above are positive and are of the form $\Delta(i) = k_i(\Delta(0) + \Delta\nu)$ where $0 < k_i < 1$ for $0 \leq i \leq s$ and $k_i = 1$ for $i > s$.*

Proof: From the above equations $\Delta(s)$ can be written as

$$\Delta(s) = \beta p(0)\Delta(s) + \beta h(1)(\Delta(0) + \Delta\nu)$$

or

$$\Delta(s) = \frac{\beta h(1)}{1 - \beta p(0)}(\Delta(0) + \Delta\nu).$$

We also have

$$\beta h(1) = \beta(1 - p(0)) < 1 - \beta p(0)$$

therefore, $\Delta(s)$ can be written as

$$\Delta(s) = k_s(\Delta(0) + \Delta\nu)$$

where $0 < k_s < 1$. Now we show that all $\Delta(x)$ values for $x < s$ have the same form by using full induction. Suppose that all $\Delta(i)$ values for $i = x+1, \ldots, s$ are of the form

$$\Delta(i) = k_i(\Delta(0) + \Delta\nu) \;\; 0 < k_i < 1.$$

Using equation (79), the value of $\Delta(x)$ can be calculated

$$\begin{aligned}\Delta(x)(1 - \beta p(0)) &= \beta p(1)\Delta(x+1) + \ldots + \beta p(s-x)\Delta(s) + \beta h(s+1+x)(\Delta(0) + \Delta\nu) \\ &= \beta[p(1)k_{x+1} + \ldots + p(s-x)k_{s-x} + h(s+1-x)](\Delta(0) + \Delta\nu)\end{aligned}$$

Since the k_i $i = x+1, \ldots, s-x$ values are all less than one, we have

$$\Delta(x)(1 - \beta p(0)) < \beta(1 - p(0))(\Delta(0) + \Delta\nu)$$

or

$$\Delta(x) = k_x(\Delta(0) + \Delta\nu) \;\; 0 < k_x < 1.$$

Therefore, by induction, all $\Delta(i)$ $i = 0, \ldots, s$ are of the above form. Specifically, for $i = 0$ we have

$$\Delta(0) = k_0(\Delta(0) + \Delta\nu)$$

or

$$\Delta(0) = \frac{k_0 \Delta\nu}{1 - k_0} > 0.$$

Since $\Delta\nu > 0$, we conclude that all $\Delta(i)$ $i = 0, \ldots$ values are positive which completes the proof.

Now, we go back to equation (76) and try to find under what conditions the policy u is also optimal, i.e. satisfies the optimality equation for all states, when the service cost is ν'. For every state $0 \leq x \leq s$ we have from equation (75)

$$V^{\nu}(x) = \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \geq -\nu + cx + V^{\nu}(0) \tag{80}$$

also from the above lemma we have

$$\Delta(x) < \Delta\nu + \Delta(0) \tag{81}$$

therefore, subtracting (81) from (80), we have

$$V^{\nu'}(x) > -\nu' + cx + V^{\nu'}(0) \tag{82}$$

that is, the u policy is optimal for $0 \leq x \leq s$ states. For $x > s$ states from equation (75) we have

$$\begin{aligned} cx - \nu + V^{\nu}(0) &\geq \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \\ &= \beta \sum_{i=0}^{\infty} p(i)[cx + ci - \nu + V^{\nu}(0)] \\ &= \beta[cx - \nu + V^{\nu}(0)] + \beta c \lambda \end{aligned} \tag{83}$$

Obviously, this inequality strengthens as x increases. Also, due to the optimality of state $x = s$ as the largest state of the idling set, no value of $x < s+1$ can satisfy (83). Therefore, $x = s+1$ is the smallest integer(state) which satisfies (83). We know from lemma (3) that for policy u, $V^{\nu}(0)$ is a non-increasing function of ν(since $\Delta(0) > 0$). Hence, inequality (83) weakens as ν increases and the maximum value of ν for which the inequality still holds for $x = s+1$ is the one satisfying

$$c(s+1) - \nu^* + V^{\nu^*}(0) = \beta[c(s+1) - \nu^* + V^{\nu^*}(0)] + \beta c \lambda \tag{84}$$

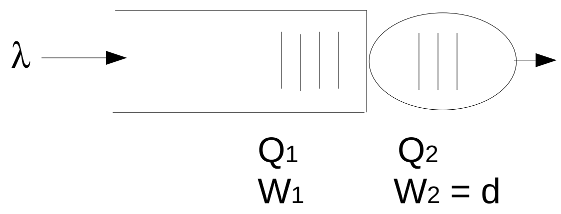

Figure 14: A bulk service queuing system.

or

$$\nu^* = c(s+1) + V^{\nu^*}(0) - \frac{\beta c \lambda}{1-\beta}. \tag{85}$$

Therefore, as long as the value of the service cost is smaller than ν^*, state $x = s+1$(and so the larger states) stay in the active region and the policy u with threshold state s remains optimal.
To summarize the above arguments, we showed that if policy u with threshold state s is optimal for a service cost ν(and produces a value function $V^{\nu}(.)$), then it is also optimal for all values of the service cost ν' where $\nu \leq \nu' \leq \nu^*$. But comparing equations (85) and (44), we find that ν^* is the value of the service cost that makes state $s+1$ the threshold value of the optimal policy. therefore for $\nu' > \nu^*$ values, the same argument can be repeated for the optimal policy u' with its threshold state being $x = s+1$ and so the property is proved.

D Properties of some bulk service queues with continuous service

In our broadcast system the bulk size is infinite and service time is a constant. Let us for example consider a single discrete-time broadcast queue of type $M/D_1^{\infty}/1$ with arrival rate λ and service time d where the service occurs only at discrete time instants of distance d (figure 14). Here we denote by Q_1 and Q_2 the number of customers in the queue and in service respectively and by W_1 and W_2 the corresponding waiting times. The total queue length and waiting time (Q and W) will be the sums of the the two terms. By definition, the value of W_2 is fixed and is equal to d. Also, since the waiting room in the queue is completely emptied at the beginning of every service period, the number of customers who will be waiting for the beginning of the next period (Q_2) will have a Poisson distribution with rate λd. The distribution of the waiting time of the customers in the queue is also easily obtained by considering the fact that by PASTA the residual time (of the current period) seen by the arrivals is Unif$[0, d]$. Therefore, the average waiting time in the queue (W_1) is $\frac{d}{2}$. The average value of the number of waiting customers(Q_1) is easily obtained from W_1 using the Little's law and is $\frac{\lambda d}{2}$. Since we assume that the queue never becomes empty, at the start of each service period Q_1 becomes zero and starts increasing according to a Poisson process with rate λ after that until the start of the next service which occurs d seconds later. Q_1 is therefore a cyclostationary process with period d and is Poisson(λ) inside every time interval of the form $(kd, (k+1)d]$ with $k \in N$. Therefore , we have

$$P[Q_1(\tau + kd) = n] = \frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} \quad 0 < \tau \leq d, \;\; k = 0, 1, \dots \tag{86}$$

and

$$P[Q_1 = n] = \int_0^d \frac{1}{d}\frac{e^{-\lambda\tau}(\lambda\tau)^n}{n!} d\tau. \tag{87}$$

We use the following algebraic lemma to simplify the above integral:

Lemma 4 *For $n = 0, 1, \dots$ and λ and d positive real numbers, we have:*

$$\int_d^\infty \frac{\lambda e^{-\lambda\tau}(\lambda\tau)^n}{n!} d\tau = \sum_{i=0}^{n} \frac{e^{-\lambda d}(\lambda d)^i}{i!}. \tag{88}$$

proof: We use induction to prove the result. The equality obviously holds for $n = 0$. Now, if the equality holds for n, for $n+1$ we have:

$$\int_d^\infty \frac{\lambda e^{-\lambda\tau}(\lambda\tau)^{n+1}}{(n+1)!} d\tau = \left[-\frac{\lambda e^{-\lambda\tau}(\lambda\tau)^{n+1}}{(n+1)!}\right]_d^\infty + \int_d^\infty \frac{\lambda e^{-\lambda\tau}(\lambda\tau)^n}{n!} d\tau. \tag{89}$$

If the equality holds for n, the second term is the sum from 0 to n and the first term is the same term for $n+1$. Hence, the lemma follows.
Equation 87 can now be written as

$$P[Q_1 = n] = \frac{1}{\lambda d}\int_0^d \frac{\lambda e^{-\lambda\tau}(\lambda\tau)^n}{n!} d\tau = \tag{90}$$

$$\frac{1}{\lambda d}\left[1 - \int_d^\infty \frac{\lambda e^{-\lambda\tau}(\lambda\tau)^n}{n!} d\tau\right] = \tag{91}$$

$$\frac{1}{\lambda d}\left[1 - \sum_{i=0}^{n} \frac{e^{-\lambda d}(\lambda d)^i}{i!}\right] = \tag{92}$$

$$\frac{1}{\lambda d} P[X > n] \tag{93}$$

with X being a Poisson(λ) random variable.
Table 1 summarizes these properties. As we see, the waiting times are finite and independent of the arrival rate. This is a direct result of the infinite bulk capacity of the server. This fact can also be seen in a queueing system of the $M/M_1^\infty/1$ type where the

Parameter	Distribution	Mean
W_1	$\text{Unif}[0, d]$	$\frac{d}{2}$
Q_1	$\frac{1-F(n)}{\lambda d}$	$\frac{\lambda d}{2}$
W_2	constant	d
Q_2	$\text{Poisson}(\lambda d)$	λd

Table 1: Properties of a bulk service queue ($F(.)$: CDF of Poisson(λd) distribution)

service times are exponentially distributed with parameter μ and a service can start as soon as an arrival lands on the empty queue. This queue is Markovian and has a 2-dimensional Markov chain representation which we have analyzed and found the average total queue length and the average total waiting time to be

$$\bar{Q} = \frac{\rho(2\rho^2 + 2\rho + 1)}{\rho^2 + \rho + 1};$$

and

$$\bar{W} = \frac{Q}{\lambda}$$

where $\rho = \frac{\lambda}{\mu}$. Here, it can also be easily seen that as we expect, $\bar{W}$ approaches the finite value $\frac{2}{\mu}$ as $\lambda \to \infty$. In general, for a $M/G_1^\infty/1$ queue, an arrival is either served immediately (if it arrives to the empty queue), or will be served at next service which will be right after the end of the current service. If we denote by p_0 the probability of queue being empty, $\bar{X}$ the average service time and, by $\bar{R}$ the average residual service time seen by the (Poisson) arrivals, we have

$$\bar{W} = \bar{X}p_0 + (1 - p_0)(\bar{X} + \bar{R}) \tag{94}$$

Since $0 < p_0 < 1$ and $\bar{R} \leq \bar{X}$, we can bound the average waiting time by

$$\bar{X} < \bar{W} < 2\bar{X}. \tag{95}$$

In other words, the infinite service capacity of the server never allows the waiting time to be more than two service periods.

References

[1] A. Bar-Noy. Optimal broadcasting of two files over an asymmetric channel. *J. Parallel and Distributed Computing*, pages Vol. 60, pp474–493, 2000.

[2] 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.

[3] Dimitris Bertsimas and José Ni no Mora. Restless bandits, linear programming relaxations and a primal-dual index heuristic. *Operations Research, Vol. 48*, pp80-90, 2000.

[4] M. Chaudhry and J. Templeton. *A First Course in Bulk Queues.* Wiley, New York., 1983.

[5] Cidera. Http://www.cidera.com.

[6] D. R. Cox. *Queues.* Methuen's monographs on statistical subjects, New York, Wiley, 1961.

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

[8] H. D. Dykeman et. al. Scheduling algorithms for videotex systems under broadcast delivery. *IEEE Int. Conf. on Comm. ICC86*, Vol. 3,pp1847-51, 1986.

[9] K. Stathatos et. al. Adaptive data broadcast in hybrid networks. *Proc. 23rd VLDB conf.*, Athens, Greece, 1997.

[10] M. J. Donahoo et. al. Multiple-channel multicast scheduling for scalabel bulk-data transport. *INFOCOM'99*, pp847-855, 1999.

[11] Q. Hu et. al. Dynamic data delivery in wireless communication environments. *Workshop on Mobile Data Access*, pp213-224, Singapore, 1998.

[12] R. Epsilon et. al. Analysis of isp ip/atm network traffic measurements. *Perf. Eval. Rev.*, pages pp15–24, Vol. 27, No. 2, 1997.

[13] S. Acharya et. al. Balancing push and pull for data broadcast. *Proc. ACM SIGMOD, Tuscon, Arizona.*, 1997.

[14] J. C. Gittins. *Multi-Armed Bandit Allocation Indices.* John Wiley & Sons, 1989.

[15] J. C. Gittins. Bandit processes and dynamic allocation indices. *J. Roy. Statist. Soc.*, Vol. 41, pp148-177, 1979.

[16] D. M. Jones J. C. Gittins. A dynamic allocation index for the sequential design of experiments. *Progress in Statistics, Euro. Meet. Statis., Vol. 1, J. Gani et. al. Eds.*, New York, North-Holland, 1974, pp241-266.

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

[18] Michael N. Katehakis and Arthur F. Veinott Jr. The multiarmed bandit problem: decomposition and computation. *Mathematics of Operations Research, Vol. 12, No. 2*, pp262-268, 1987.

[19] G. P. Klimov. Time-sharing service systems. ii. *Theory of Probability and its applications, Vol. XXIII, No. 2*, pp314-321, 1978.

[20] G. P. Klimov. Time-sharing service systems. i. *Theory of Probability and its applications, Vol. XIX, No. 3*, pp532-551, 1974.

[21] J.W. Wong M.H. Ammar. The desgning of teletext broadcast cycles. *Perf. Eval. Rev.,*, pages Vol. 5, pp235–242, 1985.

[22] J.W. Wong M.H. Ammar. On the optimality of cyclic transmission in teletext systems. *IEEE Trans. Comm.*, pages Vol. 35, pp68–73, Jan. 1987.

[23] S. Hameed N. Vaidya. Scheduling data broadcast in asymmetric communication environments. *Tech. report TR96-022, Dept. Computer Sci. Texas A and M Univ.*, 1996.

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

[25] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of optimal queueing network control. *Mathematics of Operations Research, Vol. 24, No. 2*, pp293-305, 1999.

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

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

[28] B. Ryu. Modeling and simulation of broadband satellite networks- part ii: Traffic modeling. *IEEE Comm. Mag.*, Vol. 3, No. 7, July 1999.

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

[30] DirecPc System. Http://www.direcpc.com.

[31] N. Vaidya and H. Jiang. Data broadcast in asymmetric wireless environments. *Proc. 1st Int. Wrkshp Sat.-based Inf. Serv.(WOSBIS). NY*, Nov. 1996.

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

[33] 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.

[34] P. Whittle. Arm-aquiring bandits. *Ann. Prob.*, 9, pp284-292, 1981.

[35] P. Whittle. Multi-armed bandits and the gittins index. *J Roy. Statist. Soc. Ser. B.*, Vol. 42 pp143-149, 1980.