PH.D. THESIS

Broadcast Scheduling in Information Delivery Networks

by Majid Raissi-Dehkordi
Advisor: John S. Baras

CSHCN PhD 2002-7
(ISR PhD 2002-12)

ABSTRACT

Title of Dissertation: BROADCAST SCHEDULING IN INFORMATION DELIVERY NETWORKS

Majid Raissi-Dehkordi, Doctor of Philosophy, 2002

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

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 dissertation, 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.

Since the equal file length assumption is not appropriate for applications such as cache broadcasting in the Internet delivery systems, we also investigate an extension of the problem where the files have random lengths. After investigating some of the properties of the optimal solution, we derive an asymptotically optimal index policy for that case as well. Also, through simulation studies, the performance of the policy is compared with that of some other heuristic polices designed by intuitive arguments. The index policy is also extended to systems with deterministic, unequal file sizes and its performance is evaluated and compared to other policies via simulation studies.

The formulation and analytical procedures used in deriving the index policies in this dissertation allow for introduction of other extensions of the problem like assigning weights to the data files (studied in Chapter 3) or taking into account the channel errors and correlation between the arrivals. We will present our formulation of the last two extensions and discuss some of the numerical results to motivate future work on these problems.

BROADCAST SCHEDULING IN

INFORMATION DELIVERY NETWORKS

by

Majid Raissi-Dehkordi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2002

Advisory Committee:

Professor John S. Baras, Chairman/Advisor
Dr. Richard J. La
Dr. Tzong-yow Lee
Professor Armand M. Makowski
Dr. Babis Papadopoulos

DEDICATION

To my parents

for their continuous dedication and encouragement and,

to my wife

for her endless love and support.

ACKNOWLEDGEMENTS

I would like to thank my advisor, professor John S. Baras for his support and encouragement during my studies. He gave me the opportunity to work on different subjects and problems and acquire a lot of knowledge and experience.

I am thankful to the members of the dissertation committee, Professors Richard La, Tzong-yow Lee, Armand M. Makowski and Babis Papadopoulos for kindly reviewing this dissertation and serving on my thesis committee.

I am specially grateful to Professor Armand M. Makowski for his help and advice through different stages of this research.

My thanks also go to my friends, colleagues and staff at the Center for Satellite and Hybrid Communication Networks, Institute for Systems Research and Electrical and Computer Engineering department for their advice, support and help.

I am also grateful for the support of my research work and graduate studies through the following contracts and grants (all with the University of Maryland at College Park): NASA NAG59150, Lockheed Martin corporation (fellowship), the Lockheed Martin Chair Endowment, the Maryland Industrial Partnership program, Hughes Network Systems corporation and the state of Maryland.

Last, but not the least, I express my sincere gratitude to my parents for everything they did for me in my life and to my wife Sarvin for her support and patience during my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 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 [19, 51], 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

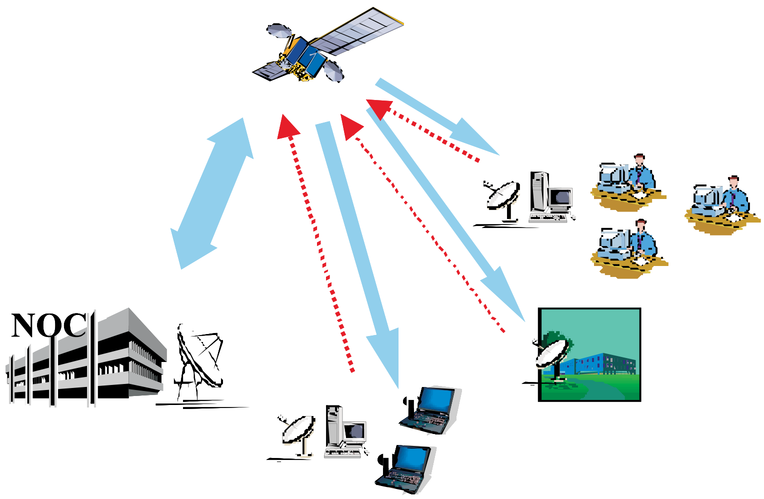

Figure 1.1: Typical architecture of a satellite information delivery system.

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 1.1 and 1.2 show two examples of these type of systems. In both systems, we assume that all the users who are waiting for a specific 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

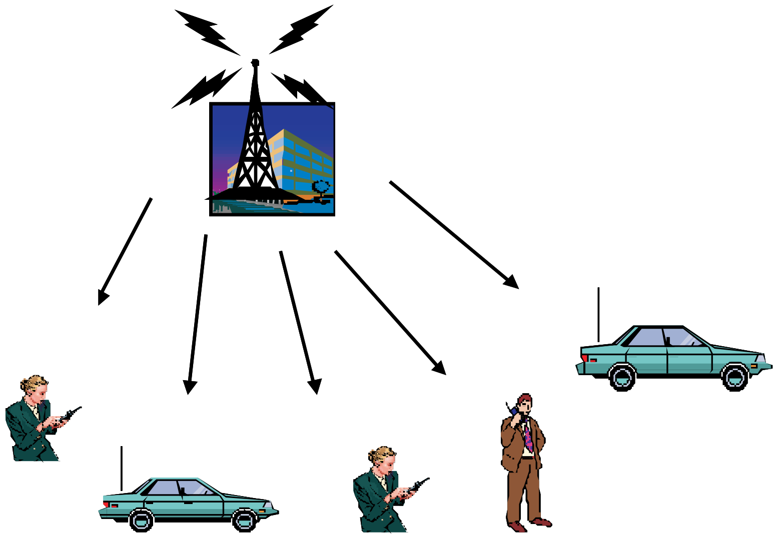

Figure 1.2: Typical architecture of a wireless information delivery system.

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 infor-

[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

mation 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 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 [55], Cidera [9]) 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 [27, 20, 15]. 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.
The above work was initially motivated by its applications in the DirecPc or other satellite data delivery systems to improve the performances of their Internet delivery services in future. Parallel to this effort, we also worked on another project that dealt with the analysis of the traffic in the current systems which mainly takes advantage of the large coverage of the satellite system. In that project, we collected the traffic flowing at the hybrid gateway of the DirecPc system and performed a statistical analysis of that traffic and derived several useful quantitative as well as qualitative results about its components. Those results were used later in other projects for more realistic performance evaluation studies on that system. Therefore, the above problems are related in the sense that they both address the problems associated with the same system.

1.1 Contents and organization of the thesis

This thesis is organized as follows. Following the literature review provided in this chapter, Chapter 2 serves as the main building block of our work. In that chapter, after a detailed review of the previous works on broadcast scheduling, the mathematical formulation of the standard form of the problem where all files have equal lengths as a Markov Decision Problem(MDP) is introduced. In Chapter 3, a separate section is dedicated to reviewing the principles of our approach that is based on the *Restless Bandit* problem formulation [61]. After proving the required properties, we find both the exact and approximate asymptotically optimal index policies for that problem. Finally, Chapter 3 ends with a detailed investigation of the performance of our policy where we compare our results with the results of other well-known heuristic policies. Chapter 4 uses the same method introduced in Chapter 3 to address the scheduling problem in a *pull* system when the files have different lengths. Unlike the standard form of the problem and despite the potential applications of the results to cache broadcast systems, to our knowledge, there has not been any previous work to address this problem before. We define the formulation of the problem and prove some of the properties of the optimal policy. We also extract an index function based on similar optimization arguments. The final section of Chapter 4 is dedicated to experimental evaluation of our policy. We compare the results of our policy with the performances of a number of other heuristic policies which we could come up with by extending the well-known heuristic policies used in other circumstances. In Chapter 5 we divert from the broadcast scheduling problem to present the results we obtained from statistical analysis of the traffic in the DirecPc system. Finally, Chapter 6 is dedicated to our formulation and preliminary results for some other extensions of the broadcast scheduling problem and suggestions for future work.

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.

Chapter 2

Broadcast scheduling, literature review and formulation of the problem

This chapter serves as the main building block of our approach for finding the optimal scheduling policy for different systems. First, we review the related work by other researchers on this topic and review their approaches. Next, we present our MDP formulation of the problem as a maximization problem and point to the shortcomings of this initial formulation as an N-dimensional problem.

We present the Bandit problem methodology and the reduction of the problem to N one-dimensional optimization problems. We then find the optimal solution for each one-dimensional problem and present the near-optimal solution to the original problem using those results. The final section of this chapter is dedicated to numerical results for evaluating the performance of our policy and comparing it to some other policies.

2.1 Related work

The series of works by Ammar and Wong are probably the first papers addressing the broadcast scheduling problem in detail. In [39, 40] 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 [57, 41] 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 [54] 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 [3] 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 [16, 18] 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 [14] 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 [12, 54] do not contain the notion of optimality. In [14], 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 [27], 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 [54], 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 [12] 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 [14] and [54] 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* [61] 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.

2.2 Problem formulation

In our formulation, we denote by $N(> 1)$, the number of information packages stored in the system. In this chapter we present the formulation of the case where all packages have equal sizes. This assumption is also made in [54, 14, 12] 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 pending requests for that package, and K servers as in figure 2.1. Due to the broadcast nature of the system, the queues are of the bulk service type [8] 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 $X_i(t)$

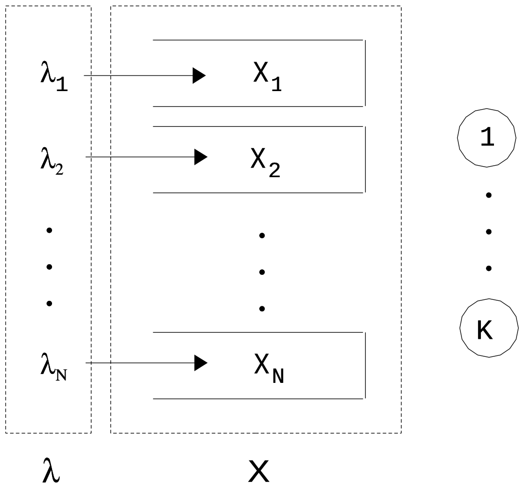

Figure 2.1: The pull type broadcast as a queuing system.

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

$$X_i(t+1) = X_i(t) - X_i(t)\mathbf{1}[i \in d(t)] + A_i(t) \tag{2.1}$$

where $d(t) \subset \{1, \ldots, N\}$ is the set containing the indices of the K pages broadcasted at time t. Figure 2.2 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

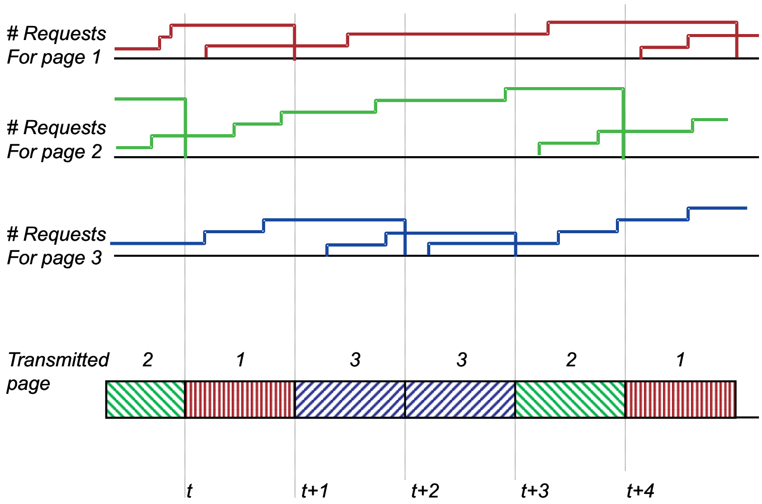

Figure 2.2: Sample path of a system with three pages.

law the average waiting time can be written as

$$\bar{W} = \frac{1}{\lambda} \sum_{i=1}^{N} c_i \bar{X}_i. \tag{2.2}$$

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 (2.2), we use the expected discounted reward criteria defined as

$$J_\beta(\pi) = E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i X_i(t)\right] \tag{2.3}$$

where π is the scheduling policy resulting in $J_\beta(\pi)$. Equations (2.3) and (2.1), with the initial condition $X(0)$, define the following DP problem with J^* denoting the optimal value defined as

$$J^*_\beta(\pi) = \min_\pi E\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i X_i(t)\right]. \tag{2.4}$$

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

$$(1-\beta)J_\beta(\pi) = E\left[\sum_{i=1}^{N} c_i X_i(0)\right] + \beta E\left[\sum_{t=0}^{\infty}\beta^t \sum_{i=1}^{N} c_i A_i(t)\right] - E\left[\sum_{t=0}^{\infty}\beta^t \sum_{i\in d(t)} c_i X_i(t)\right] \tag{2.5}$$

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

$$\widehat{J^*_\beta}(\pi) = \max_{\pi} E\left[\sum_{t=0}^{\infty}\beta^t \sum_{i\in d(t)} c_i X_i(t)\right]. \tag{2.6}$$

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

$$r(s,d) = \sum_{l\in d} c_l x_l \tag{2.7}$$

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 [48]) 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

$$L(V(s)) = \max_{d\in D}\left[r(s,d) + \beta \sum_{s'\in S} p^d(s,s')V(s')\right] \quad \forall s \in S \tag{2.8}$$

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

$$V(s) = L(V(s)) \quad \forall s \in S. \tag{2.9}$$

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.

2.3 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 (2.9) 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 2.3.1 *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*

$$S_p^d(\mathbf{x}) \leq S_p^d(\mathbf{x}+\mathbf{e}_i) \leq c_i + S_p^d(\mathbf{x}). \tag{2.10}$$

where $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0)$ *is the unit vector in* R^N *with* i*th 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 2.3.2 *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 2.3.1 we have

$$S_p^{d_p^{\mathbf{y}}}(\mathbf{y}) \leq S_p^{d_p^{\mathbf{y}}}(\mathbf{x}) + c_i \tag{2.11}$$

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

$$V(\mathbf{y}) \leq c_i + \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{x}). \tag{2.12}$$

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

$$V(\mathbf{x}) = \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{x}) \geq \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{x}). \tag{2.13}$$

inequality (a) follows by combining the two above results.
Also, according to lemma 2.3.1 we have

$$S_p^{d_p^{\mathbf{x}}}(\mathbf{x}) \leq S_p^{d_p^{\mathbf{x}}}(\mathbf{y}) \tag{2.14}$$

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

$$V(\mathbf{x}) \leq \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{y}). \tag{2.15}$$

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

$$V(\mathbf{y}) = \sum_p P(p) S_p^{d_p^{\mathbf{y}}}(\mathbf{y}) \geq \sum_p P(p) S_p^{d_p^{\mathbf{x}}}(\mathbf{y}). \tag{2.16}$$

Hence inequality (b) follows.

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

Theorem 2.3.3 *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

$$c_i x_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_j x_j + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N. \tag{2.17}$$

We need to show

$$c_i y_i + \beta \sum_A P(A) V(\mathbf{y} + A - y_i \mathbf{e}_i) \geq c_j y_j + \beta \sum_A P(A) V(\mathbf{y} + A - y_j \mathbf{e}_j) \quad j = 1, \ldots, N \tag{2.18}$$

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

$$c_i x_i + c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_j x_j + \beta \sum_A P(A) V(\mathbf{y} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N \; j \neq i. \tag{2.19}$$

From 2.17 we have

$$c_i x_i + c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_i \mathbf{e}_i) \geq c_i + c_j x_j + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j) \quad j = 1, \ldots, N. \tag{2.20}$$

Also , from lemma 2.3.2 part(a) we have

$$V(\mathbf{y} + A - x_j \mathbf{e}_j) \leq c_i + V(\mathbf{x} + A - x_j \mathbf{e}_j) \tag{2.21}$$

or

$$\beta \sum_A P(A) V(\mathbf{y} + A - y_j \mathbf{e}_j) \leq c_i + \beta \sum_A P(A) V(\mathbf{x} + A - x_j \mathbf{e}_j). \tag{2.22}$$

From (2.20) and (2.22), equation (2.19) 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 (2.9) only reveals limited properties of the optimal policy. Our main approach in this dissertation requires some background from the Bandit problems and also a few properties of the bulk service queueing systems. Therefore, we end this section at this point and explain our main approach in the following chapter after providing the necessary material.

Chapter 3

Preliminary properties, Bandit problem formulation and results

This chapter serves as the main building block of this dissertation. Here we explain our approach for addressing the broadcast scheduling problem and the resulting scheduling policy. However, this formulation is only useful when the underlying system possesses certain properties. Therefore, we dedicate the first section to exploring some of the properties of bulk service queueing systems which will be used later in this chapter. That will be followed by a brief review of stochastic scheduling and our main approach i.e., *Restless Bandit problem* formulation. Finally, we present the resulting policy and and an extensive evaluation of its performance compared with other well-known policies.

3.1 Properties of controlled bulk service queues

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 [8] 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 B for the interested readers. 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

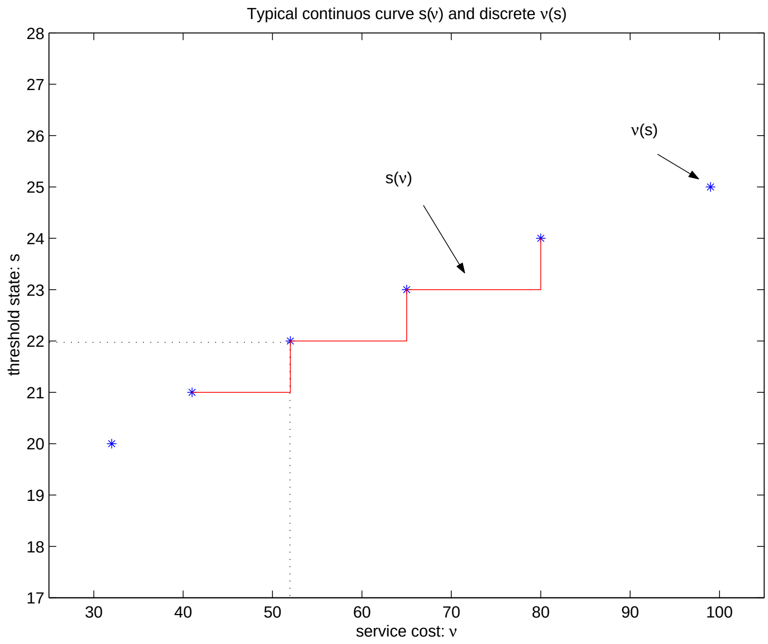

Figure 3.1: Typical $s(\nu)$ and $\nu(s)$ curves.

Property 3.1.1 *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 3.1.2 *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 3.1).*

Proof: Appendix B.

Instead of finding the threshold state for every value of ν, we can assign to every state s(figure 3.1), 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.

3.2 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 *Multi-armed 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 $R(t) = r_i\left(s_i(t)\right)$ 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

$$E\left[\sum_{t=1}^{\infty} \beta^t R(t)\right] \tag{3.1}$$

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 [26, 22] 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

$$\nu_i(s_i(t)) = \max_{\tau \geq t} \frac{E\left[\sum_{l=t}^{\tau} \beta^l r_i\left(s_i(l)\right)\right]}{E\left[\sum_{l=t}^{\tau} \beta^l\right]} \tag{3.2}$$

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 [21, 63, 30] as well as extensions to the original problem [62, 58, 32, 31] 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 [62, 58] which resulted in the so-called $c\mu$ rule as the optimal policy (also through other approaches e.g. [4, 10]). However, we were not able to use any of those formulations for our bulk-service scheduling problem. We therefore use what Whittle [61] 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.

3.3 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 [61, 43, 7, 44] 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 [48] converts the original dynamic programming problem

$$V(s) = \max_{d \in D} \left[r(s,d) + \beta \sum_{j \in S} p^d(j|s) V(j) \right] \quad \forall s \in S \tag{3.3}$$

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

$$z(s,d) = E\left[\sum_{t=0}^{\infty} \beta^t I[x(t) = s \,\&\, d(t) = d]\right] \tag{3.4}$$

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

$$Maximize \sum_{n=1}^{N} \left[\sum_{s \in S_n} r_n(s,0) z_n(s,0) + \sum_{s \in S_n} r_n(s,1) z_n(s,1)\right] \tag{3.5}$$

subject to

$$\sum_{l \in \{0,1\}} z_n(j,l) - \sum_{s \in S_n} \sum_{l \in \{0,1\}} \beta p_n^l(j|s) z_n(s,l) = \alpha_n(j) \quad for \;\; n = 1, \ldots, N \;\; and \;\; j \in S_n. \tag{3.6}$$

where

$$z_n(s,1) = E\left[\sum_{t=0}^{\infty} \beta^t I[x_n(t) = s \,\&\, n \in d(t)]\right] \tag{3.7}$$

and

$$z_n(s,0) = E\left[\sum_{t=0}^{\infty} \beta^t I[x_n(t) = s \,\&\, n \notin d(t)]\right] \tag{3.8}$$

and $p_n^1(j|s)$ ($p_n^0(j|s)$) 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

$$\sum_{n=1}^{N} \sum_{s \in S_n} z_n(s,1) = K/(1-\beta). \tag{3.9}$$

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

$$Maximize \ \sum_{n=1}^{N} \left[\sum_{s \in S_n} r_n(s,1) z_n(s,1) \right] + \nu \left(K/(1-\beta) - \sum_{n=1}^{N} \sum_{s \in S_n} z_n(s,1) \right) \tag{3.10}$$

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

$$Maximize \ \sum_{n=1}^{N} \left[\sum_{s \in S_n} (r_n(s,1) - \nu) z_n(s,1) \right] + K\nu/(1-\beta) \tag{3.11}$$

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 (3.11) can be decoupled into N separate problems

$$Maximize \sum_{s\in S_n}(r_n(s,1)-\nu)z_n(s,1) \tag{3.12}$$

subject to

$$\sum_{l\in\{0,1\}} z_n(j,l) - \sum_{s\in S_n}\sum_{l\in\{0,1\}} \beta p_n^l(j|s)z_n(s,l) = \alpha(j) \quad for \ n=1,\ldots,N \ and \ j\in S_n. \tag{3.13}$$

The solution to the Lagrangian Relaxation problem (3.11) is a function of the parameter ν and is an upper bound to the solution of problem (3.5) 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 (3.12). 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 3.1 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 [50] 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.

3.4 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 $\nu(s)$.

$$V^s(0) = \beta \sum_{i=0}^{\infty} p(i) V^s(0+i) \tag{3.14}$$

$$
\begin{aligned}
V^s(1) &= \beta \sum_{i=0}^{\infty} p(i) V^s(1+i) && (3.15) \\
&\vdots \\
V^s(s) &= \beta \sum_{i=0}^{\infty} p(i) V^s(s+i) && (3.16) \\
V^s(s) &= -\nu(s) + cs + \beta \sum_{i=0}^{\infty} p(i) V^s(i) && (3.17) \\
V^s(s+1) &= -\nu(s) + c(s+1) + \beta \sum_{i=0}^{\infty} p(i) V^s(i) && (3.18) \\
&\vdots
\end{aligned}
$$

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 (3.17) is equal to $V^s(0)$ therefore we have

$$\nu(s) = cs + V^s(0) - V^s(s). \tag{3.19}$$

Equation (3.16) 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

$$V^s(s) = \frac{\beta c\lambda}{1-\beta} \tag{3.20}$$

and

$$V^s(s+i) = ci + \frac{\beta c\lambda}{1-\beta}. \tag{3.21}$$

Substituting (3.20) in (3.19) gives

$$\nu(s) = cs + V^s(0) - \frac{\beta c\lambda}{1-\beta}. \tag{3.22}$$

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

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

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

$$h(n) = \sum_{i=n+1}^{\infty} p(i) \tag{3.23}$$

and

$$m(n) = \sum_{i=0}^{n} ip(i). \tag{3.24}$$

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

$$A_s \mathrm{W}_s = B_s \tag{3.25}$$

where

$$A_s = \begin{bmatrix} 1-\beta p(0) & -\beta p(1) & \ldots & -\beta p(x) & \ldots & -\beta p(s-1) \\ 0 & 1-\beta p(0) & \ldots & -\beta p(x-1) & \ldots & -\beta p(s-2) \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & 1-\beta p(0) & \ldots & -\beta p(s-x-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 & \ldots & 1-\beta p(0) \end{bmatrix} \tag{3.26}$$

and

$$B_s = \beta \begin{bmatrix} h(s-1)\left[\frac{\beta c\lambda}{1-\beta} - cs\right] + c\lambda - cm(s-1) \\ h(s-2)\left[\frac{\beta c\lambda}{1-\beta} - c(s-1)\right] + c\lambda - cm(s-2) \\ \vdots \\ h(s-x-1)\left[\frac{\beta c\lambda}{1-\beta} - c(s-x)\right] + c\lambda - cm(s-x-1) \\ \vdots \\ h(0)\left[\frac{\beta c\lambda}{1-\beta} - c\right] + c\lambda - cm(0). \end{bmatrix} \tag{3.27}$$

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 (3.22) it can be seen that the index function satisfies

Property 3.4.1 *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 (3.25) 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 3.4.2 *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 3.4.3 *For $V(0)$ values we have*
$V^{s+1}(0) = \frac{\beta}{1-\beta p(0)} \left[p(1)V^s(0) + \ldots + p(s)V^s(s-1) + \lambda + h(s)(\frac{\beta\lambda}{1-\beta} - s - 1) - m(s)\right]$.

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 3.4.4 *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 (3.22) we have

$$\nu(s+1) - \nu(s) = 1 + V^{s+1}(0) - V^s(0) \tag{3.28}$$

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

$$V^{s+1}(0) - V^s(0) = \frac{\beta}{1-\beta p(0)} \sum_{i=1}^{\infty} p(i)[V^s(i-1) - V^s(i)]. \tag{3.29}$$

Since $V^s(i-1) \leq V^s(i) \leq V^s(i-1) + 1;\ \ i = 1, 2, \ldots$(Lemma 2.3.2), we have

$$\frac{-\beta(1-p(0))}{1-\beta p(0)} \leq V^{s+1}(0) - V^s(0) \leq 0. \tag{3.30}$$

Using equation (3.28), we have

$$1 - \frac{\beta(1-p(0))}{1-\beta p(0)} \leq \nu(s+1) - \nu(s) \leq 1 \tag{3.31}$$

and since $\beta < 1$ the left hand term is always greater than 0 which completes the proof. This property was used in section 3.3 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).

3.5 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

$$
\begin{aligned}
V(0) &= \beta(1-p)V(0) + \beta p V(1) && (3.32)\\
V(1) &= \beta(1-p)V(1) + \beta p V(2) && (3.33)\\
&\vdots\\
V(s) &= \beta(1-p)V(s) + \beta p V(s+i) && (3.34)\\
V(s) &= -\nu + cs + \beta(1-p)V(0) + \beta p V(1) && (3.35)\\
V(s+1) &= -\nu + c(s+1) + \beta(1-p)V(0) + \beta p V(1) && (3.36)\\
&\vdots
\end{aligned}
$$

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

$$\nu(s) = cs + \frac{\beta c p}{1-\beta}\left[\left(\frac{\beta p}{1-\beta(1-p)}\right)^s - 1\right]. \tag{3.37}$$

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 3.2. 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.

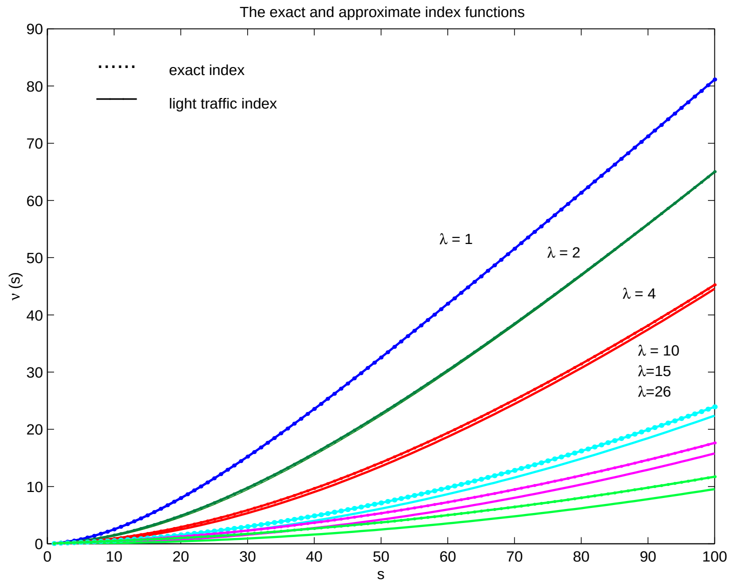

Figure 3.2: The exact index function and the light traffic approximation.

3.6 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 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 [54] 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

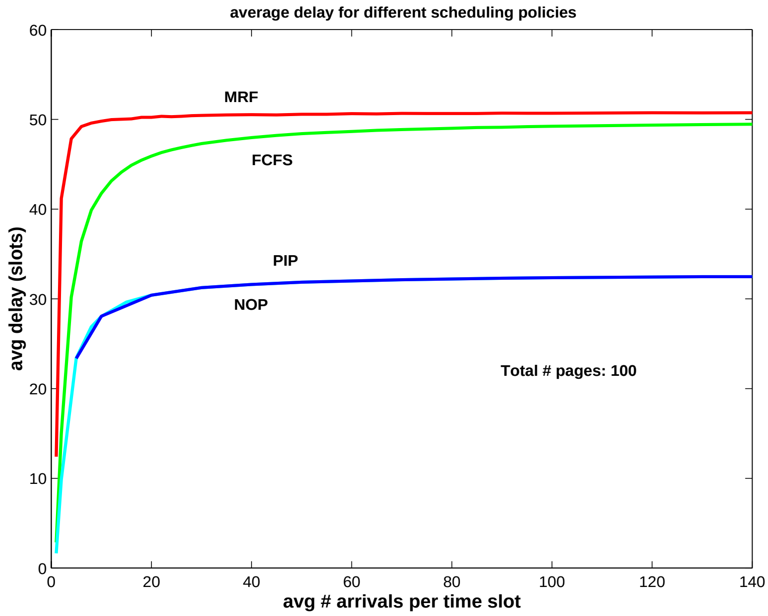

Figure 3.3: Comparison of the total average waiting time for different scheduling policies with the distribution of the arrival rates having a Zipf distribution.

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 3.3 shows the results of these experiments. As we observe, the performance of our policy is much better than

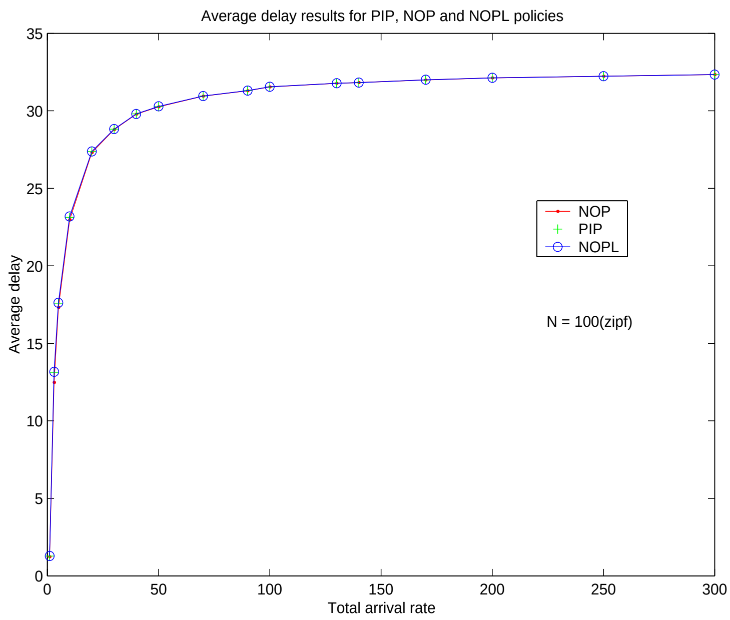

Figure 3.4: Performance comparison of the PIP, NOP and NOPL policies.

MRF and FCFS and identical to PIP. Also in figure 3.4 the performances of PIP and NOP policies are compared with the light traffic approximation of the NOP index which we call NOPL. It 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 3.5). Figures 3.6 and 3.7 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

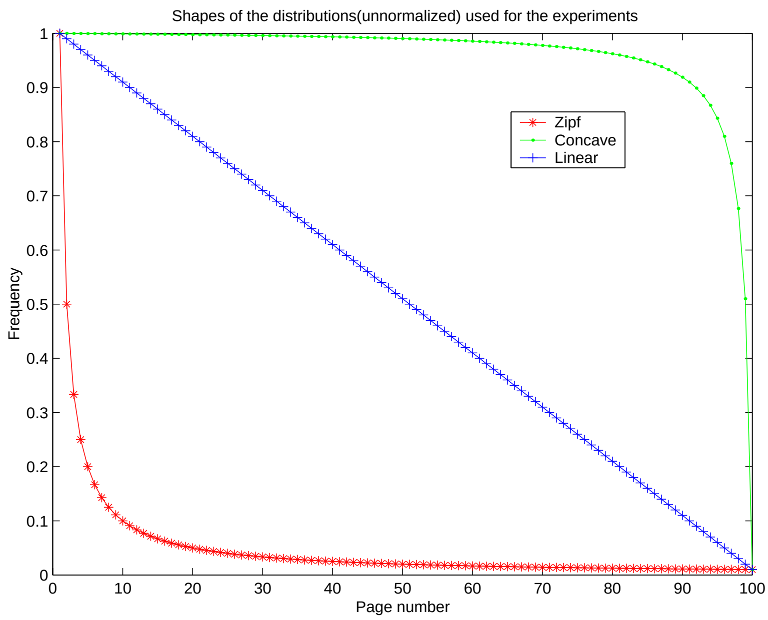

Figure 3.5: The Zipf distribution and the other two distributions used as the probability of the different pages in the experiments.

function, any monotonic increasing function of that index can be used as an index as well. The results suggest that what Su [54] 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 3.8 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. Therefore, we tried to use the same analogy and extend its definition so that the weight

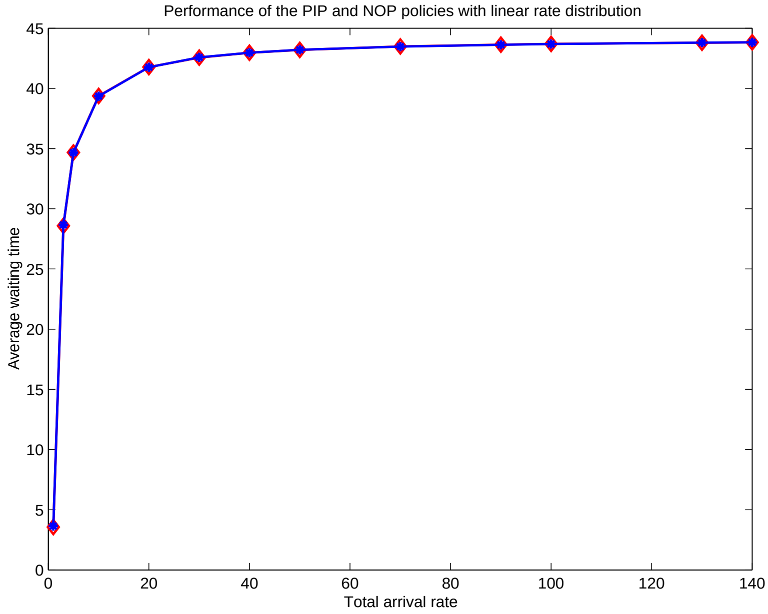

Figure 3.6: Comparison of the total average waiting time for PIP and NOP scheduling policies with the distribution of the arrival rates having a linear shape.

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 3.9 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

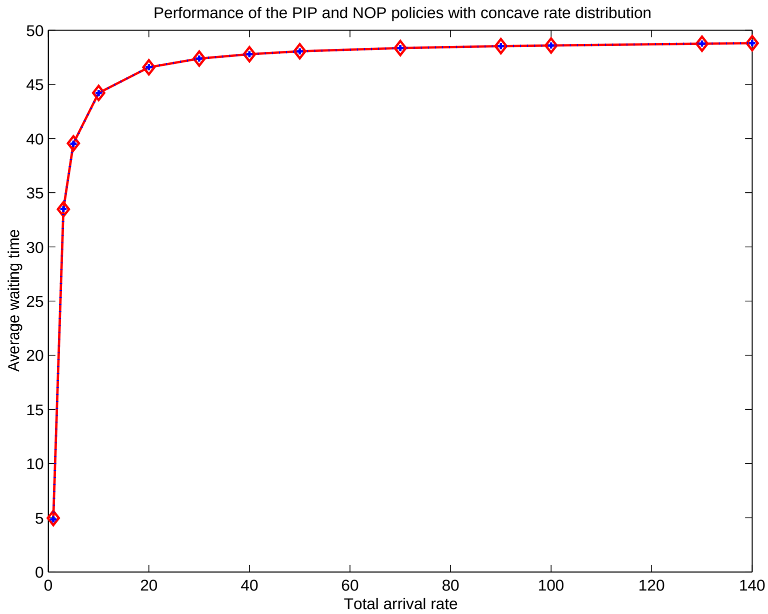

Figure 3.7: Comparison of the total average waiting time for PIP and NOP scheduling policies with the distribution of the arrival rates having a concave shape.

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.

3.7 Conclusion

In this chapter we derived a scheduling policy for the scheduling of broadcasts in a *pull* system with equal file sizes. The policy defines an index function for each page in the system and at every decision instant the first K pages with the largest values of the index are broadcast. The performance of our policy is almost identical to the performance of the PIP policy however, since we have taken an

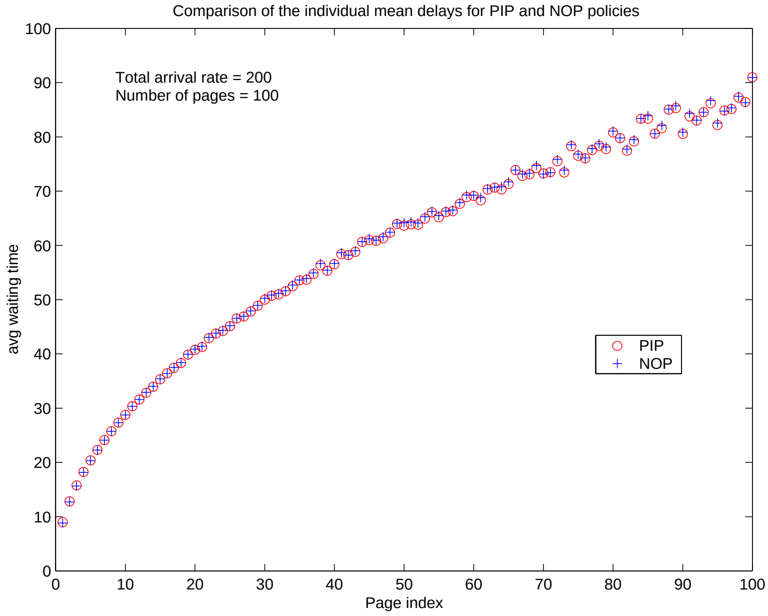

Figure 3.8: Average waiting times for the requests for each of the 400 pages under different policies.

analytical path for its derivation, it can be readily applied to cases with non-Poisson arrivals or when there are priority weights assigned to the pages. Other policies, due to their heuristic reasoning , do not address these general cases. Our approach shows that the index function scales linearly with the c coefficient. Using this result and through a number of experiments, we also introduced a heuristic extension to the PIP to include the weighted case as well. Another advantage of our approach is the guidelines it provides to consider the scheduling problem for a system where the pages do not have fixed lengths. This case is particularly of interest for cache broadcasting in the Internet and the previous methods do not address this important case. Chapter 4 is dedicated to the analysis of this type of *pull* systems.

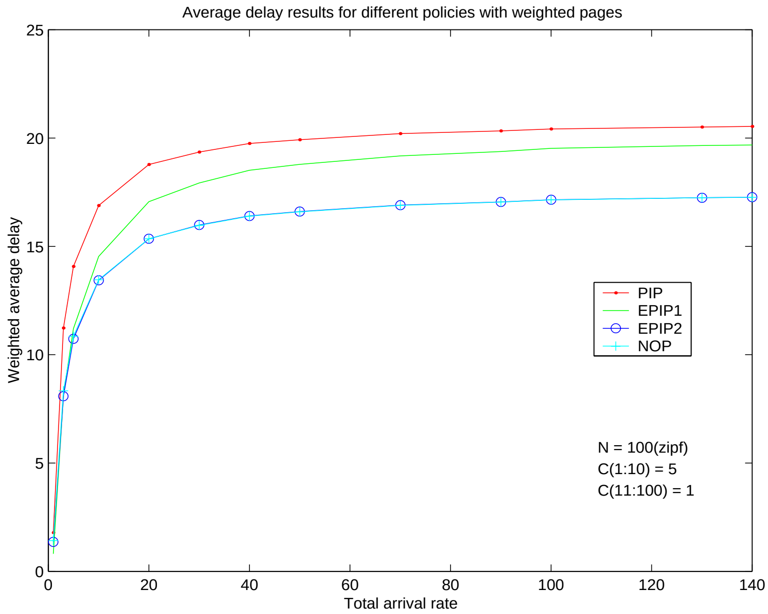

Figure 3.9: Performance comparison of NOP and different versions of the PIP policy for the weighted average delay case.

Chapter 4

Broadcast scheduling in systems with variable-length files

4.1 Introduction

In this chapter we investigate the problem of optimal scheduling of the broadcast messages in a system where the pages do not have equal sizes. This situation is more realistic than the fixed-length setting when we need to investigate the scheduling of the cached web pages. Caching is a popular method for improving the download time in the Internet. In a system with caching, a local cache stores copies of the most popular pages visited by the users and serves the future requests locally by sending the stored copy to them. Obviously, the proper updating mechanism should be implemented by the cache system to guarantee the validity of the information. Since the cached pages are the most popular pages, they automatically become the most eligible pages for broadcast delivery. Therefore, in a satellite or centralized wireless system, the issue can be further improved by broadcasting the cached pages instead of one-to-one transmissions. However, since here the information packages are different web pages, the assumption of all of them having equal sizes is not valid anymore and pages should be allowed to have different and even potentially random sizes. The fixed-size assumption makes the formulation too complicated and virtually intractable. Therefore, we start with the case where the page lengths are Geometric random variables with possibly different mean values. Later in this report, we use the obtained results to consider the fixed-length case as well.

4.2 Related work

Unlike the previous case with equal file sizes, where we could refer to the works by many other researchers, to our knowledge there has not been any published work on this more general setting, nor any of the previous works have tried to extend their heuristic algorithms to accommodate this new extension of the problem. However, we may refer to Vaidya [41] who studied the effect of having pages with different sizes in a *push* system and presented a method for designing the broadcast cycle for that case.

4.3 Problem formulation

Here we present our formulation of the *pull* broadcast scheduling with different file sizes. We denote by $N(> 1)$, the number of information pages stored in the system. Similar to the previous case, 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 difference between this new case and the previous one is that the transmission of a page does not necessarily end in one broadcast period and can span several periods depending on the size of the page. The page sizes are assumed to have Geometric distributions with parameter q_i for type i pages. In other words

$$P[broadcast\,of\,page\,i\,spans\,n\,time\,units] = q_i(1-q_i)^{n-1},\ 0 < q_i < 1,\ \ i = 1,\dots,N. \tag{4.1}$$

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 can be broadcast, 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 $t = 0, 1, \dots$ is a decision time (and also a possible preemption time). Due to the preemption assumption, the full transmission of a page of length n can take longer than n periods. Hence the waiting time of the requests for a page ends after 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 that page, need to wait till the beginning of the next transmission of the whole page. We also assume that the system has $K(1 \le K < N)$ identical broadcast channels. In this *pull* broadcast system, the system receives the requests for all pages 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.

Here, again we assume that the request arrival process for each page $i; i = 1, \dots, N$ is a Poisson process with rate λ_i. This assumption allows us to study the discrete-time system where all the arrivals during every time period arrive as a batch at the end of that time period. Obviously, the batch sizes are random variables with $Poisson(\lambda_i)$ distribution. The state of this system at each time t is shown by $\mathbf{X}(t) = (X_1(t), Y_1(t), X_2(t), Y_2(t), \ldots, X_N(t), Y_N(t))$: 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 beginning of 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

$$(X_i(t+1), Y_i(t+1)) = \begin{cases} (0, Y_i(t) + A_i(t)) & with\ prob. \quad q_i \qquad if\ i \in d(t) \\ (X_i(t), Y_i(t) + A_i(t)) & with\ prob. \quad (1 - q_i) \quad if\ i \in d(t) \\ (X_i(t), Y_i(t) + A_i(t)) & \qquad\qquad\qquad\qquad if\ i \notin d(t) \end{cases} \tag{4.2}$$

if $X_i(t) > 0$ and

$$(X_i(t+1), Y_i(t+1)) = \begin{cases} (0, A_i(t)) & with\ prob. \quad q_i \qquad if\ i \in d(t) \\ (Y_i(t), A_i(t)) & with\ prob. \quad (1 - q_i) \quad if\ i \in d(t) \\ (0, Y_i(t) + A_i(t)) & \qquad\qquad\qquad\qquad if\ i \notin d(t) \end{cases} \tag{4.3}$$

if $X_i(t) = 0$. Here $A_i(t)$; $i = 1, \ldots, N$ is the discrete time arrival process of the new requests for page i during the time interval $[t, t+1)$ and $d(t) \subset \{1, \ldots, N\}$ is the set containing the indices of the K pages broadcast at time t. Figure 4.1 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

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

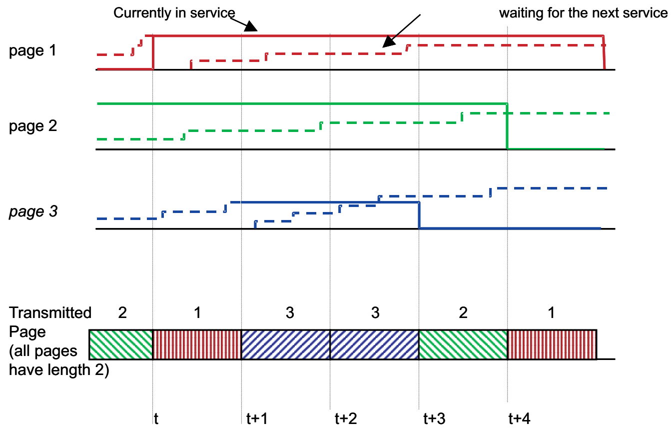

Figure 4.1: Sample path of a system with three pages.

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. 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

$$\bar{W} = \frac{1}{\lambda}\sum_{i=1}^{N} c_i(\bar{X}_i + \bar{Y}_i). \tag{4.4}$$

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. Similar to the previous problem, instead of minimizing (4.4), we use the total discounted reward criteria and try to minimize the total discounted expected number of waiting requests defined as

$$J_\beta(\pi) = E[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i(X_i(t) + Y_i(t))] \tag{4.5}$$

where π is the scheduling policy resulting in $J_\beta(\pi)$. Equations (4.5) and (4.2), together with the initial condition $(X(0), Y(0))$, define the minimization problem

$$J^*_\beta(\pi) = \min_{\pi} E[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{N} c_i(X_i(t) + Y_i(t))]. \tag{4.6}$$

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

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

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

$$\widehat{J_\beta}(\pi) = \max_{\pi} E\left[\sum_{t=0}^{\infty}\beta^t \sum_{i\in d(t)} q_i c_i(X_i(t)I[X_i(t)>0]+Y_i(t)I[X_i(t)=0])\right]. \quad (4.7)$$

To facilitate the analysis, we assume that the state space of each queue $i;\ i = 1,2,\ldots,N$ is a finite set S_i and denote the state space of the system by $S = S_1 \times S_2 \times \ldots \times S_N$. This problem is in fact a DP problem with decision space $D = \{d;\ d \subset \{1,2,\ldots,N\}\ and\ |d| = K\}$. The expected reward for broadcast of pages in $d \in D$ at state $s = \{x_1, y_1, \ldots, x_N, y_N\} \in S$ is

$$r(s,d) = \sum_{i\in d} q_i c_i(x_i I[x_i > 0] + y_i I[x_i = 0]) \quad (4.8)$$

In the following, we show the optimal value function of this problem by is $V(s);\ s \in S$ which satisfies the optimality equation

$$V(s) = \max_{d\in D}\left[r(s,d) + \beta \sum_{s'\in S} p^d(s,s')V(s')\right] \quad \forall s \in S \quad (4.9)$$

where $p^d(s,s')$ is the stationary transition probability of going from state s to state s' under decision d.

We use the Restless Bandit approach to address this problem too. That formulation allows us to reduce the original $2N$-dimensional problem into N similar

2-dimensional problems and focus on finding either analytical or experimental solutions for the latter. However, due to the complexity of the state transitions, this case is much more complicated than the previous case. As before, we first explore some of the properties of a single-queue system and will use the results for the original problem.

4.4 Some properties of a single controlled bulk service queue

In this section we examine the single-queue sub-problem. Imagine one of our bulk service queues with Poisson arrivals and Geometric 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. Here the reward obtained by serving the queue at any time is equal to the number of customers in the queue with probability q and is 0 otherwise. There is also a fixed cost ν for each service. 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:

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

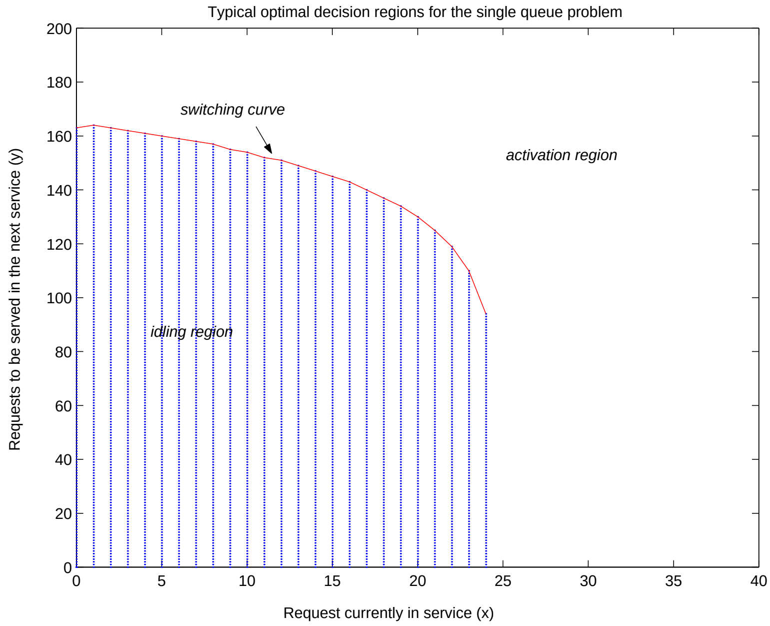

Figure 4.2: Typical shapes of the idle and active regions for a single queue problem.

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

$$R(t) = \begin{cases} cx(t) - \nu & with\, prob.\, q \quad if\;\; d(t) = 1\; and\; x(t) > 0 \\ cy(t) - \nu & with\, prob.\, q \quad if\;\; d(t) = 1\; and\; x(t) = 0 \\ -\nu & with\, prob.\, 1 - q \quad if\;\; d(t) = 1 \\ 0 & if\;\; d(t) = 0 \end{cases}$$

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.

In order to be able to apply the Restless Bandit approach and find an index policy for this problem, we need to find if the single-queue 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 every time. Figure 4.2 shows one example of the form of the optimal policy with the idle and active regions distinguished. The monotonicity property requires that as the service cost ν in-

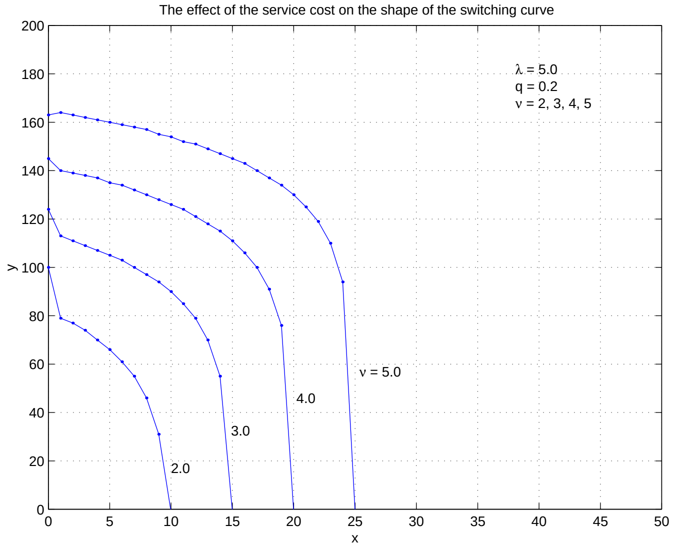

Figure 4.3: Investigating the effect of the ν parameter on the optimal decision space for a typical system.

creases from $-\infty$ to ∞, the idling region increases monotonically from $\emptyset$ to the whole state space. 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 4.3 shows a few results for this system with different ν values. The idling region for each case is the convex 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 4.4.1 *The optimal policy of the single-queue problem with bulk service and random file lengths as above is of the threshold type in the x direction. In* *other words, if $d(x,y)$ is the decision defined by the optimal policy for state (x,y) we have*

$$if\ \ d(x,y) = 1\ \ then\ \ d(x+i,y) = 1;\ \forall x > 0\ and\ i > 0; \tag{4.10}$$

Proof: Appendix C.

Unlike the above case, it is very difficult to prove the threshold property in the y direction. Proving the threshold property of the optimal policy of the DP problems is not a new topic. As examples, we can refer to papers by Lin and Kumar [36], Hajek [23], Koole [33], Viniotis and Ephremides [59] and many other works. To point out the difficulties of our specific problem we refer to the fact that in most of the above works, the threshold property is proved using related properties of the value function i.e., the sub(super)-modularity and convexity properties. Submodularity imposes an inequality constraint on the values of the value function of the optimal policy at any four neighbor points, namely

$$V(x,y+1) - V(x,y) \leq V(x+1,y+1) - V(x+1,y). \tag{4.11}$$

This property mostly arises in single-input single-output queueing systems where at each event the queue length either increases/decreases by one or remains unchanged. Unfortunately in our system, with the batch nature of both arrivals and departures, this property is no longer the key property for the threshold property. The problem of batch arrivals can be alleviated by assuming that the page sizes and the server speeds are such that the light-traffic assumption holds. But, the bulk service property of the queues requires a different inequality to be satisfied by the value function i.e.,

$$V(x,y+1) - V(x,y) \leq V(0,y+1) - V(0,y) \tag{4.12}$$

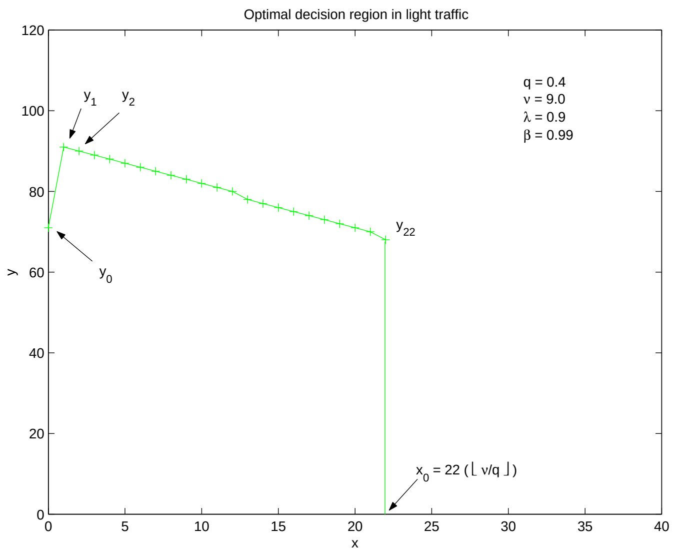

Figure 4.4: Typical form of the optimal policy for a single queue problem in light traffic.

which is much more difficult to prove. Nonetheless, despite the differences in the middle stages of our problem with the other works, the general methods used by them is still applicable to our problem. The two most popular methods for proving the structural properties of the solutions to DP problems use induction along with one of the Value Iteration or the Policy Iteration methods[48]. We examined both of these methods extensively and found that the Policy Iteration allows us to explore more properties of the optimal policy. The proof of the threshold property in the y direction can not be separated from other structural properties of the optimal policy. If we focus our attention on the light-traffic regime, we will find that the optimal policy $d(x, y)$ has certain properties. Defining $x_0 \triangleq \lfloor \frac{\nu}{q} \rfloor$, these properties can be listed as follows (figure 4.4).

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 $\beta >> 0$ we have $y_0 > x_0$.

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 simplifying assumption to make the problem more tractable. A general proof of these properties requires careful consideration of all cases that may happen for different problem settings. The most important factor is the value of y_0 with respect to the y_1 through y_{x_0} values. In appendix C we present a detailed discussion of our analysis of this problem for a typical case. Despite the interesting observations and results we obtained from this analysis, the proof is not yet complete. We were basically able to prove the properties 2,3 and 4 via the Policy Iteration arguments using the assumption that properties 1 and 5 are given. Although property 5 does not seem to be difficult to hold, property 1 is not easy to prove. Nonetheless, since our main concern is the validity of the monotonicity property in the typical cases that happen in our problem, we rely on our numerical results together with the partial results we obtained from theoretical analysis of the problem to apply the Restless Bandit approach to this problem. In the next section, we will use our findings from this section and appendix C to find a close approximation of the relation between the service cost ν and the switching curve. This will be the main step to calculate the index function and there we will also verify the monotonicity property via the approximate index function.

Before finishing this section, it is constructive to visually analyze the form of the optimal policy and the switching curve. Figure 4.5 shows the switching curves for various problem settings. Each service will produce an expected immediate reward of qx plus a future reward that would be generated by the current y component (plus possibly new arrivals) shifting to the x position. We will also have a fixed cost ν for each service. Therefore, it is reasonable to expect the optimal policy to be idle for small values of x and y and be active for larger values. We can also expect the activation level for the y component to be larger than that for the x component because of the discount factor β. Now, for larger values of service cost ν, larger immediate or future rewards are needed to make the *active* decision worthwhile, hence the idling region expands as seen in the top figure. Increasing the value of q increases the expected reward obtained from the *active* decision, therefore the idling region shrinks as in the middle figure. The effect of the arrival rate is not so obvious. As can be seen in the bottom figure, the idling region increases with the arrival rate only in the y direction. It can be justified by the fact that the arrival rate only affects the future rewards therefore, if the immediate reward (which only depends on x)is large enough to make the *active* decision worthwhile, the value of the arrival rate will not have any effect on that. However, for smaller values of x where we need to count on the future rewards. If the arrival rate is high it is better to wait for the next reasonably large batch of the arrivals to come and then serve all of them together, otherwise it will not be beneficial because it unnecessarily delays the current x and y customers and increases their total delays.

In the next section we will focus on the light-traffic regime to find an analytical expression for the index function.

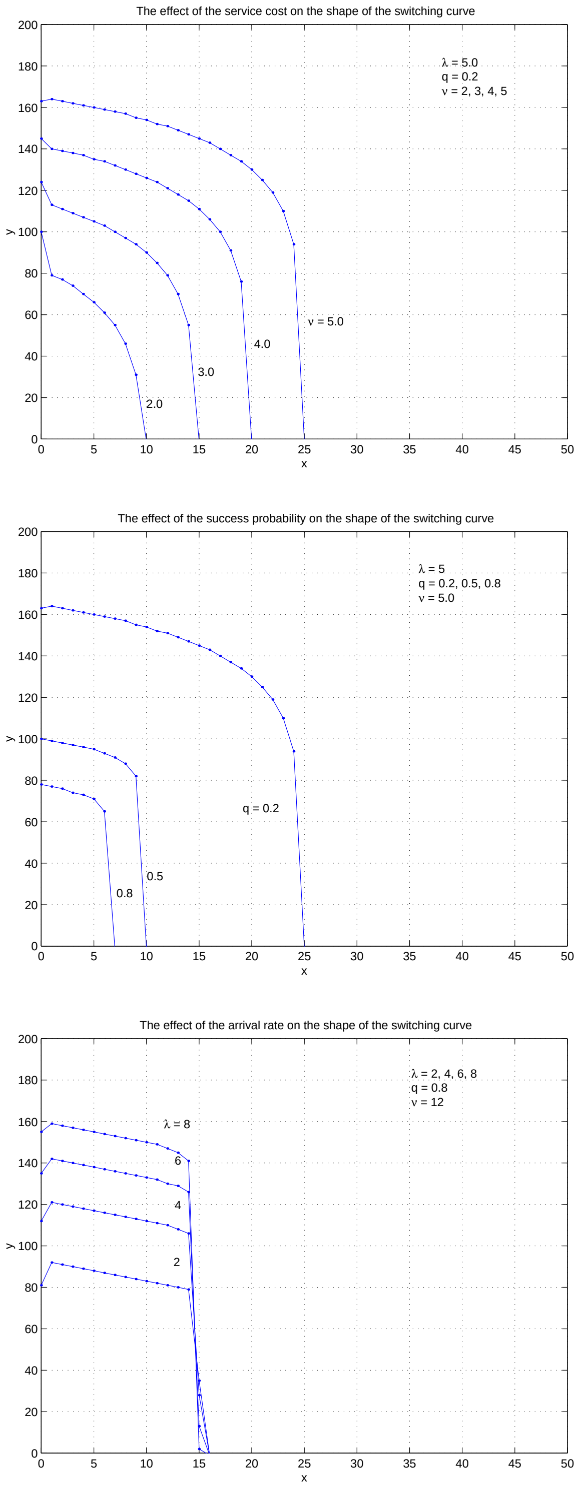

Figure 4.5: Typical shapes of the switching curve for a single queue problem.

4.5 Computation of the index

In the previous case of the broadcast scheduling problem with equal size files, the index was a function of the queue length x and arrival rate λ and we found a way to compute the index value. In that problem, even if we were not able to find a method for calculation of the index, we could still think of the numerical calculations for several values of the arrival rate and for a range of queue lengths and storing the values to be used by the scheduling algorithm. In the present system, however, the index is a function of four parameters x, y, λ and q and that approach is impossible. Therefore, we need to find an analytical approach to calculate the index.

Due to the batch nature of arrivals to each queue, the form of the index function is very complicated in the general case. However, by considering the light traffic case, very good approximations of this function can be found. The light traffic regime is particularly more useful for the *pull* delivery systems since the preference of these systems to *push* delivery is in low to moderate traffic regimes where the latter system may result in too many superfluous broadcasts.

In the light traffic regime, we assume that the probability of more than one arrival during a broadcast period is negligible and show by p_0 and $p_1 = 1 - p_0$ the probabilities of zero and one arrivals during a period, respectively. In this situation, it is possible to find an analytical formula that results into a method for computing the index value for each queue. The details of our derivations are shown in appendix C. We showed that for a fixed value of ν, q and λ, the switching curve can be defined by a point on the $x = 0$ axis, called y_0, a point on the $x = 1$ axis, called y_1, and the points defined by the integer approximation of a straight line with slope $-1/\beta a$, which starts from the $(1, y_1)$ point and ends on the $x = \lceil \nu/q \rceil$ line. The values of y_0 and y_1 satisfy the following equations

$$y_1 - y_0 = \frac{1}{\beta a}\left[\frac{\nu}{q} - 1\right] - \frac{ap_1}{q} \tag{4.13}$$

$$y_0 = \frac{\nu}{q} + \frac{\beta p_1}{1-\beta}\left(1 - c^{-y_0}\right) \tag{4.14}$$

where

$$a = \frac{q}{1-\beta(1-q)}$$

$$c = \frac{1-\beta p_0}{\beta p_1}.$$

For the scheduling problem, we need to find the required ν that puts a given (x, y) pair on the switching curve and the above relations can be used inversely. First, the corresponding $(1, y_1)$ point can be found from

$$y_1 = y + (x-1)/\beta a. \tag{4.15}$$

Next, y_0 can be calculated by solving the following equation

$$y_1 + \frac{1}{\beta a}\left[\frac{\beta p_1}{1-\beta} + 1\right] + \frac{ap_1}{q} = y_0\left(1 + \frac{1}{\beta a}\right) + \frac{p_1}{a(1-\beta)}c^{-y_0}. \tag{4.16}$$

Having found the value of y_0, the corresponding ν is

$$\nu = qy_0 - \frac{q\beta p_1}{1-\beta}\left(1 - c^{-y_0}\right). \tag{4.17}$$

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 found from the last equation above.

An interesting observation at this point is to compare the above light traffic approximation to the same approximation for the equal file size case we considered in a previous work. The equal file size case is in fact a special case of the current setting with $q = 1$. In other words, serving the queue is always successful and therefore the x component is always zero. Since the state space collapses into only the y axis, we expect the value of ν for a certain y_0 to be the same as the previous case. In fact that is exactly the case and both formulas are equal for $q = 1$ which may be seen as a consistency check for our calculations.

For the general problem with arbitrary arrival rate, the location of y_0 and y_1 and also the right border does not change much from the light traffic approximation (with p_1 replaced by λ) but the straight line tends to "bend" downward. We have also derived an experimental second order correction factor that changes the straight line to a parabolic curve closer to the real curve in higher rates. However, in our experiments the light traffic approximation without the correction factor gave satisfactory results for the range of arrival rates we considered. This method can be used as a near-optimal scheduling method but our next goal after finding this policy is to find a lower-complexity heuristic index policy which perform equally well compared to this policy. In the next section, we present the results of our experiments with a number of different heuristic policies.

Our light traffic approximation of the switching curve also allows us to investigate the changes in the curve as ν increases. In fact, we can state the following property

Property 4.5.1 *Both of y_0 and y_1 values are non-decreasing functions of ν.*

The proof of this property which uses the above equations is provided in appendix C. Since the slope of the upper border of the idling region is independent of ν and also the right border is a non-decreasing with ν ($x = \nu/q$), we can state:

Property 4.5.2 *The idling region is a non-decreasing function of ν and, for each value of ν, it contains the idling regions associated with smaller ν values.*

Although it is not a proof for the general case, its correctness for the general case is supported both by intuition (thinking of ν as the service cost) and by the results of a large number of numerical calculations.

4.6 Results

In this chapter we compare the results of our index policy for several experiments with the results from using 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, through our experiments, we tested some well-known policies used in simpler broadcast systems, which allowed us to evaluate the performance of our policy compared to those policies. Our goal was two-fold. We were interested in finding if any of those policies can beat our policy and, we were also looking for possible low complexity policies which perform close to our policy in different situations, hence being proper candidates for practical applications. Another set of experiments was performed in a slightly different system than what we analyzed in this work, i.e. unequal but deterministic file sizes. The analytical approach to this type of system was failed in the early stages due to the introduction of new dimensions to the state space of the system. However, we found it constructive to think about logical heuristic policies and to compare their results with our policy modified for the new case.

4.6.1 Random file sizes

We compared our policy, which we call NOP(Near-Optimal Policy), to six other policies in the first round of experiments. 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 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.

We performed our experiments with three different choices for the set of the average file sizes. These choices were based on the fact that, depending on the nature of the files broadcasted by this system, the distribution of the sizes can be different. For example, if the files are web pages, we may expect their distribution to have a heavy-tail behavior. This argument is more meaningful in our next set of experiments with deterministic file sizes, since in the random case only the average sizes are taken from these distributions and the actual file sizes are still according to geometric distributions. However, performing the experiments with these new distributions even in the random case allows us to have a better comparison between different policies in different situation.

The three distributions we tried to simulate by proper choice of the file sizes are uniform, Pareto and exponential distributions. For the uniform distribution we picked the average file sizes to be

$$l(i) = round\left(L + (i - N)\frac{L-1}{N-1}\right) \tag{4.18}$$

with $N = 50$ and $L = 50$. Here, N is the number of files in the system and L is the size of the largest file. The minimum size is fixed at 1. To simulate a truncated Pareto distribution with maximum value L and shape factor a, we used the following assignment

$$l(i) = round\left(\left[1 - \frac{i}{N}(1 - L^{-a})\right]^{-1/a}\right) \tag{4.19}$$

with $a = 1.8$ and the same values for L and N. Similarly an exponential distribution was modelled by the following file sizes

$$l(i) = round\left(1 - \frac{1}{a}\log\left[1 - \frac{(i-1)\left(1 - e^{a(1-L)}\right)}{N}\right]\right) \tag{4.20}$$

where $a = 25$ and $L = 50$.

In addition to the file size distribution, each policy was tested under 7 different total arrival rates namely, $5, 10, 20, 50, 100, 150, 200$. The upper limit is roughly high enough to represent the saturation region of the average delay, i.e., the region where the average delay does not change much with the increase in the work arrival rate. To summarize, for each policy, $3 \times 7 = 21$ results are available.

Candidate policies

Inspired by both intuition and the results of our previous work on the systems with equal file sizes, we picked six different policies in for our experiments. These policies are:

- **NOP**: The light traffic approximate indexing policy derived in the previous chapters.

- **PIP**: The original policy introduced in [54] extended for the new two dimensional setting where the index is defined by

$$\nu_i = \frac{x_i + c_y y_i}{\sqrt{\lambda_i}} \tag{4.21}$$

- **HP1**: Heuristic policy defined as

$$\nu_i = \frac{(x_i + c_y y_i) q_i}{\sqrt{\lambda_i}} \tag{4.22}$$

- **HP2**: Heuristic policy defined as

$$\nu_i = \frac{(x_i + c_y y_i) \sqrt{q_i}}{\sqrt{\lambda_i}} \tag{4.23}$$

- **HP3**: Heuristic policy defined as

$$\nu_i = \frac{(x_i + c_y y_i)}{q_i \sqrt{\lambda_i}} \tag{4.24}$$

- **HP4**: Heuristic policy defined as

$$\nu_i = \frac{(x_i + c_y y_i)}{\sqrt{q_i \lambda_i}} \tag{4.25}$$

- **HP5**: Heuristic policy defined as

$$\nu_i = \frac{(x_i^2 + y_i^2 + x_i y_i)}{\lambda_i} \tag{4.26}$$

- **MRF**: Maximum-Request-First index defined as

$$\nu_i = (x_i + c_y y_i) \tag{4.27}$$

where c_y is a weight parameter and needs to be tuned experimentally. Initially, we started with tuning c_y for all policies. However, it was found quickly that the PIP, HP3 and HP4 perform much worse than the other policies in all cases. Therefore, those policies were dropped from the list of the candidates for a near-optimal policy. Figures 4.6, 4.7 and 4.8 show the performance of each of the remaining policies for different choices of the c_y value for the pareto file size case. Based on the results, we conclude that HP2 and MRF policies are less sensitive to the c_y value, but

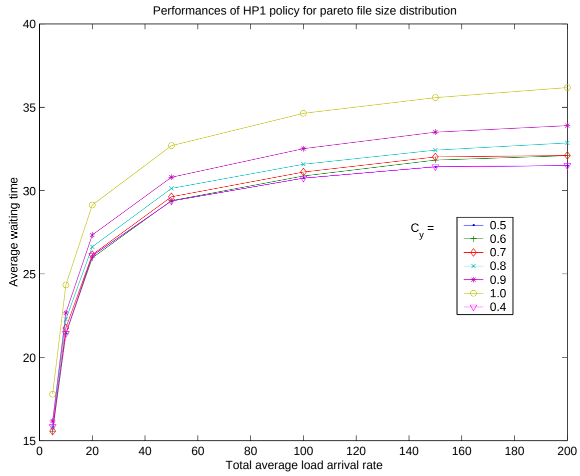

Figure 4.6: Tuning of the c_y parameter for HP1 policy.

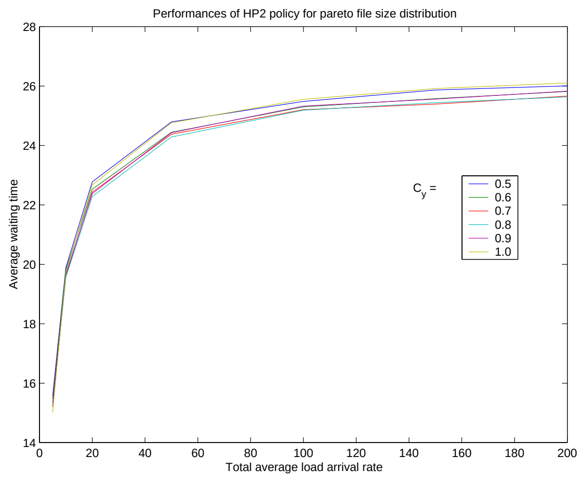

Figure 4.7: Tuning of the c_y parameter for HP2 policy.

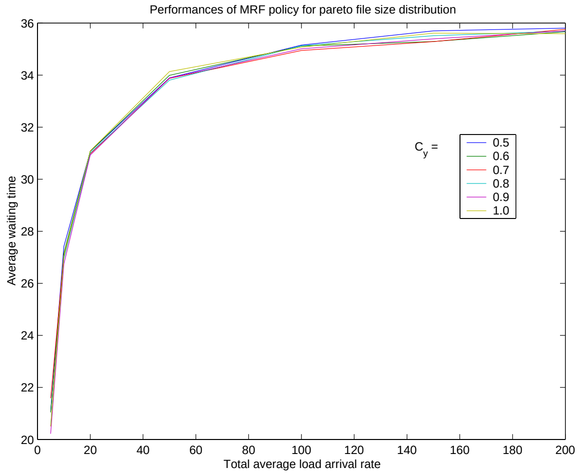

Figure 4.8: Tuning of the c_y parameter for MRF policy.

overall, they perform best with $c_y = 0.9$. However, HP1 is more sensitive to this coefficient and achieves its optimum performance with approximately $c_y = 0.5$. Our experiments with uniform and exponential distributions also gave the same results for the c_y values. Having each policy optimized, we can now compare the performances of the above policies with each other. Figure 4.9 shows the results obtained from each policy under different simulation settings. It is clear from the results that NOP and HP2 are the best policies with NOP slightly outperforming HP2 in most cases.

To summarize, we have found that the optimized versions of all candidate policies are inferior to NOP. According to the results, for practical purposes, the heuristic HP2 (with $c_y = 0.5$) can be used as a low complexity alternative for NOP.

4.6.2 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 sys-

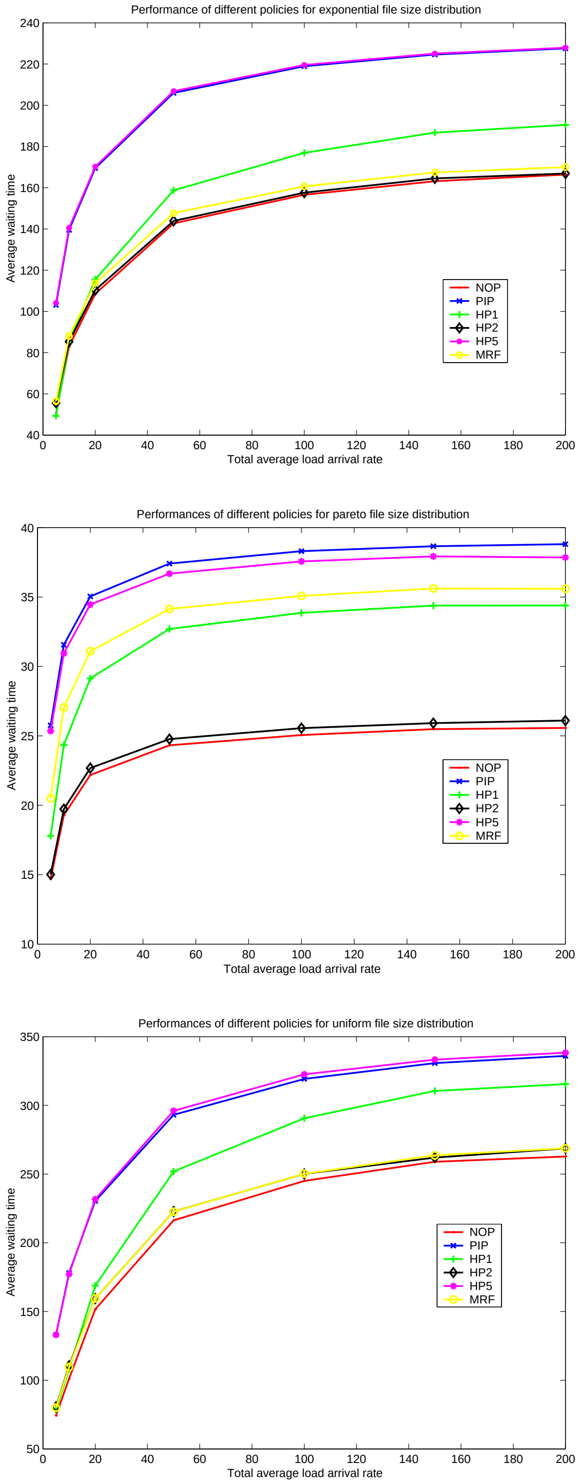

Figure 4.9: Comparing the performance of different scheduling policies for different choices of the file size distribution.

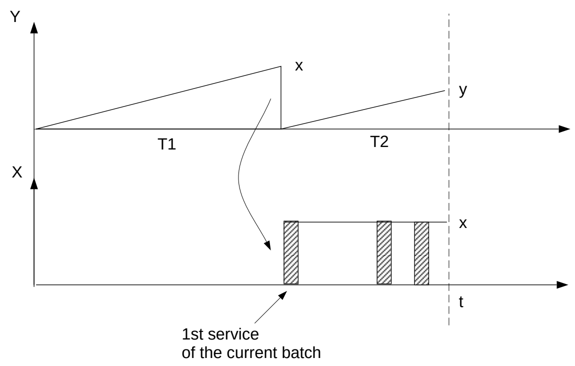

Figure 4.10: Evolution of the state variables by time for a single queue.

tems are examples of this type of systems. Unfortunately, the analytical approach to this problem proved to be too complicated and our efforts have not produced any sound results. However, in the absence of an optimal 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 and HP2 plus a new heuristic policy which we call LTWF1 which essentially approximates the "Longest-total-waiting-time-first" policy. 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. For LTWF1, the index is defined as

$$\nu = \frac{x^2 + y^2 + xy}{\lambda}. \tag{4.28}$$

Figure 4.10 explains how this index which is the sum of the areas of the two triangles and the rectangle in the diagram is derived. Since LWTF1 does not use the information about the number of already broadcasted segments of the file, another versions of it named LWTF2 was defined as

$$\nu = \frac{x^2 + y^2 + xy}{\lambda} - lx \tag{4.29}$$

where l is the number of already broadcast segments of the file. In fact, the LWTF2 index excludes the broadcast times of the first l segments from the total waiting time. The HP5 policy used in the random file size case is in fact the LWTF1 policy modified for the random case. We also tried to include the remaining waiting time of the customers in the index function by approximating it by $(L - l)y$ and also by $\frac{(L-l)y^2}{l\lambda}$ which basically estimates the remaining waiting time based on the time spent to serve the recent l segments of the file. The first modification only worsened the performance slightly. But the second modification gave completely unacceptable results compared to the other policies. Therefore, we don't discuss those two policies any more. 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 7 different arrival rates and three choices of the file size distribution. Since the file sizes are deterministic, assuming different distributions for those sizes have a better meaning in this case. Figure 4.11 shows the results. The first observation is the poor performance of the LTWF policies compared to the other policies. This can be an indication of this type of policy not being 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 although, as we also observed in last section, HP2 in some cases performed slightly better then NOP. We associate this difference to using the approximate light traffic version of NOP and expect this negligible difference to disappear if we enhance our approximation. To our surprise, MRF showed a highly variable performance. For uniform and exponential file sizes, it performed almost the same as HP2, i.e. ,

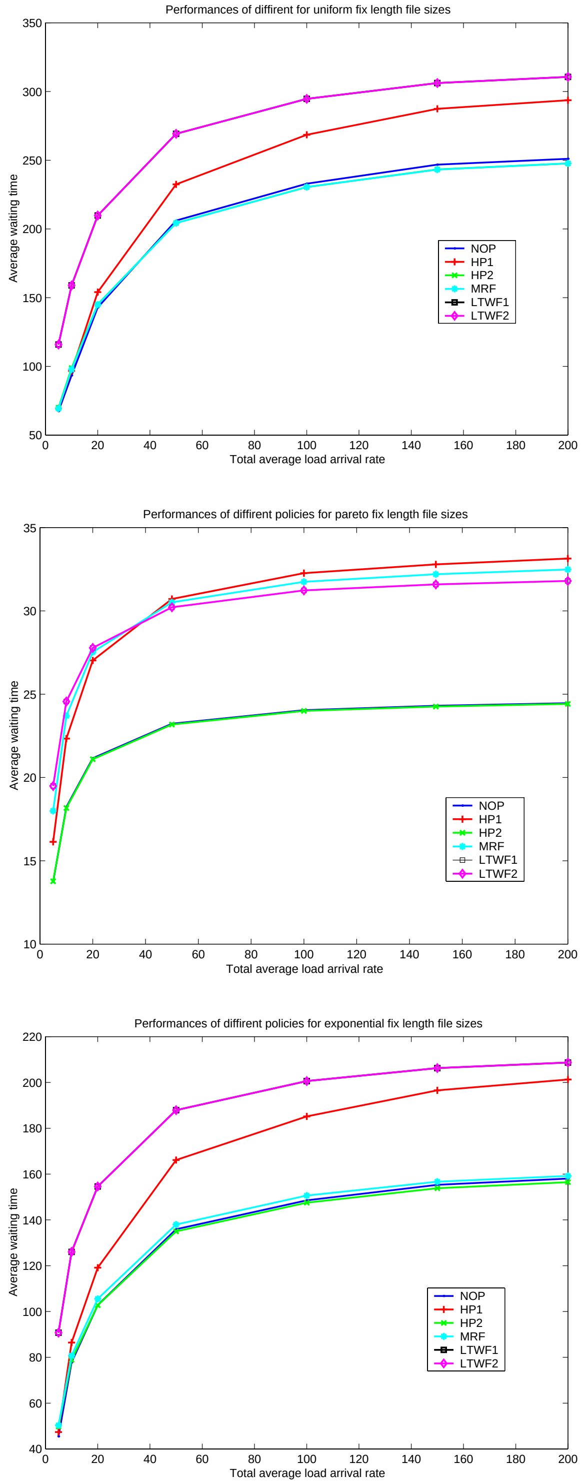

Figure 4.11: Performance of different policies for different choices of the file size distribution.

the best performance. But for pareto file sizes, it performed poorly and similar to LTWF and HP1 policies. However, as a general rule, we do not consider MRF to be as good as NOP or HP2. The poor performance of LTWF policies also suggests that having exact information about the number of remaining segments of the files does not necessarily enhances the performance. However, like many other observations, this is only a conjecture unless a formal proof shows its validity. Based on our work on the theoretical side of this version of the problem, we expect this not be a simple problem.

As a conclusion, we may infer that for practical purposes, HP2 can also be used as a low complexity policy for a wide range of file size distributions.

4.7 Conclusion and future work

The problem discussed in this chapter can be regarded as an extension of the original problem and includes that as a special case. We investigated the properties of the simpler single-queue problem and used the results to find an asymptotically optimal policy for the original problem. However, the file size is not the only direction for an extension to the original problem and other practical issues in some systems require changes in other initial settings of this problem. The abstract nature of the original problem allows for many different extensions to the problem to be introduced. Some other extensions can be introduced by changing the restriction on the arrival processes. In all the previous studies so far, we have assumed that the request arrival process to each queue is an i.i.d. Process which is also independent of all other properties of the system. Two immediate extensions are found when we relax either of the above restrictions i.e., the i.i.d. property and the independence of the arrivals and the state. In the following we take a closer look at these extensions and present a framework to take them into account in future research. A thorough analytical study of either of these new problems requires a step-by-step approach to carefully formulate and investigate them. However, we previously found that the core of the problem, when using the Restless Bandit approach, is the properties of the single queue system. Hence, we perform a qualitative study of the single-queue problem for each case and investigate the effect of each extension on the index function by comparing the numerical results by the associated results from the original single-queue problem.

4.7.1 Markovian arrivals

Previously, in our formulation, we assumed that the requests arrive in batches at integer time instants. The size of each batch was a random variable with $Poisson(\lambda)$ distribution and independent of the sizes of other batches. Here we like to consider the case where the batch size sequence is a Markov process i.e., the size of each batch depends on the sizes of the previous batches. To have a specific set up, we let the batch size distribution to remain Poisson but allow the sequence to have a memory of length one i.e., each batch size also depends on the value of the previous size. An appropriate model for this arrival sequence is the integer-valued Poisson AR(1) process. This type of process has been used in different studies to model the presence of memory in integer-valued random processes in some statistical modelling applications [38, 37] and we present a brief introduction of this model below.
Poisson AR(1) process The Poisson AR(1) process X_t; $t = 0, 1, \ldots$ is defined by the following difference equation

$$X_t = \alpha \star X_{t-1} + W_t; \tag{4.30}$$

where W_t is an iid Poisson process with rate $\lambda_w > 0$, $0 \leq \alpha \leq 1$ is a constant and, the thinning operator $\star$ defines the random variable $\alpha \star X$ as

$$\alpha \star X = \sum_{i=1}^{X} Y_i(\alpha) \tag{4.31}$$

where Y_i is a sequence of i.i.d., Bernoulli random variables with success probability α. In other words, $\alpha{\star}X$ is a Binomial random variable with maximum value X and success probability α. It is straightforward to verify that if X_{t-1} is Poisson with rate λ, then $\alpha{\star}X_{t-1}$ is also Poisson with rate $\alpha\lambda$. Also, if W_t is Poisson with rate λ_w and independent of the X values and X_0 is Poisson with rate $\lambda = \lambda_w/(1-\alpha)$, then $X_t;\ t = 0, 1, \ldots$ will be a stationary process with a Poisson marginal distribution with rate λ.

A difference between the Poisson AR(1) process and its continuous version i.e., the Gaussian AR(1), is the fact that there are two random components in the integer case. Some of the Properties of the stationary Poisson AR(1) process are listed below.

$$\begin{aligned}
E[X_t|X_{t-1}] &= \alpha X_{t-1} + \lambda_w && (4.32)\\
E[X_t] &= \frac{\lambda_w}{1-\alpha} && (4.33)\\
V[X_t|X_{t-1}] &= \alpha(1-\alpha)X_{t-1} + \lambda_w && (4.34)\\
V[X_t] &= \frac{\lambda_w}{1-\alpha} && (4.35)\\
\rho_k &= corr(X_t, X_{t-k}) = \alpha^k && (4.36)\\
P(X_t|X_{t-1}) &= \sum_{x=0}^{min(X_t,X_{t-1})} \begin{pmatrix} X_{t-1} \\ x \end{pmatrix} \alpha^x (1-\alpha)^{X_{t-1}-x} \frac{e^{-\lambda}\lambda^{X_t-x}}{(X_t-x)!}. && (4.37)
\end{aligned}$$

As we expect, the equation for the correlation function matches the correlation function of the continuous AR(1) process. in the following, we use this process to model the request arrival process to a broadcast queue and will try to study the implications of adding memory to the process. Obviously, by setting $\alpha = 0$, the arrival process becomes an i.i.d. Poisson process with rate λ_w.

Broadcast queues with Markov arrivals

In our previous studies of the different broadcast scheduling problems using the Restless Bandit approach, we found that the problem mainly reduces to the problem of controlling each queue individually and independent of the other queues. In our setting, serving the queue while containing x requests, generates a reward of x and we also incur a fixed cost ν. Leaving the queue idle neither generates a rewards nor involves any cost. However, because of the Markovian property of the arrival process, the state of the queue at each time instant is the number of requests in the queue together with the size of the latest arrival batch. If we denote by $V(x,i)$ the value function associated with the optimal policy for having x requests in the system and the most recent arrival of size i(included in x), then $V(.,.)$ satisfies the optimality equations

$$V(x,i) = -\nu + x + \beta \sum_{j=0}^{\infty} p(j|i) V(j,j) \tag{4.38}$$

for all states (x,i) in the active region and

$$V(x,i) = \beta \sum_{j=0}^{\infty} p(j|i) V(x+j,j) \tag{4.39}$$

for the states in the idle region. $p(j|i)$ is the conditional probability of having j arrivals at time t given that we had i arrivals at $t-1$, for $i,j \geq 0$. This probability is given by equation 4.37 above. A careful look at the above formulation shows that the state always remain in the $x \geq i$ region (except possibly at time 0). Therefore, only that area is of our concern in this study.

In the original problem, we found that the optimal policy for any given value of the service cost ν, is characterized by a border state $x(\nu)$ where it was optimal to

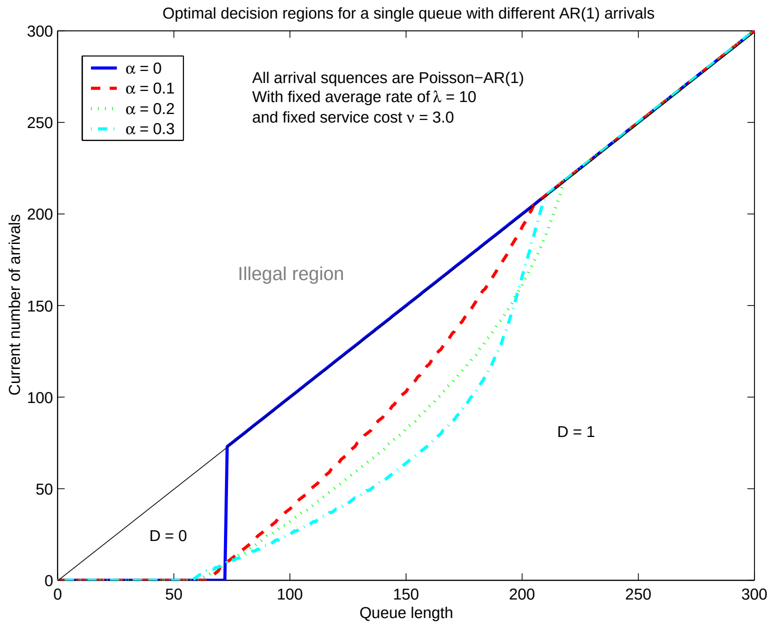

Figure 4.12: The effect of the correlation on the optimal policy.

leave the queue idle for all states less than $x(\nu)$ and serve the queue otherwise. In the current system, the state is a point on the $x - i$ plane so the optimal policy is characterized by a curve that separates the idle and active regions. The *strength* of the memory of the system can be controlled by the value of α. $\alpha = 0$ results in a memoryless system while $\alpha = 1$ results in a constant sequence i.e., unit correlation function. Our first set of experiments aim to investigate the effect of the memory on the index function. To this end, we compute the optimal policy for several systems with only their α values being different. In all experiments we have used the Value Iteration method for finding the optimal policy. Figure 4.12 shows the results for a queue with fixed arrival rate $\lambda = 10$, $\beta = 0.99$ and service cost $\nu = 3.0$ with four different values of α. The rate of the innovation process W_t in each experiment is changed to always result in a total arrival rate of 10 requests per time unit. The $\alpha = 0$ case is the original memoryless problem with i being augmented as the new state variable. Obviously, in this case, the threshold state will remain the same for all i values as it can be clearly seen in the figure. The value of the threshold state is around $x = 73$ which is also what our recursive method for the original problem would calculate. For $\alpha = 0.1 (\lambda_w = 9)$, the threshold state depends on the value of i and takes smaller values, compared to the previous case, for small "i"s and larger values for larger "i"s. This behavior can be explained by the variations in the next "instantaneous" arrival pdf due to the current arrival size i. It can be shown that $(J|I = i)$ is a Poisson random variable with rate $\alpha i + \lambda_w$. For $i = 10$, the next arrival distribution has the same rate of 10. Therefore, the threshold state remains the same (intersection of $\alpha = 0$ and $\alpha = 0.1$ curves in the figure). However, since the threshold state is an increasing function of the arrival rate, it moves toward larger values as i becomes larger than 10 and toward smaller values as it becomes smaller than 10. The other two curves show the same behavior for $\alpha = 0.2$ and $\alpha = 0.3$ cases.

The second set of experiments investigates the dependence of the optimal decision regions on the value of ν. In these experiments, we fixed α at 0.1 and computed the optimal policy for four different values of ν. Figure 4.13 shows the results and as we expect, the edge of the idling region is pushed toward larger x values as ν increases. Although not serving as a proof, the figure nevertheless shows the monotonic increase of the idling region with increasing ν values.

As a conclusion of the above experiments, and from the viewpoint of guessing the value of the index associated with every state, we can compare the index with the case with $\alpha = 0$. For any x, the value of the index associated with state (x, i) is larger than the corresponding $\nu(x)$ value if $i < \lambda$, smaller if $i > \lambda$ and equal if $i = \lambda$. The above deviations from the standard case become larger as α increases.

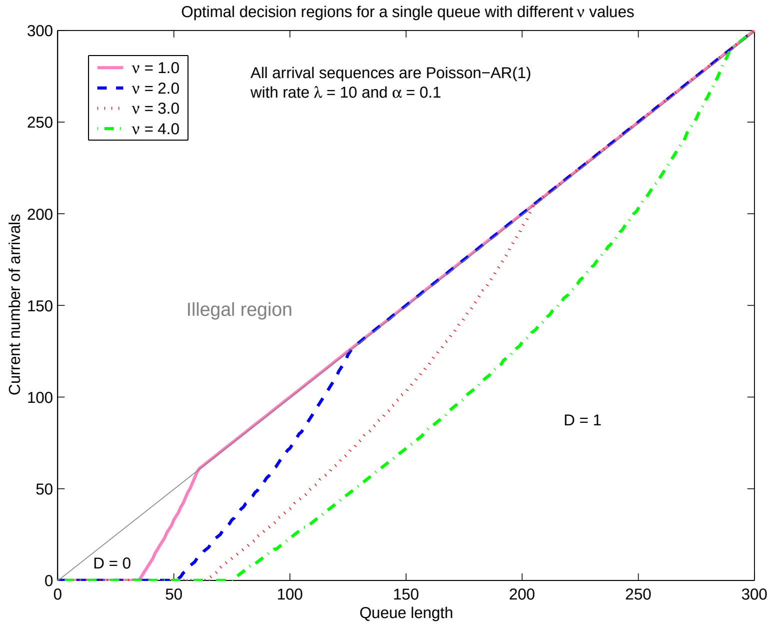

Figure 4.13: The effect of service cost on the optimal policy.

4.7.2 Systems with transmission errors

Another direction to extend the original problem is to allow for transmission errors in the system. After every broadcast, a fraction of the receivers fail to receive the file completely and therefore send another request for that file to the system. In our modelling of this effect, we assume a simple bernoulli model for the failure of each customer in receiving a broadcasted file. The failure probability is shown by q and we assume that the retransmission requests are received by the system before the next decision instant. If we show the value function of the optimal policy by $V(x);\ x = 0, 1, \ldots,$ it satisfies

$$V(x) = -\nu + (1-q)x + \beta \sum_{i=0}^{\infty} p(i|x)V(i) \qquad (4.40)$$

for the active region and

$$V(x) = \beta \sum_{i=0}^{\infty} p(i|0)V(x+i) \qquad (4.41)$$

for the idle region. Here $p(i|x)$ is the probability of going to state i after the broadcast of a file for x clients. This probability is composed of two parts, a Poisson random variable with rate λ and a Binomial distribution with success probability q and maximum value x. In other words

$$p(i|x) = \sum_{j=0}^{min(i,x)} \left(\begin{array}{c} x \\ j \end{array} \right) q^j (1-q)^{i-j} \frac{e^{-\lambda}\lambda^{i-j}}{(i-j)!}. \tag{4.42}$$

The similarity between the transition probabilities for this case and the previous case is obvious. In fact, this model captures another form of dependence in the arrival process where the arrivals also depend on the size of the recent serviced batch. We remind that a value of $q = 0$ results in the original system with no errors.

In order to investigate the effect of the q parameter on the index function, we calculate the index function for many values of x and for three choices for q i.e., $q = 0$(the classic case), $q = 0.05$ and $q = 0.1$. We used the Value Iteration method for each x and q value to find the corresponding ν. Figure 4.14 shows the results and it can be easily seen that the index function increases with increasing q. This effect is expected since a larger q results in a larger contribution to the base arrival rate λ from the retransmit requests and a larger total rate always results in a larger index function.

We end this section by reminding that both of the above situations need more detailed analyses and, when solved, will give useful results for practical systems.

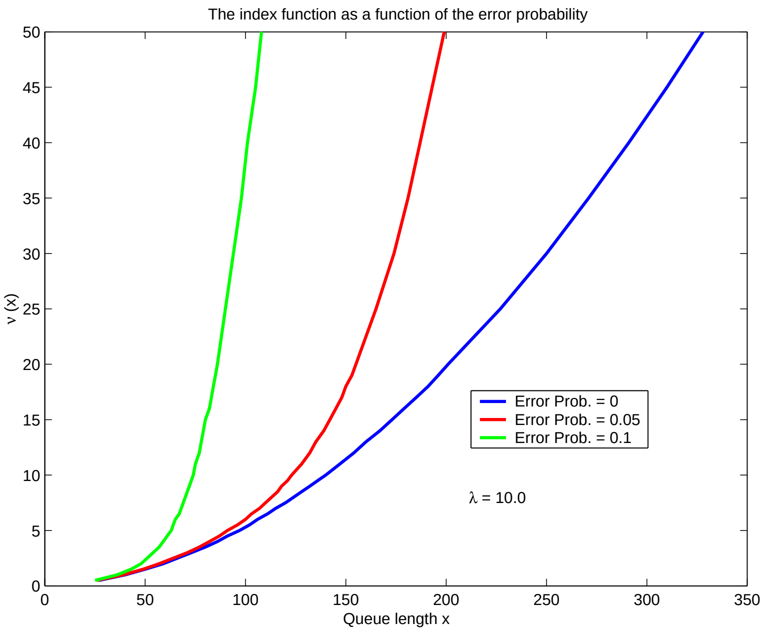

Figure 4.14: The effect of error probability on the index function.

Chapter 5

Analysis of the Internet traffic

5.1 Introduction

This chapter highlights the main results of a traffic analysis project that was performed on a satellite data delivery system. This topic, although different in subject, is related to the previous topic of broadcast scheduling in the sense that they both deal with the different aspects of the same system.

The goal of this project was to extract realistic models for the statistical behavior of the various components of the Internet traffic flowing through the hybrid gateway of the DirecPC system. Having correct models for the statistical properties of the traffic is an essential part of any work on performance evaluation of the system, either through simulations studies or by analytical methods. There has been a lot of research work in the recent years on this topic and interesting observations have been reported in some papers. As examples, we can refer to [46] as a detailed empirical modeling of the TCP traffic and [56] as one of the first modeling experiments that indicated the failure of the Poisson model for certain aspects of the traffic. Later, other works ([29, 11]) revealed that the reason for this deviation and specifically, the long-range dependence (LRD) property, is because of the heavy-tailed distribution of the file sizes transferred by the network. Among the other works on traffic modeling, we can name the work by Paxon and Floyd [56] where they derived the empirical distributions of the various components of the traffic like file sizes, connection durations etc. and also [60, 1, 51] where the qualitative behavior of the traffic is studied. A review of the different works on traffic modeling shows that although certain properties of the traffic like the tail distribution of the file sizes and its resulting LRD property in the aggregate traffic, do not change from time to time and from network to network, other more detailed properties like the best fit to some distributions may change by time and by the network. In this project we study the traffic flowing through a hybrid network i.e., the DirecPC network and try to find if our findings are inline with the previous findings for other networks. Here, in Section 5.2 a brief review of the system and our measurements is presented. Section 5.3 is dedicated to the methods that were used in our analyses. Finally, in section 5.4, we present and discuss the results of the experiments.

5.2 DirecPC environment

The DirecPC system is a hybrid Internet delivery system that allows its users to have a high download bandwidth by using a satellite link. Users or, in the DirecPC terminology, the Hybrid Hosts(HH) initiate every transaction by sending a request in the form of an IP packet with its own IP address as the source and the Internet Host(IH) address as the destination address. This packet is sent by the user using one of the different access methods (e.g. dial-up service, direct satellite access) to the HGW. After the arrival of the packet to the HGW, it is sent to the actual

Internet Host. The IH sends the requested packets to the IP address of the HH but all the packets destined to any HH address pass through the HGW. As soon as the HGW realizes that the packet is to be received by an HH, it sends the packet to the satellite link via the queuing system in the gateway. Once sent to the satellite link, the HH receives the packet using its satellite receiver. One of the components of the DirecPC system is its queuing system. More users with better quality of service requirements can be served with an efficient queuing system and a crucial element for designing a good queuing system is a good understanding of the incoming traffic to the system and its statistical properties.

In this project we have tried to fit several statistical models to different aspects of the traffic to find a reasonable model that captures those characteristics in the best way. The traffic we were interested on is the traffic that flows through the satellite link i.e. the IH to HH traffic. The traffic is collected from the LAN at the HGW that contains three types of traffic : traffic from HH to HGW, traffic from HGW to IH and traffic from IH to HH that is of our interest. We used the *tcpdump* program to record the LAN traffic (all the traffic) and used are own *awk* scripts and *C* programs to extract the IH to HH traffic from it and also to further extract the traffic related to different protocols from the trace. The other components of the traffic are also subject of other projects but in the rest of this report by "traffic" we mean the IH to HH traffic.

5.3 Methods

Depending on the statistical properties of the different components of the traffic, different models can be fitted to them. For quantities like the connection size usually the samples are independent and identically distributed and we are mostly interested in fitting a well-known probability distribution to them. However, when the process of interest is not i.i.d., its correlation structure is also of interest. Packet and connection interarrival times are examples of correlated processes as we will see later in this chapter. Analysis of each data set involves certain steps that we will discuss in the next section.

5.3.1 Initial tests

First, we need to ensure that the data is stationary i.e., its statistical characteristics remain constant in time. Although there is no clear distinction between stationarity and long-range-dependence, various methods are available to test the stationarity of a sequence using hypothesis testing. We used a simple method defined in [5]. The sequence is first divided into N pieces and the mean-square value of each piece is computed. If the sequence is stationary, the sequence of the mean-square values should be an i.i.d. sequence. We then use a simple independence test called the Reverse Arrangement test [5] to test the confidence level of the hypothesis which states that the mean-square values are independent.

In practice, we found that for most cases, a sequence of approximately 20 minutes length passes the stationarity test with a %95 confidence interval, which is intuitively correct, considering the relatively larger time constant of the hourly trends in the traffic.

After finding a stationary section of every sequence, the first test is to find about its autocorrelation properties. Here, we make a visual judgment using the Autocorrelation Function(ACF) of each time series defined as

$$ACF(k) = \frac{cov(X(t), X(t-k))}{cov(X(t), X(t))}; \;\; k = 0, 1, \ldots \tag{5.1}$$

where $X(t)$ is the original time series and $cov(.,.)$ represents the covariance function. When the values of the ACF at $k > 0$ points are reasonably inside the %95 confidence interval, we treat the series as a non-correlated series. Otherwise, we will seek more information about its correlation structure if it shows any long-range-dependence(LRD) properties.

5.3.2 Probability models

We used four well-known probability distributions as candidates to fit to each of the i.i.d. data sets. These distributions are the exponential distribution with pdf

$$f_X(x) = \lambda e^{-\lambda x} \qquad x > 0;\ \ \lambda > 0, \tag{5.2}$$

Lognormal distribution with pdf

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} x^{-1} e^{\frac{(\log x - \mu)^2}{2\sigma^2}};\ \ x > 0, \tag{5.3}$$

Weibull distribution with

$$f_X(x) = \left(\frac{\beta}{\alpha}\right)\left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x}{\alpha}\right)^{\beta}} \qquad \alpha, \beta > 0 \quad x > 0. \tag{5.4}$$

and Pareto distribution with

$$f_X(x) = \frac{ak^a}{x^{a+1}} \qquad a, k > 0 \quad x > k. \tag{5.5}$$

The choice of the exponential distribution is based on the fact that many aggregate processes approach a Poisson process in the limit. The Weibull and Lognormal distributions have been used in other traffic modeling projects to fit the data, and the Pareto distribution is chosen to capture any possible heavy-tail behavior in the data sets.

5.3.3 Estimators and fitting tests

In order to fit the data to each of the above distributions, the Maximum Likelihood estimators [28] for each distribution were used. For a sample data set $X = x_1, x_2, \ldots, x_N$, these estimators are:

$$\widehat{\lambda} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.6}$$

for the exponential distribution,

$$\widehat{\lambda} = \frac{1}{N}\sum_{i=1}^{N} \log x_i \tag{5.7}$$

and

$$\widehat{\sigma^2} = \frac{1}{N}\sum_{i=1}^{N} \left(\log x_i - \mu^2\right)^2 \tag{5.8}$$

for the lognormal distribution,

$$\widehat{\alpha} = \left[\frac{1}{N}\sum_{i=1}^{N} t_i^{\widehat{\beta}}\right]^{\frac{1}{\widehat{\beta}}} \tag{5.9}$$

and

$$\widehat{\beta} = \left[\left(\sum_{i=1}^{N} t_i^{\widehat{\beta}} \log x_i\right)\left(\sum_{i=1}^{N} x_i^{\widehat{\beta}}\right)^{-1} - \sum_{i=1}^{N} \log x_i\right]^{-1} \tag{5.10}$$

for the Weibull distribution and finally

$$\widehat{k} = min x_i \tag{5.11}$$

and

$$\widehat{a} = n\left[\sum_{i=1}^{N} \log(x_i/\widehat{k})\right]^{-1} \tag{5.12}$$

for the Pareto distribution.

In the cases where it is desired to investigate more about the autocorrelation of the data sets, specially if the goal is to detect the LRD properties, we use the wavelet estimation method introduced in [1]. Briefly, this method uses the fact that ([2])

if X is a self-similar process with Hurst parameter $H \in (0.5, 1)$, the expectation of the energy E_j within a bandwidth 2^{-j} around frequency $2^{-j}\lambda_0$ is

$$\mathbf{E}[E_j] = E\left[\frac{1}{N_j}\sum_k |d_{j,k}|^2\right] = c|2^{-j}\lambda|^{1-2H} \tag{5.13}$$

where c is a constant independent of j and $d_{j,k}$; $k = 1, \ldots, N_j$ are the wavelet coefficients at scale j and time $2^j k$ of the *discrete wavelet transform* of X. Based on this observation, a plot of $\log_2 \mathbf{E}[E_j]$ versus j will be linear with slope $2H - 1$ and we use this method to detect the self-similar property and if so, to find the Hurst parameter. In our experiments we counted the number of packets arrived in successive $10ms$ intervals and used that discrete-time sequence for the wavelet test. We also used the Whittle estimator [6] for estimation of the H parameter for comparison purposes.

Finally, in order to compare the fits given by the above four distribution to a given data set, we use a non-parametric discrepancy test introduced in [47]. This test, called the λ^2 test, is a modified χ^2 test which is known to be less sensitive to the number and width of the bins. It should be mentioned at this point that none of our data sets really matches any of the above distributions in a statistical sense i.e., within a reasonable confidence interval. We are only looking for models that reasonably capture the shape of the empirical distributions.

5.4 Results

In this section we discuss some of the results of our experiments on the collected traffic trace. The trace was 7396 seconds long and contained 12696587 TCP and 644814 UDP packets. A total of 1352 different users were active for some period during the measurement time. The TCP and UDP packets carried 10572270728

Protocol	#packets	%packets	#bytes	%bytes
TCP	12,696,587	%92.8	10,572,270,728	%98
UDP	644,814	%4.7	151,229,464	%1.4
Other	340,656	%2.5	8266712	%0.6

Table 5.1: share of different protocols in the traffic.

and 151229464 bytes of data, respectively, that results in an approximate rate of 1449905 bytes per second. Table 5.1 shows the shares of the two major protocols in the traffic. As we can see in the traffic, the TCP is by far the most frequent protocol that makes about %98 of the traffic. Because of this result, we further concentrated our analysis on the TCP traffic. The TCP traffic was analyzed to model the packet arrival as well as the connection arrival processes. A second analysis gave the composition of the TCP traffic in terms of the shares of different protocols in it. The TCP traffic is composed of the traffic generated by all the applications using this protocol as their transport protocol. A simple analysis of the traffic shows that the HTTP,FTP and NNTP are the three most popular components of the TCP traffic and other applications like POP3 and SMTP are in the next rows. Since HTTP traffic is by far the largest component of the traffic ($\%50 - \%70$), here we mostly focus our attention on this traffic and do not present the results related to other protocols and applications.

5.4.1 TCP Packet Arrival

The arrival process for the TCP packets is a combination of the arrival pattern of the many protocols using TCP as their transport protocol. A stationary test on the aggregated packet arrival process in 20s intervals shows a high level of stationarity

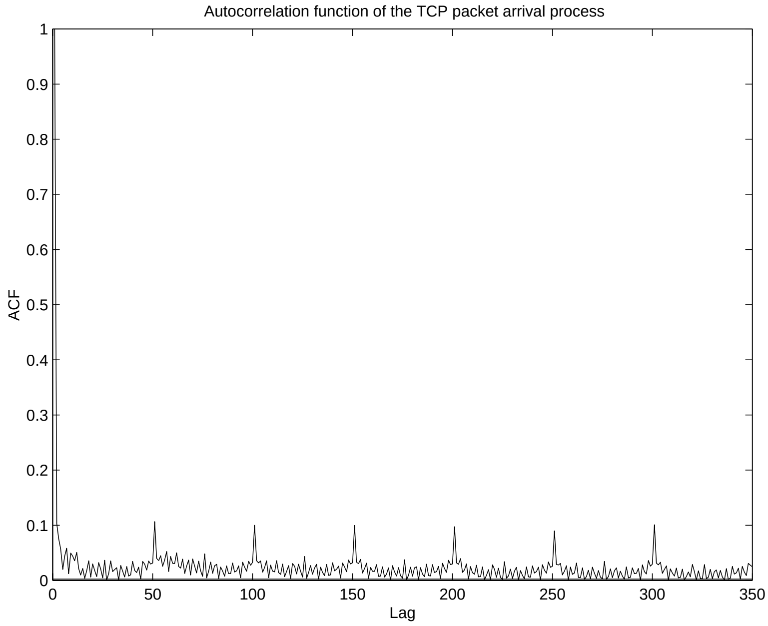

Figure 5.1: Autocorrelation function of the aggregated TCP packet arrival process.

(%90) in the part of the trace between 2500-5000 seconds. The autocorrelation function for this part of the trace is shown in figure 5.1. The first observation is that a strong periodic component with an approximate period of 0.5 seconds exists in the traffic which causes the strong periodic spikes in the autocorrelation function. detailed analysis of the traces showed that this component is caused by periodic consumption of part of the bandwidth by other services that share the bandwidth with the Internet traffic in the hybrid gateway. However, if we ignore this effect and focus on the values of the autocorrelation function at other points it shows a non-negligible tail for the correlation function. Therefore we use the wavelet tool to investigate the scaling properties of this process. Figure 5.2 shows an almost linear scaling behavior from scales 7 to 13 i.e. scales of 1 to 80 seconds. If we accept that level of linearity for the trace, the slope of the line fitted to that part gives a Hurst parameter of $H = 0.84$. The Whittle estimator with the

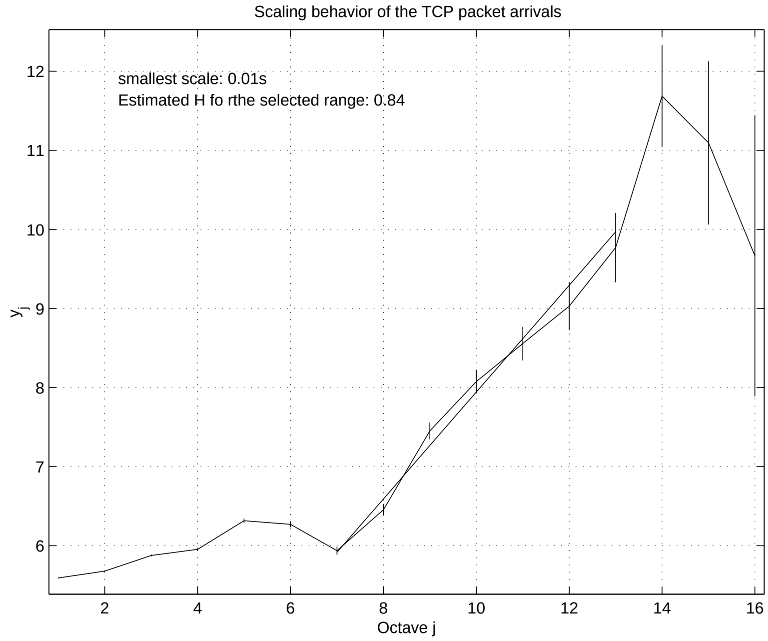

Figure 5.2: Global scaling behavior of the aggregated TCP packet arrival process.

assumption of a FGN model for the aggregated process in 1 sec intervals gives also an H value of 0.844 with the %95 confidence interval of [0.828 0.859] which matches our previously estimated value. These results suggest that the aggregated TCP packet arrival process is an LRD process with the above Hurst parameter. This observation is in line with many previous reports (e.g. [60, 1, 45]) where the LRD behavior has been observed in different data sets. As we see in the graph, the behavior of the process is very different in smaller scales. Specifically, the peak around scales 5 and 6 is probably caused by the periodic component around those time scales. In the following sections, we will try to investigate more about the properties of the TCP traffic. However, since the TCP traffic is composed of many different protocols, we only focus on its major component i.e., the HTTP traffic so that we can talk about a unified model for this component. Otherwise, due to the different nature of different applications, it would be unreasonable to propose

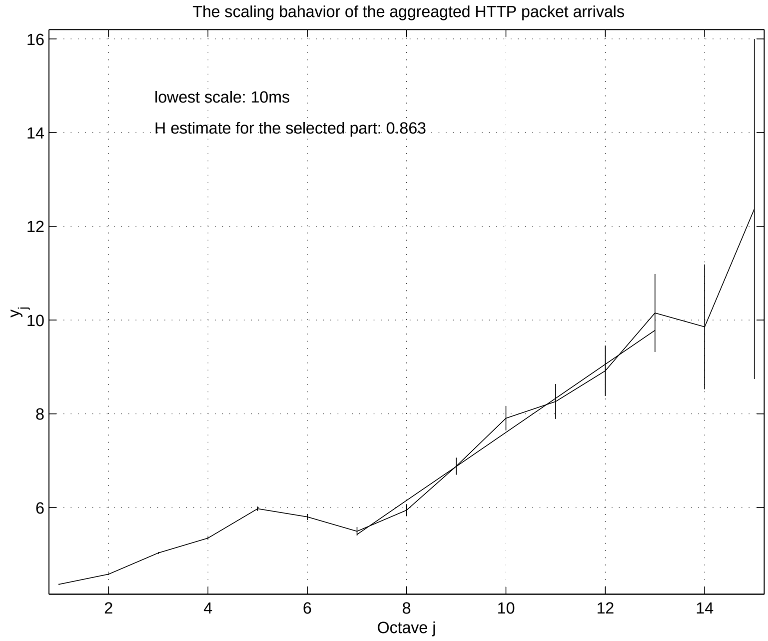

Figure 5.3: The scaling behavior of the aggregated HTTP packet arrival process.

such overall models.

5.4.2 HTTP traffic

As we expect (figure 5.3), the scaling behavior of the HTTP packet arrival process is very similar to that of the parent TCP traffic and shows a self-similar behavior with Hurst parameter 0.84 over the 5 to 13 scaling range or equivalently between $0.32s$ and $82s$ time scales. Since our traffic trace is a collection of a large number of different client-server connections, it is natural to focus on the behavior of a typical connection during our search for a reason for the above self-similar property. This is mainly motivated by a famous paper by Taqqu *et. al.* [17] where they prove that the superposition of a large number of independent alternating (ON-OFF) renewal processes, each with independent ON and OFF durations, and at least one of the

ON or OFF durations having a heavy-tail distribution with tail slope a, results in a fractional Brownian motion process with Hurst parameter $H = (3 - a)/2$. The ON-OFF property is a natural feature of the HTTP traffic and therefore it is reasonable to try to relate the self-similar property of the aggregate to the individual client-server connections.

The HTTP traffic between a client-server pair usually consists of of successive periods of silence (OFF) and file transfer (ON). Each file transfer period starts when the user requests a file from the server by clicking on a link and the silence or the *think time* period starts after the end of the previous file transfer and is continued until the user requests a new file. In order to find the distribution of the ON and OFF durations in our traffic, we extracted all distinct client-server communications from the trace and measured the ON and OFF periods in each of them by treating all the connections with their start times less than 0.5 seconds apart as part of the same ON periods and, the end of the longest connection as the end of that ON period. After finding all the available ON and OFF periods from the HTTP trace, we constructed two separate data sets of ON and OFF durations and investigated the distribution of each set separately. A visual ACF test on each set showed a reasonable independence structure and figure 5.4 shows the complementary empirical distribution of each set in the log-log scale. It can be seen from these figures that the OFF periods do not show any heavy-tail behavior however, the ON durations drop, for large values, with an approximate slope of -1.2. This construction then suggests a Hurst parameter of $H = 0.9$ for the packet arrival process which is not very far from our direct estimation of this parameter (0.84).

The ON periods are in fact the durations of the TCP connections that carry the

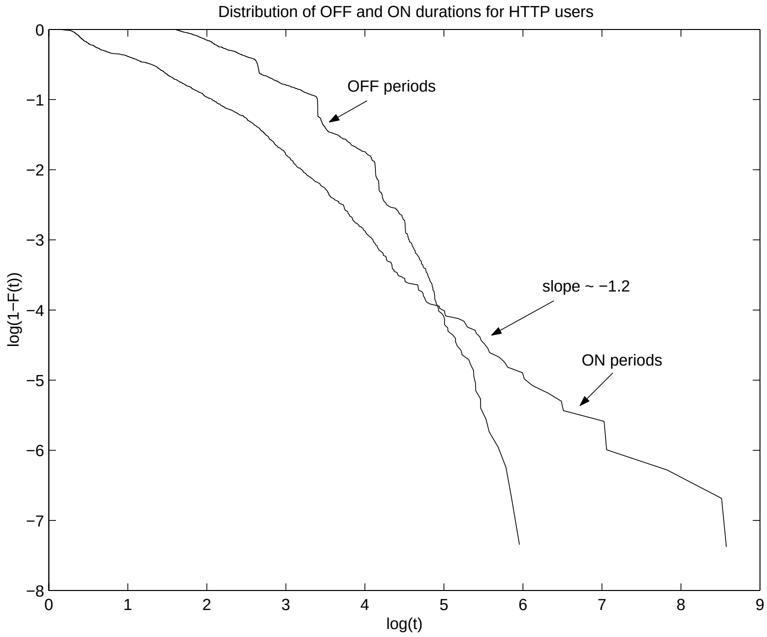

Figure 5.4: Distribution of the Length of the ON and OFF times for HTTP traffic.

different segments of the web pages to the clients. Therefore, we can expect to see a close relationship between the distribution of the ON durations and the file size distributions, at least with regard to the tail behavior which is the main cause of the self-similar property.

5.4.3 HTTP file size distribution

We measured the size of the files transferred by the HTTP protocol by subtracting the final and initial segment number for each connection. However, to avoid the errors caused by the censor effect in our finite-time data, we only used the connections which started during the first hour of the trace to make sure that all the connections are taken into account. Figure 5.5 shows the empirical distribution of this data set along with the best fits by the four candidate distributions for

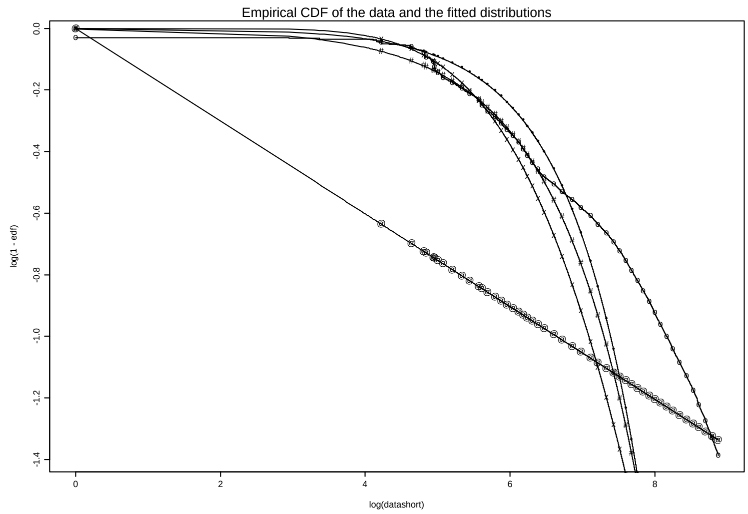

Figure 5.5: The CDF of the sizes of the HTTP connections and the fitted distributions to the lower %65 of the data.

fitting. It is clear from the figure that the file size distribution has basically two different behaviors. In fact, the shown distributions are the best fits to the lower %65 section of the distribution. Table 5.2 shows the result of the fitting based on the λ^2 discrepancy test explained before. Here λ is a measure of discrepancy and $\sigma(\lambda)$ is its variance. According to this table, the Weibull distribution results in the best fit to this data and the Pareto distribution is the worst of them. However, the situation is reversed for the upper %35 of the data. Figure 5.6 shows that the Pareto distribution can capture the heavy-tail behavior of the upper portion of the data much better than the other distribution. The estimated shape factor for the Pareto fitting is approximately 1.1 which results in a slightly heavier tail than the tail of the distribution of the ON times. However, we still consider this result as a confirmation of the fact that the self-similar nature of the HTTP traffic is largely caused by the heavy-tail distribution of the file sizes [11]. In fact, in [11], they

Distribution	$\lambda - \sigma(\lambda)$	$\lambda + \sigma(\lambda)$
Exponential	0.2270	0.2272
Lognormal	0.2130	0.2132
Weibull	0.1232	0.1232
Pareto	2.5231	2.5243

Table 5.2: Results of the discrepancy test on the fitted distributions to the lower %65 of the HTTP connection sizes.

estimated a slope of -1.2 for the tail of the file size distribution while estimating a Hurst parameter of 0.7 to 0.8. So our results show more consistency between the file size and the Hurst parameter.
Although our experiments and results also include detailed studies about the statistical properties of the FTP and NNTP protocols, we do not include the results here and only mention that the same steps were taken to have a reliable analysis of each data set. Our results showed that the FTP connections arrival process has a marginal distribution very close to an exponential distribution (figure 5.7) and although the autocorrelation function for lags larger than 0 is not completely inside the %95 confidence interval, it may be still considered as an independent sequence resulting in a Poisson model for the FTP connection arrival process. Also, the duration of the FTP connections has a heavy tail as can be seen in figure 5.8. Therefore, it seems that an $M/G/\infty$ model [10, 34] can be a proper model for the aggregate FTP traffic.

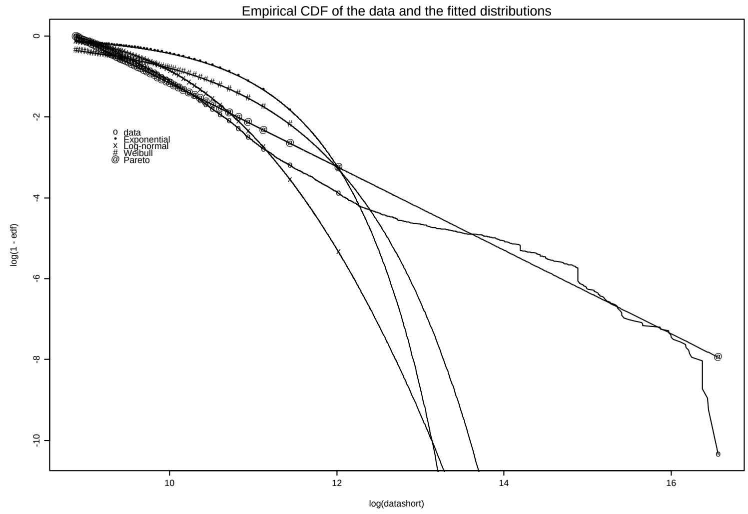

Figure 5.6: The CDF of the sizes of the HTTP connections and the fitted distributions to the upper %35 of the data.

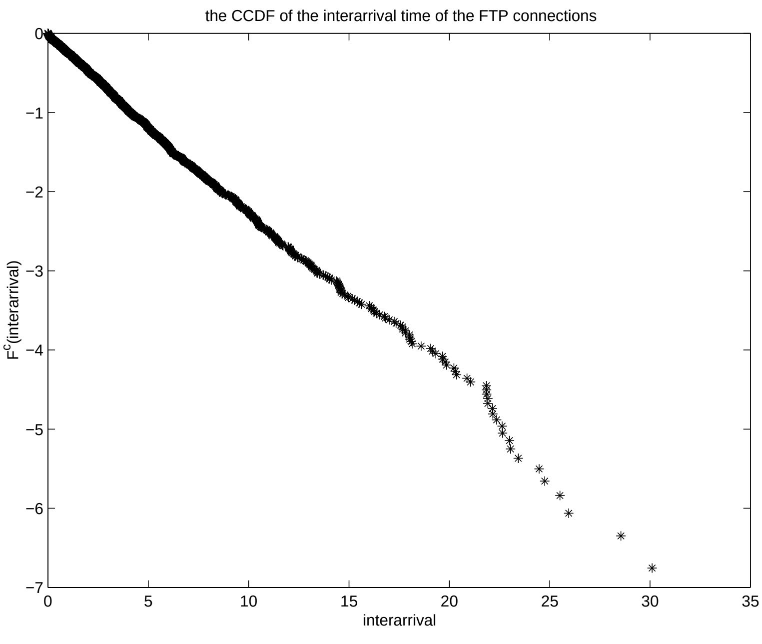

Figure 5.7: Distribution of the interarrival times of the FTP connections.

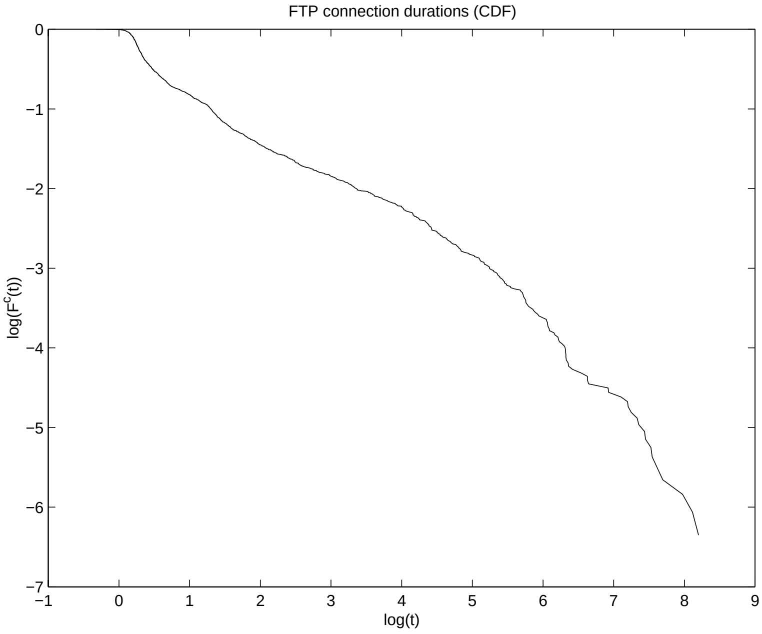

Figure 5.8: Distribution of the durations of the FTP connections.

5.5 Implications of the HTTP model, future work

A major implication of the ON-OFF model for the HTTP connections is in the resource allocation applications. This issue is studied in [13]. They construct a Markov chain for a system where a fixed bandwidth is shared by N HTTP users. In that system all users have the same ON-OFF behavior where the OFF periods are generated by sampling the *think time* distribution and the ON periods for each user start when the user begins to transfer a file. All file sizes have the same distribution. Obviously, in this system, the OFF durations are independent. However, since the ON durations depend on the file sizes as well as the bandwidth for every user, they will depend on the number of users in their ON state. Let's assume that the OFF durations have an exponential distribution with mean $1/\lambda$ and the file sizes have an exponential distribution with mean $1/\mu$. In that case, the number of ON users

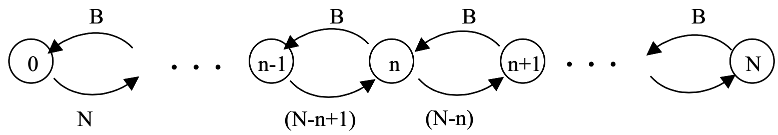

Figure 5.9: Markov chain representation of a single-class processor sharing system.

constitutes a Markov chain as shown in figure 5.9 where B is the total bandwidth. In this system, the distribution of the bandwidth received by each user can be found by calculating the stationary distribution of the Markov chain. However, the assumption of exponential distribution for the file sizes prevents this model to be practical in real situation. Heyman et.al. [13] showed that the stationary distribution of this chain is insensitive to the file size distribution, i.e. as long as the file size distribution has a mean of $1/\mu$, the stationary distribution remains the same no matter what the actual shape of the distribution is.

A very useful extension of this model is the case where the users do not all get the same share of the bandwidth. Instead, users are divided into classes and the bandwidth is divided according to some weighted processor sharing fashion. More specifically, suppose that there are N_1 users of class 1 and N_2 users of class 2 in the system and weights w_1 and w_2 are assigned to classes 1 and 2, respectively. In a typical state with x_1 users of class 1 and x_2 users of class 2 in their ON state, the bandwidth received by each class 1 user is

$$\frac{w_1 B}{x_1 w_1 + x_2 w_2} \tag{5.14}$$

and the bandwidth for each class 2 user is

$$\frac{w_2 B}{x_1 w_1 + x_2 w_2}. \tag{5.15}$$

If we assume that the file size and think time distributions are both exponential

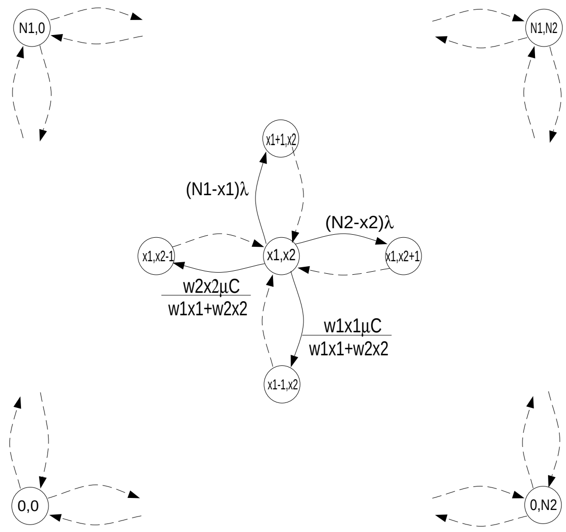

Figure 5.10: Markov chain representation of a two-class weighted processor sharing system.

as before, the system can be represented by a two-dimensional Markov chain as in figure 5.10.

A number of initial simulation studies on this system suggested that the insensitivity property holds here as well. A first look at this chain also showed that it has one of the necessary conditions for insensitivity i.e.,the *instantaneous attention* property. This property requires the system to start processing a job as soon as it arrives into the system and obviously, a processor sharing system satisfies this requirement. With that assumption, we were looking for proper well-known numerical and analytical methods (e.g. Matrix geometirc methods [35, 42], Diffusion approximation [49, 25], ...) to find the stationary distribution of this chain.

However, a more thorough study of the insensitivity phenomena and the related works and particularly [52],[53] and [24] showed that another necessary condition for insensitivity i.e. the *job local balance* property does not always hold for this

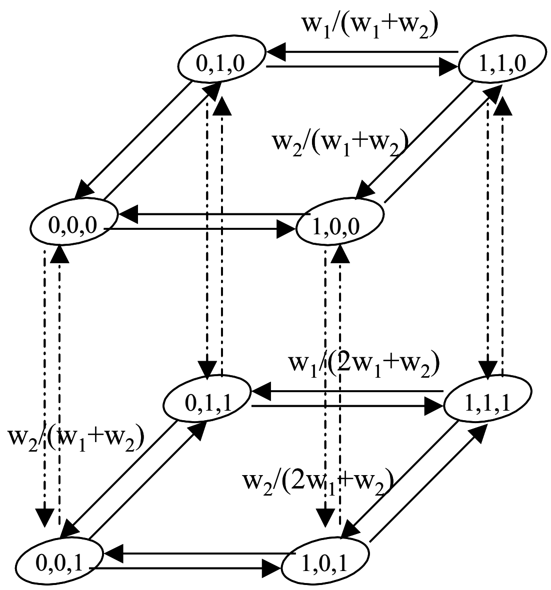

Figure 5.11: A two-class system to investigate the local balance property.

Markov chain. In our system, the job local balance property with respect to a specific user s_0 requires that

For all states g that contain user s_0 *in its ON state, the flow out of g due to ON to OFF transitions of* s_0 *equals the flow into g due to OFF to ON transitions of* s_0.

This fact can be studied by constructing a simple 2×1 chain which distinguishes between the users of the same class (figure 5.11) and finding the flows generated by the departure or arrival of that user. It can be shown that these flows do not necessarily satisfy the job local balance property and therefore the chain is not insensitive.

In the absence of the insensitivity property, the analysis of the system becomes a difficult task that deserves a separate study. Therefore, we did not continue this path for this project. however, due to the practical importance of this issue, we would like to introduce it as a subject for future work on resource allocation for Internet users.

Chapter 6

Summary and concluding remarks

In this dissertation we investigated the problem of broadcast scheduling in satellite and wireless systems. One of the most important problems in such systems is to find the optimal scheduling of the broadcasts of different information pages that results in the minimum average waiting time for the users. In our setting, we assumed that the system has N different information pages and K; $K < N$ broadcast channels and the request arrival rates for all pages are known. We presented a survey of the previous works and the Dynamic Programming(DP) formulation for this problem in Chapter 2 and also showed that the DP formulation by itself does not lead to any analytical solution. We used Restless Bandit approach to attack this problem and, by investigating the properties of a single-queue scheduling problem, showed that this approach can be applied to our problem to find near-optimal scheduling policies.

The first version of the problem where all files have equal lengths and different weights can be assigned to the files was investigated in Chapter 3. Our analysis led us to an index policy that, at each decision time, assigns an index to each queue and only broadcasts the pages that correspond to the K largest index values.

We also found a closed-form expression for the index function for the cases with small request arrival rates. Furthermore, having a near-optimal policy allowed us to evaluate the performances of several other heuristic policies presented by other researchers and to introduce extensions to those policies in some cases.

Another important extension of this problem where the pages were different in size was presented in Chapter 3. There, we used the same formulation and through either analytical or numerical methods showed that the problem can be solved using the same method. After deriving a formula for the index function for the case with Geometrically distributed file sizes, we performed a large number of simulation studies to compare the performance of this policy with a number of other heuristic policies. We also used the results to present an index policy for the deterministic file size case and evaluated its performance through simulation studies. Our results in both cases were either better than or equal to the results given by other heuristic policies.

Other possible extensions of this problem were introduced in Chapter 4. We presented our formulation for the cases where the arrivals are Markovian or the transmission error probability is not negligible. We also presented our preliminary findings to be used in future work on this matter. Another equally important direction for work on this type of systems is to consider the scheduling problem in the presence of deadlines and we think a similar approach can be used for that problem as well. As a general conclusion, we believe that the Restless Bandit formulation is a viable approach to address different problems in broadcast scheduling.

Finally, in Chapter 5, we presented some of the results of a traffic analysis project on the Internet traffic flowing through a satellite Internet delivery system. Our findings in these results were inline with similar studies on other networks and also enabled us to construct statistical models for various elements of the TCP traffic for use in simulation studies.

Appendix A

Derivation of the maximization problem

A.1 Derivation of the maximization problem for equal file sizes

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

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

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

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

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

$$\beta J_\beta = \beta E\left[\sum_{t=0}^{\infty}\beta^t[\sum_{i=1}^{N} c_i X_i(t)]\right] \tag{A.2}$$

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

$$\widehat{J_\beta} = E[\sum_{t=0}^{\infty}\beta^t \sum_{i\in d(t)} c_i X_i(t)] \tag{A.3}$$

which completes the derivation.

A.2 Derivation of the maximization problem for random file sizes

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

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

but if $d(s)$ is the set of pages transmitted at time s we have from equation (2.1) and (4.3)

$$X_i(s+1)+Y_i(s+1) = \begin{cases} X_i(s) + Y_i(s) + A_i(s) & if\ i \notin d(s) \\ X_i(s) + Y_i(s) + A_i(s) \quad with\, prob.\ 1-q_i & if\ i \in d(s) \\ Y_i(s) + A_i(s) \quad with\, prob.\ q_i & if\ i \in d(s)\ and\ X_i(s) > 0 \\ A_i(s) \quad with\, prob.\ q_i & if\ i \in d(s)\ and\ X_i(s) = 0 \end{cases} \tag{A.4}$$

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

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

We also have

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

therefore

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

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

$$\widehat{J_\beta} = E\left[\sum_{s=0}^{\infty} \beta^s \sum_{i \in d(s)} q_i c_i (X_i(t) I[X_i(t) > 0] + Y_i(t) I[X_i(t) = 0])\right] \tag{A.6}$$

which completes the derivation.

Appendix B

Analysis of some bulk service queueing systems

B.1 Proofs related to a single controlled bulk service queue

The following results investigate the form of the optimal policy for a single bulk queue. Here we assume that there is a constant service cost ν for each service(active period) but no cost for remaining idle. The reward for serving the queue at state (x, y) is defined by equation (2.3). The optimal policy is the policy that chooses to serve the queue for some states and remains idle for the other states so that the total expected discounted sum of the rewards is maximized.

B.1.1 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 B.1.1 *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*

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

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 B.1.2 *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 B.1.1 we have

$$S_p^{\pi_p^{x+1}}(x+1) \leq S_p^{\pi_p^{x+1}}(x) + c \tag{B.2}$$

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

$$V(x+1) \leq c + \sum_p P(p) S_p^{\pi_p^{x+1}}(x). \tag{B.3}$$

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

$$V(x) = \sum_p P(p) S_p^{\pi_p^x}(x) \geq \sum_p P(p) S_p^{d_p^Y}(X). \tag{B.4}$$

inequality (a) follows from combining the two above results.

Also, according to lemma B.1.1 we have

$$S_p^{\pi_p^x}(x) \leq S_p^{\pi_p^x}(x+1) \tag{B.5}$$

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

$$V(x) \leq \sum_p P(p) S_p^{\pi_p^x}(x+1). \tag{B.6}$$

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

$$V(x+1) = \sum_p P(p) S_p^{\pi_p^{x+1}}(x+1) \geq \sum_p P(p) S_p^{\pi_p^x}(x+1). \tag{B.7}$$

Hence inequality (b) follows.

Now, we can prove the following property:

Theorem B.1.3 *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:

$$cx - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq \beta \sum_{i=1}^{\infty} p(i) V(x+i) \tag{B.8}$$

Starting with the above property, we have

$$V(x+i) \leq c + V(x-1+i) \tag{B.9}$$

or

$$\beta \sum_{i=1}^{\infty} p(i) V(x+i) \leq c + \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{B.10}$$

Using the hypothesis, we have

$$cx - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq c + \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{B.11}$$

or

$$cx - c - \nu + \beta \sum_{i=1}^{\infty} p(i) V(i) \leq \beta \sum_{i=1}^{\infty} p(i) V(x-1+i) \tag{B.12}$$

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

B.1.2 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:

$$
\begin{aligned}
V^{\nu}(0) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(0+i) \geq -\nu + 0 + V^{\nu}(0) \\
V^{\nu}(1) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(1+i) \geq -\nu + c + V^{\nu}(0) \\
&\vdots \\
V^{\nu}(x) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \geq -\nu + cx + V^{\nu}(0) \\
&\vdots \\
V^{\nu}(s) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(s+i) \geq -\nu + cs + V^{\nu}(0) \\
V^{\nu}(s+1) &= -\nu + c(s+1) + V^{\nu}(0) \geq \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(s+1+i) \\
&\vdots
\end{aligned}
\tag{B.13}
$$

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:

$$
\begin{aligned}
V^{\nu'}(0) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(0+i) \\
V^{\nu'}(1) &= \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(1+i)
\end{aligned}
\tag{B.14}
$$

$$\vdots$$

$$V^{\nu'}(x) = \beta \sum_{i=0}^{\infty} p(i) V^{\nu'}(x+i)$$

$$\vdots$$

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

$$V^{\nu'}(s+1) = -\nu + c(s+1) + V^{\nu'}(0)$$

$$\vdots$$

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:

$$\Delta(x) = \beta \sum_{i=0}^{\infty} p(i) \Delta(x+i) \ \ for \ \ x \leq s \tag{B.15}$$

and

$$\Delta(x) = \beta \sum_{i=0}^{\infty} p(i) \Delta(i) + \Delta\nu = \Delta(0) + \Delta\nu \ \ for \ \ x > s \tag{B.16}$$

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

$$\Delta(x) = \beta \sum_{i=0}^{s-x} p(i) \Delta(x+i) + \beta(\Delta(0) + \Delta\nu) h(s+1-x) \ \ for \ \ x \leq s \tag{B.17}$$

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 B.1.4 *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 (B.17), 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 (B.14) 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 (B.13)

$$V^{\nu}(x) = \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \geq -\nu + cx + V^{\nu}(0) \tag{B.18}$$

also from the above lemma we have

$$\Delta(x) < \Delta\nu + \Delta(0) \tag{B.19}$$

therefore, subtracting (B.19) from (B.18), we have

$$V^{\nu'}(x) > -\nu' + cx + V^{\nu'}(0) \tag{B.20}$$

that is, the u policy is optimal for $0 \leq x \leq s$ states. For $x > s$ states from equation (B.13) we have

$$\begin{aligned} cx - \nu + V^{\nu}(0) &\geq \beta \sum_{i=0}^{\infty} p(i) V^{\nu}(x+i) \\ &= \beta \sum_{i=0}^{\infty} p(i)[cx + ci - \nu + V^{\nu}(0)] \\ &= \beta[cx - \nu + V^{\nu}(0)] + \beta c \lambda \end{aligned} \tag{B.21}$$

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 (B.21). Therefore, $x = s+1$ is the smallest integer(state) which satisfies (B.21). We know from lemma (B.1.4) that for policy u, $V^{\nu}(0)$ is a non-increasing function of ν(since $\Delta(0) > 0$). Hence, inequality (B.21) weakens as ν increases and the maximum value of ν for which the inequality still holds for $x = s+1$ is the one satisfying

$$c(s+1) - \nu^* + V^{\nu^*}(0) = \beta[c(s+1) - \nu^* + V^{\nu^*}(0)] + \beta c \lambda \tag{B.22}$$

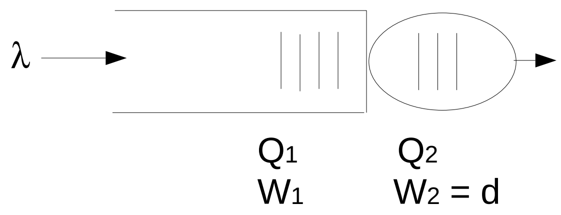

Figure B.1: A bulk service queuing system.

or

$$\nu^* = c(s+1) + V^{\nu^*}(0) - \frac{\beta c \lambda}{1-\beta}. \tag{B.23}$$

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 (B.23) and (3.22), 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.

B.2 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

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

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

time instants of distance d (figure B.1). 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 $\frac{\lambda d}{2}$. We have shown that the pmf of this random variable is the normalized complementary cmf of a Poisson process with rate λd(appendix A). Table B.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 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

$$\bar{W} = \bar{X}p_0 + (1 - p_0)(\bar{X} + \bar{R}) \tag{B.24}$$

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

$$\bar{X} < \bar{W} < 2\bar{X}. \tag{B.25}$$

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

Appendix C

Analysis of a bulk service queue with random file lengths

The following results investigate the form of the optimal policy for a single bulk queue with Geometric service times. Here we assume that there is a constant service cost ν for each service(active period) but no cost for remaining idle. The reward for serving the queue at state (x, y) is defined by equation (4.8). The optimal policy is the policy that chooses to serve the queue for some states and remains idle for the other states in such a way that the total expected discounted sum of the rewards is maximized.

C.1 Threshold property in the x direction

In this part we prove that the optimal policy is of the threshold type in the x direction. We show the value function of the optimal policy π^* by $V(.,.)$ and first prove some properties of this function in the following lemma.

Lemma C.1.1 *Let $S^d_{A,Q}(x, y)$ denote the resulting discounted reward sum when* *the initial condition is* (x, y)*, arrivals occur as time sequence* A*, service success occurs as sequence* Q *and the fixed decision sequence* $d = d_0, d_1, \ldots$ *is applied to the system. Then we have*

(a) $S^d_{A,Q}(x, y) \leq S^d_{A,Q}(x + 1, y) \leq S^d_{A,Q}(x, y) + c\,;\ x > 0$

(b) $S^d_{A,Q}(0, y) \leq S^d_{A,Q}(0, y + 1) \leq S^d_{A,Q}(0, y) + c$

(c) $S^d_{A,Q}(x, y) \leq S^d_{A,Q}(x, y + 1) \leq S^d_{A,Q}(x, y) + \beta c\,;\ x > 0$

Proof: (a) Consider two identical queues one with initial condition (x, y) and the other with initial condition $(x + 1, y)$. If the same set of arrival, success and decision sequences are applied to these two systems, the reward would be the same before the first successful service epoch say t_1. At that point, the second system receives a reward that is $c\beta^{t_1}$ units more than that received by the first system. Since the dynamics of the system forces the x component of the state to zero, the two systems will be identical thereafter. Therefore, the resulting rewards for both queues would be the same afterwards. Since $0 < \beta < 1$ and $c > 0$, the maximum difference occurs at $t_1 = 0$ and the minimum at $t_1 = \infty$.

(b) Since in a state of the form $(0, y)$ the service will be applied to component y, the same argument as part (a) is valid for this case.

(c) Consider two identical systems under similar arrival, success and decision sequences, with initial states (x, y) and $(x, y+1)$. the first successful service (at time t_1) changes the states to $(0, y+i)$ and $(0, y+1+i)$ respectively. The difference in the reward sequences occurs when the second successful service happens(t_2). At that time the reward of the second system is $c\beta^{t_2}$ units more than the reward of the first system and after that both systems go to identical states $(0, a_{t_2})$ where a_{t_2} is the number of new arrivals during period t_2. since t_2 occurs after t_1, the earliest time it happens can be $t_2 = 1$ which results in the right hand inequality. The left inequality is valid as before.

Using the above lemma it can be shown that

Lemma C.1.2 *The optimal value function $V(.,.)$ satisfies the following inequalities*

(a) $V(x,y) \leq V(x+1,y) \leq V(x,y) + c\,;\; x > 0$

(b) $V(0,y) \leq V(0,y+1) \leq V(0,y) + c$

(c) $V(x,y) \leq V(x,y+1) \leq V(x,y) + \beta c\,;\; x > 0$

Proof: By definition, the optimal value function $V^{\pi}(x,y)$ is the maximum attainable value of $E_{A,Q}\left[S_{A,Q}(x,y)\right]$ over all deterministic stationary policies. To prove (a), we take the expectation over A and Q of part (a) inequality of the previous lemma to get

$$E_{A,Q}\left[S^d_{A,Q}(x,y)\right] \leq E_{A,Q}\left[S^d_{A,Q}(x+1,y)\right] \leq E_{A,Q}\left[S^d_{A,Q}(x,y)\right] + c \; for\ all\ fixed\ d. \tag{C.1}$$

Here we used the fact that the success sequence is an iid, binary sequence with success probability q and independent of the arrival sequence. if we set d for every choice of A and Q to be the fixed control sequence dictated by the optimal decision policy π applied to a system with initial state $(x+1,y)$ (we show it by $d = \pi_{A,Q}(x+1,y)$) the middle term in (C.1) will be equal to $V^{\pi}(x+1,y)$. The right most term is definitely not greater than $V^{\pi}(x,y)$ because applying the control sequence optimized for the $(x+1,y)$ initial state to a system with initial state (x,y) can not produce an expected reward larger than that gained by the optimal policy. Hence we have $V^{\pi}(x+1,y) \leq V^{\pi}(x,y) + c$ which is the right inequality in (a). To prove the left inequality, again start from (C.1) and set d to be the sequence dictated by the optimal policy for A and Q but for initial state (x, y), i.e. $d = \pi_{A,Q}(x, y)$. The left most term of (a) is in fact $V^{\pi}(x, y)$ and the middle term is smaller than or equal to $V^{\pi}(x + 1, y)$ by definition. Therefore, the left inequality of (a) is valid too. Inequalities (b) and (c) can be proved using similar discussions.

We can now prove the main property of the optimal decision policy through the following theorem. Figure (4.2) can be viewed as a graphical representation of this property.

Theorem C.1.3 *The optimal policy π for the single-queue scheduling problem defined in section 4.4 is of the threshold type in the x direction. More specifically, the idling region in the (x, y) state space has the following property for every $x > 0$ and y; if $d^{\pi}(x, y) = 1$ then $d^{\pi}(x + 1, y) = 1$*

Proof: By definition

$$\begin{aligned} V(x,y) = & \\ & \max\left\{ -\nu + qcx + q\beta \sum_{i=0}^{\infty} p(i)V(0, y+i) + (1-q)\beta \sum_{i=0}^{\infty} p(i)V(x, y+i) \right. \\ & \left. , \beta \sum_{i=0}^{\infty} p(i)V(x, y+i) \right\} \end{aligned} \tag{C.2}$$

therefore, $d^{\pi}(x, y) = 1$ means

$$-\nu + qcx + q\beta \sum_{i=0}^{\infty} p(i)V(0, y+i) + (1-q)\beta \sum_{i=0}^{\infty} p(i)V(x, y+i) \geq \beta \sum_{i=0}^{\infty} p(i)V(x, y+i) \tag{C.3}$$

or

$$-\nu + qcx + q\beta \sum_{i=0}^{\infty} p(i)V(0, y+i) \geq q\beta \sum_{i=0}^{\infty} p(i)V(x, y+i). \tag{C.4}$$

Similarly,

$$V(x+1,y) = \tag{C.5}$$
$$\max\left\{-\nu + qcx + qc + q\beta\sum_{i=0}^{\infty} p(i)V(0,y+i) + (1-q)\beta\sum_{i=0}^{\infty} p(i)V(x+1,y+i)\right.$$
$$\left.,\beta\sum_{i=0}^{\infty} p(i)V(x+1,y+i)\right\}.$$

For $d^{\pi}(x+1,y) = 1$ we need to have

$$-\nu+qcx+qc+q\beta\sum_{i=0}^{\infty} p(i)V(0,y+i)+(1-q)\beta\sum_{i=0}^{\infty} p(i)V(x+1,y+i) \geq \beta\sum_{i=0}^{\infty} p(i)V(x+1,y+i). \tag{C.6}$$

Starting from equation (C.4) and using part (a) of lemma C.1.2, we have

$$\begin{aligned} -\nu + qc + qcx + q\beta\sum_{i=0}^{\infty} p(i)V(0,y+i) &\geq q\beta\sum_{i=0}^{\infty} p(i)V(x,y+i) + qc \qquad \text{(C.7)} \\ &\geq q\beta\sum_{i=0}^{\infty} p(i)[V(x,y+i)+c] \\ &\geq q\beta\sum_{i=0}^{\infty} p(i)V(x+1,y+i) \end{aligned}$$

which proves the theorem. This result shows that the for every fixed y, there is a threshold value x_{th} such that it is optimal to serve the queue for all states of the form $(x_{th}+i, y)$; $i = 0, 1, \ldots$ and remain idle for all states of the form (i, y); $i = 1, \ldots, x_{th} - 1$. State $(0, y)$ is an exception as we will show later.

C.2 Calculation of the index function in light traffic

In this chapter we derive an approximation of the switching curve in the light traffic regime. In this regime the arrival rate is a small number such that the probability of more than one arrivals during any broadcast period is negligible. Let's denote by $p_0(> 0)$ and $p_1 = 1 - p_0$ the probabilities of zero and one arrivals, respectively. We also use the threshold property in both the x $(x > 0)$ and y directions and the $y_0 > x_0$ assumption throughout this section. The latter property is valid when β is enough close to 1. Below, we investigate properties of the value function $V(.,.)$ of the optimal policy $d(.,.)$.

C.2.1 Properties of the value function

Starting from the $(0, 0)$ point we have

$$\begin{aligned} V(0,0) &= \beta p_0 V(0,0) + \beta p_1 V(0,1) \\ V(0,1) &= \beta p_0 V(0,1) + \beta p_1 V(0,2) \\ &\vdots \\ V(0,y_0-1) &= \beta p_0 V(0,y_0-1) + \beta p_1 V(0,y_0). \end{aligned}$$

where y_0 is the first point on the y axis outside of the idle region. Therefore we have

$$V(0,y) = cV(0,y-1) \quad y = 0,\ldots,y_0 \tag{C.8}$$

or

$$V(0,y) = c^y V(0,0) \quad y = 0,\ldots,y_0 \tag{C.9}$$

where

$$c = \frac{1-\beta p_0}{\beta p_1}. \tag{C.10}$$

Similarly, the same relation also holds for the other points along vertical lines inside the idle region, namely

$$V(x,y) = c^y V(x,0) \quad y = 0,\ldots,y_{x_0}; \;\; x = 1,\ldots,x_0 \tag{C.11}$$

where y_x is the first point along the $x = x$ line that is in the active region.

Another property is that the value function for all the points of the active region has a fixed increment in the x direction. To show this, let us consider a point (x, y); $x > 0$ such that $d(x, y) = 1$, i.e.

$$V(x,y) = -\nu + qx + \beta q p_0 V(0,y) + \beta q p_1 V(0,y+1) \tag{C.12}$$

$$+\beta(1-q)p_0 V(x,y) + \beta(1-q)p_1 V(x,y+1). \tag{C.13}$$

The threshold property in the x direction requires that

$$V(x+1,y) = -\nu + qx + q + \beta q p_0 V(0,y) + \beta q p_1 V(0,y+1) \tag{C.14}$$

$$+\beta(1-q)p_0 V(x+1,y) + \beta(1-q)p_1 V(x+1,y+1). \tag{C.15}$$

Let's define the following *difference* functions

$$\Delta_x V(x,y) \triangleq V(x+1,y) - V(x,y) \tag{C.16}$$

$$\Delta_{xy} V(x,y) \triangleq \Delta_x V(x,y+1) - \Delta_x V(x,y). \tag{C.17}$$

It is easy to show that, due to the threshold property of the optimal policy, function $\Delta_{xy}V(x,s)$ satisfies the following equations for all $s = y, y+1, \ldots$

$$\Delta_{xy} V(x,s+1) = \frac{1-\beta(1-q)p_0}{\beta(1-q)p_1} \Delta_{xy} V(x,s). \tag{C.18}$$

Since $\frac{1-\beta(1-q)p_0}{\beta(1-q)p_1} \geq 1$, using the results of lemma C.1.2, it can be shown that the trivial solution is the only acceptable solution for this set of equations, i.e.

$$\Delta_{xy} V(x,s) = 0; \; \forall s \geq y \tag{C.19}$$

which in turn leads us to the following result

$$\Delta_x V(x,s) = cte; \; \forall s \geq y. \tag{C.20}$$

This result is nothing but the constant increment property in the x direction. Although we used the light traffic assumption during the proof, this result also holds for the general case. The value of the increment can be calculated as follows. Let's denote by a the constant increment value in the x direction. For the general case, the value of a can be found as

$$V(x+1,y)-V(x,y)=q+(1-q)\beta\sum_{i=0}^{\infty}p(i)\left[V(x+1,y+i)-V(x,y+i)\right] \quad (C.21)$$

or

$$a=q+(1-q)\beta\sum_{i=0}^{\infty}p(i)a \quad (C.22)$$

or

$$a=\frac{q}{1-\beta(1-q)}. \quad (C.23)$$

This result also holds for $V(0,y)$ points with $y \geq y_0$. By definition

$$\begin{aligned} V(0,y) &= -\nu+qy+\beta q p_0 V(0,0)+\beta q p_1 V(0,1) \quad (C.24)\\ &+ \beta(1-q)p_0 V(y,0)+\beta(1-q)p_1 V(y,1); \forall y \geq y_0 \end{aligned}$$

which happens to be equal to the $V(y,0)$; $\forall y \geq y_0(> x_0)$ values. Therefore, the constant increment property holds with the same increment value.
Function $V(x,y)$ has the constant increment property in the y direction for $y \geq \max(y_0, y_x)$. This property can be proved with a similar argument as above and the amount of the increment in the y direction can be found to be

$$b=\frac{\beta q a}{1-\beta(1-q)}=\beta a^2. \quad (C.25)$$

We can now determine the location of the border points y_x; $x=0,\ldots,x_0$ as follows. Since by definition y_0 is the smallest y where $d(0,y)=1$, we have

$$V(0,y_0) \geq \beta p_0 V(0,y_0)+\beta p_1 V(0,y_0+1)=\beta V(0,y_0)+\beta p_1 a \quad (C.26)$$

or

$$V(0, y_0) \geq \frac{\beta p_1 a}{1-\beta}. \tag{C.27}$$

Also

$$\begin{aligned} V(0, y_0 - 1) \geq & -\nu + qy_0 - q + \beta q p_0 V(0,0) + \beta q p_1 V(0,1) \\ & +\beta(1-q)p_0 V(y_0 - 1, 0) + \beta(1-q)p_1 V(y_0 - 1, 1) \end{aligned} \tag{C.28}$$

or

$$V(0, y_0 - 1) \geq V(0, y_0) - a \tag{C.29}$$

or equivalently using C.8,

$$V(0, y_0) \leq \frac{ac}{c-1}. \tag{C.30}$$

Equations C.27 and C.30, together with equation C.8 and the increment property in the y direction result in the following inequalities

$$V(0, y_0 - 1) \leq \frac{a\beta p_1}{1-\beta} \leq V(0, y_0) \leq \frac{a(1-\beta p_0)}{1-\beta} \leq V(0, y_0 + 1). \tag{C.31}$$

The two constant values in this inequality have an interesting relation ship, that is,

$$\frac{a(1-\beta p_0)}{1-\beta} = \frac{a\beta p_1}{1-\beta} + a \tag{C.32}$$

and also

$$\frac{a(1-\beta p_0)}{1-\beta} = \frac{a\beta p_1}{1-\beta} \times c. \tag{C.33}$$

In other words, y_0 is located at a point where the slopes of the exponential portion of the $V(0, y)$ and the linear portion of it are very close (tangent in the continuous case). Similar arguments can be used to find limits on the value of the value function on the border points for $0 < x \leq x_0$. Specifically, assuming the typical case where $y_x \geq y_0$, it can be found that

$$V(x, y_x - 1) \leq \frac{b\beta p_1}{1-\beta} \leq V(x, y_x) \leq \frac{b(1-\beta p_0)}{1-\beta} \leq V(x, y_x + 1). \tag{C.34}$$

It can also be found, using the same assumption along with the previous results, that

$$V(x, y_x) = \frac{-\nu + bp_1}{q}a + b(y_x - y_0) + \beta a V(0, y_0) + ax \tag{C.35}$$

and

$$V(x+1, y_{x+1}) = \frac{-\nu + bp_1}{q}a + b(y_{x+1} - y_0) + \beta a V(0, y_0) + a(x+1). \tag{C.36}$$

Combining the above equations, we get

$$V(x+1, y_{x+1}) - V(x, y_x) = b(y_x - y_{x+1}) - a. \tag{C.37}$$

Finally, using inequality C.34, we have

$$-b \leq V(x+1, y_{x+1}) - V(x, y_x) \leq b \tag{C.38}$$

or

$$\frac{1}{\beta a} - 1 \leq y_x - y_{x+1} \leq \frac{1}{\beta a} + 1. \tag{C.39}$$

This equation puts a limit on the slope of the upper part of the switching curve for $0 < x \leq x_0$. The lower limit is always a positive number which conforms with the threshold property in the x direction.

C.2.2 Approximation of the index function

In order to find a closed form expression for the switching curve, we will try to find the border values y_x that put the (x, y_x) point on the border , i.e., make the idle and active decisions equally favorable. The resulting values are not necessarily integer but they very well approximate the actual location of the last point of the idle region and the first point of the active region. For point $(0, y_0)$ we have $d(0, y_0) = 0$ therefore

$$\begin{aligned} V(0, y_0) &= \beta p_0 V(0, y_0) + \beta p_1 V(0, y_0 + 1) \\ &= \beta p_0 V(0, y_0) + \beta p_1 V(0, y_0) + \beta p_1 a \end{aligned}$$

or

$$V(0, y_0) = \frac{\beta a p_1}{1 - \beta}. \tag{C.40}$$

Similarly, by equating the idle and active decisions for every point (x, y_x); $x = 0, \ldots, x_0$, we find that

$$V(x, y_x) = \frac{\beta b p_1}{1 - \beta}; \;\; x = 0, \ldots, x_0. \tag{C.41}$$

Since in the above derivations we have replaced $V(x, y+1)$ with $V(x, y) + b$, we need to make sure that we have $y_x \geq y_0$. We will check this requirement later in this section. For (x, y_x) we also have

$$\begin{aligned} V(x, y_x) &= -\nu + qx + \beta q p_0 V(0, y_x) + \beta q p_1 [V(0, y_x) + a] \\ &+ \beta(1 - q) p_0 V(x, y_x) + \beta(1 - q) p_1 [V(x, y_x) + b] \end{aligned}$$

or

$$V(x, y_x) = -\frac{\nu a}{q} + xa + \frac{b p_1 a}{q} + b\Delta_{x0} + \beta a V(0, y_0) \tag{C.42}$$

where $\Delta_{x0} \triangleq y_x - y_0$. If we replace the $V(x, y_x)$ and $V(0, y_0)$ by their corresponding values from equations (C.41) and (C.40), after some simplifications we have

$$\Delta_{x0} = \frac{1}{\beta a}\left[\frac{\nu}{q} - x\right] - \frac{a p_1}{q}; \;\; x = 1, \ldots, x_0. \tag{C.43}$$

This equation allows us to determine the value of y_x for any $x = 1, \ldots, x_0$ by having the y_0 value at hand. It also shows that all y_x; $x = 1; \ldots; x_0$ are along a straight line with negative slope $-1/\beta a$ which crosses the $y = y_0$ point at $x_0 = \frac{\nu - bp_1}{q}$ and since $b, p_1 < 1$ it is reasonable to assume that this point is very close to ν/q for larger values of ν. Therefore, all the y_x border points for $x = 1, \ldots, \lfloor \nu/q \rfloor$ lie above or equal to y_0. In order to find another equation for y_0, we can start from $V(y_0, y_0)$. Since (x_0, y_0) is on the border, we can write

$$\begin{aligned} V(y_0, y_0) &= V(x_0, y_0) + a(y_0 - x_0) \\ &= V(x_0, y_0) + a\left(y_0 - \frac{\nu - bp_1}{q}\right) \\ &= \frac{\beta b p_1}{1-\beta} + a\left(y_0 - \frac{\nu - bp_1}{q}\right). \end{aligned} \tag{C.44}$$

We can also write $V(y_0, y_0)$ in terms of $V(y_0, 0)$ and use the fact that $V(y_0, 0) = V(0, y_0)$ (this can be easily verified since $d(0, y_0) = d(y_0, 0) = 1$). Starting with $V(y_0, y_0 - 1)$ we have

$$\begin{aligned} V(y_0, y_0 - 1) &= -\nu + qy_0 + \beta q p_0 V(0, y_0 - 1) + \beta q p_1 V(0, y_0) \\ &+ \beta(1-q)p_0 V(y_0, y_0 - 1) + \beta(1-q)p_1 V(y_0, y_0) \end{aligned}$$

after some simplifications we have

$$V(y_0, y_0 - 1) = \frac{-\nu + qy_0}{d} + \frac{q}{d}c^{y_0 - 1}V(0,0) + \frac{f}{d}V(y_0, y_0) \tag{C.45}$$

where

$$d = 1 - \beta p_0(1-q) \tag{C.46}$$

and

$$f = \beta(1-q)p_1. \tag{C.47}$$

Using a similar approach and the above result, we can find a relation between $V(y_0, y_0 - 2)$ and $V(y_0, y_0)$. If we skip the interim calculations, the final result is

$$V(y_0, y_0 - 2) = \frac{-\nu + qy_0}{d}\left(1 + \frac{f}{d}\right) + \frac{q}{d}c^{y_0 - 2}V(0,0)\left(1 + \frac{fc}{d}\right) + (\frac{f}{d})^2 V(y_0, y_0). \tag{C.48}$$

This process can be continued until we find an equation that relates $V(y_0, 0)$ to $V(y_0, y_0)$ as follows

$$\begin{aligned} V(y_0,0) &= \frac{-\nu + qy_0}{d}\left(1+\frac{f}{d}+\ldots+(\frac{f}{d})^{y_0-1}\right) \\ &+ \frac{q}{d}V(0,0)\left(1+\frac{fc}{d}+\ldots+(\frac{fc}{d})^{y_0-1}\right) \\ &+ (\frac{f}{d})^{y_0}V(y_0,y_0). \end{aligned}$$

Since $f < d$ and also $fc < d$, the above equation becomes

$$\begin{aligned} V(y_0,y_0-2) &= \frac{-\nu + qy_0}{d}\frac{1-(\frac{f}{d})^{y_0}}{1-\frac{f}{d}} \\ &+ \frac{q}{d}V(0,0)\frac{1-(\frac{fc}{d})^{y_0}}{1-\frac{fc}{d}} \\ &+ (\frac{f}{d})^{y_0}V(y_0,y_0). \end{aligned}$$

Now, if we substitute $V(y_0, y_0)$ with its equivalent value from equation (C.44) and also replace $V(y_0, 0)$ by $V(0, y_0) = \frac{\beta p_1 a}{1-\beta}$, after a few steps we have

$$\begin{aligned} \frac{\beta p_1 a}{1-\beta} &= a\left[1-(\frac{f}{d})^{y_0}\right]\left(y_0-\frac{\nu}{q}\right) \\ &+ \left[1-(\frac{fc}{d})^{y_0}\right]V(0,0) \\ &+ \frac{\beta^2 p_1 a^2}{1-\beta}\left(\frac{f}{d}\right)^{y_0} + a\left(\frac{f}{d}\right)^{y_0}\left(y_0-\frac{\nu-bp_1}{q}\right). \end{aligned} \tag{C.49}$$

This is the main equation with y_0 as its unknown variable. Since we are mainly interested in the cases with $\beta \simeq 1$, we can perform more simplifications. As $\beta \to 1$, we have $a, b \to 1$ and also $\frac{f}{d}, \frac{fc}{d} \to 0$, therefore we have

$$y_0 = \frac{\beta p_1}{1-\beta} + \frac{\nu}{q} - \frac{1}{a}V(0,0). \tag{C.50}$$

We also know that

$$V(0,y_0) = c^{y_0}V(0,0) = \frac{\beta p_1 a}{1-\beta} \tag{C.51}$$

or

$$\frac{1}{a}V(0,0) = c^{-y_0}\frac{\beta p_1}{1-\beta}. \tag{C.52}$$

Substituting into equation (C.50), we get

$$y_0 = \frac{\nu}{q} + \frac{\beta p_1}{1-\beta}\left(1 - c^{-y_0}\right). \tag{C.53}$$

This equation is our light traffic approximation for values of β close to 1 for calculating the value of y_0. It also shows that, for relatively large β, y_0 remains well above the $x_0 \simeq \nu/q$ value. Having found y_0, other y_x values can be found from equation (C.43), which together with the vertical line $x = \lfloor \nu/q \rfloor$ define the borders of the idle region. To double check the correctness of our approximations, below we compare our above analytical results with the exact numerical results for some cases. Figure C.1 shows the exact shape of the switching curve calculated from numerical solution of the dynamic programming equation, together with the approximate curve defined by the above method for two values of $\lambda = 0.9, 1.0$, with different q and ν choices. It can be seen that in all cases there is a good match between the exact and approximate curves. Although the light traffic approximation is derived from the assumption of zero probability for more than one arrivals, it is still possible to apply the above method to the general case by replacing p_1 by λ. However, we no longer expect a good matching between the exact and approximate curves. In fact, for relatively short file sizes, or equivalently large q values, the approximate method continues to match the exact curve to a reasonable degree for arrival rates up to $\lambda = 13$ (figure C.2). But this matching disappears quickly in systems with relatively large file sizes as can be seen in figure C.3 where for $q = 0.2$ the approximate value is not acceptable for rates higher than 1.

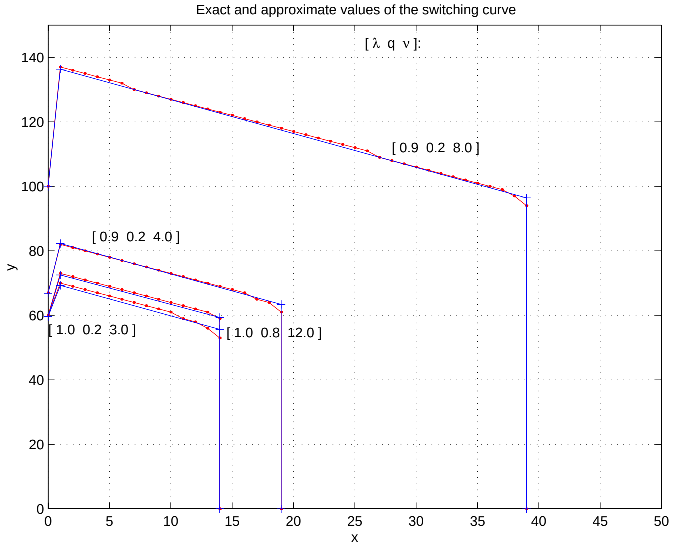

Figure C.1: Comparison between the exact and approximate curves for different parameter values.

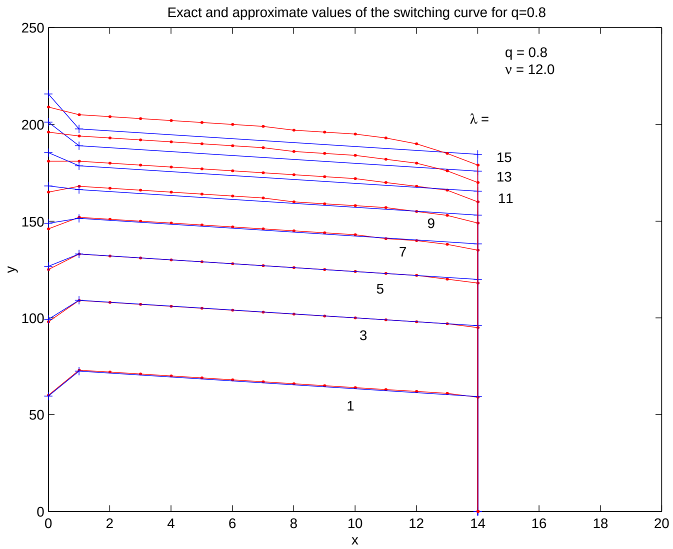

Figure C.2: Comparison between the exact and approximate curves for different rates and q=0.8.

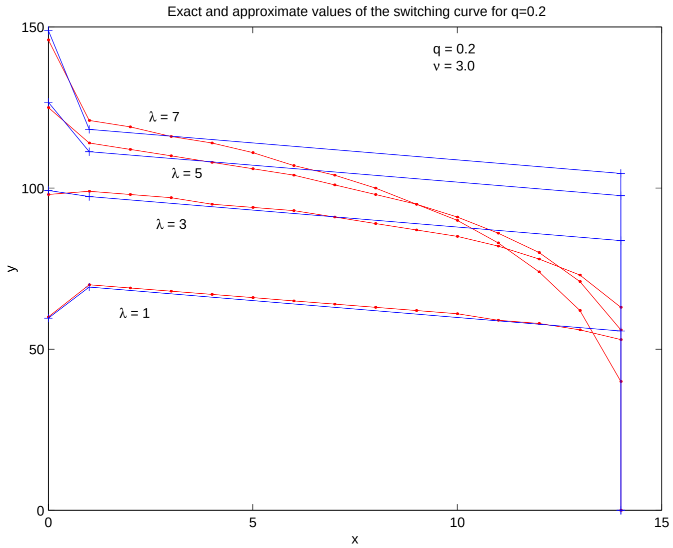

Figure C.3: Comparison between the exact and approximate curves for different rates and q=0.2.

C.2.3 Relation between the switching curve and the service cost in light traffic

We need to show that both y_0 and y_1 values are non-decreasing functions of ν. The non-decreasing property, in fact, comes from the integer nature of these values. Otherwise they would have been increasing functions. We start with y_0 and prove that $\partial\nu/\partial y_0 > 0$. We have

$$\nu = qy_0 - \frac{q\beta p_1}{1-\beta}\left(1 - c^{-y_0}\right). \tag{C.54}$$

Therefore

$$\frac{\partial\nu}{\partial y_0} = q - \frac{q\beta p_1}{1-\beta}\ln c\, c^{-y_0}. \tag{C.55}$$

The above derivative is positive only if we have

$$\frac{\beta p_1}{1-\beta}\ln c\, c^{-y_0} < 1 \tag{C.56}$$

since

$$c = \frac{1-\beta p_0}{\beta p_1} > 1, \tag{C.57}$$

for $y_0 \geq 0$ we have $c^{-y_0} \leq 1$. To show that the remaining term is also less than one, we write

$$\begin{aligned}\frac{\beta p_1}{1-\beta} \ln c &= \frac{\beta p_1}{1-\beta} \ln(1 + \frac{1-\beta}{\beta p_1}) \\ &= \frac{\ln(1+t)}{t}\end{aligned}$$

where

$$t = \frac{1-\beta}{\beta p_1} > 0. \tag{C.58}$$

Since $\ln(1+t)$ is a strictly concave increasing function and we have $\ln(1+t) = t$ at $t = 0$, therefore for all $t > 0$ we have $\ln(1+t) < t$ which proves the result. Therefore, y_0 is a non-decreasing function of ν.

For y_1, we have

$$y_1 = y_0 + \frac{1}{\beta a}\left[\frac{\nu}{q} - 1\right] - \frac{ap_1}{q}. \tag{C.59}$$

Again, using the above result about $\partial y_0/\partial \nu$, it is obvious that y_1 is also a non-decreasing function of ν.

C.3 Other properties of the optimal policy

This section contains the results we obtained from applying the Policy Iteration method to prove the properties of the optimal policy and the decision region defined by it. As we mentioned in Chapter 4, the proof is not complete and requires more efforts. However, the discussions and partial results presented below will be essential for a complete proof. Depending on the form of the optimal policy and the decision region defined by it, the proof can have many special cases where each need to be addressed individually. Here, at some points, we deliberately ignore the special cases and only focus on the typical cases to capture the main theme of our arguments which can be applied to the other cases as well.

In this method, we consider a sequence of policies $d^n(.,.)$; $n = 0, \ldots$ generated by successive application of the Policy Iteration method [48] to an initial arbitrarily selected policy. In regular conditions, this sequence will converge to the optimal policy for the problem. If the initial policy $d^0(.,.)$ is chosen such that it possesses certain properties and if we can show that those properties are inherited by the successive policies generated as above, then the optimal policy as the limit of the sequence will have those properties as well. In our problem, several properties need to be proved at the same time. the minimum number of properties required for the proof are as follows

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 $\beta >> 0$ we have $y_0 > x_0$.

Here, we will only show that if the $d^n(.,.)$ policy has all of the above properties, then $d^{n+1}(.,.)$ will have properties 2, 3 and, 4. A complete proof needs to check the other two properties as well, though the last property does not seem to be a strict requirement if β is enough large.

Let's denote by $V^n(x,y)$ the value function associated with policy $d^n(x,y)$. By definition, for (x,y) such that $d^n(x,y)=1$ we have

$$\begin{aligned} V^n(x,y) &= -\nu + qx + \beta q p_0 V^n(0,y) + \beta q p_1 V^n(0,y+1) \\ &+ \beta(1-q)p_0 V^n(x,y) + \beta(1-q)p_1 V^n(x,y+1) \end{aligned} \tag{C.60}$$

and if $d^n(x,y)=0$,

$$V^n(x,y) = \beta p_0 V^n(x,y) + \beta p_1 V^n(x,y+1). \tag{C.61}$$

As we showed before, since $d^n(.,.)$ is a threshold policy, $V^n(x,y)$ has the constant increment property in the active region in the x direction for $x>0$ and in the y direction for $y \geq \max(y_0, y_x)$. The modified policy $d^{n+1}(.,.)$ is found by applying the DP operator L to the $V^n(.,.)$ function as follows.

$$d^{n+1}(x,y) = arg \max_{i=0,1} \{L_i V^n(x,y)\} \tag{C.62}$$

where

$$L_0 V^n(x,y) \triangleq \beta p_0 V^n(x,y) + \beta p_1 V^n(x,y+1) \tag{C.63}$$

and

$$\begin{aligned} L_1 V^n(x,y) \triangleq\ & -\nu + qx + \beta q p_0 V^n(0,y) + \beta q p_1 V^n(0,y+1) \\ & +\beta(1-q)p_0 V^n(x,y) + \beta(1-q)p_1 V^n(x,y+1). \end{aligned} \tag{C.64}$$

Similarly, for the points on the y axis we have

$$L_0 V^n(0,y) \triangleq \beta p_0 V^n(0,y) + \beta p_1 V^n(0,y+1) \tag{C.65}$$

and

$$\begin{aligned} L_1 V^n(0,y) \triangleq\ & -\nu + qy + \beta q p_0 V^n(0,0) + \beta q p_1 V^n(0,1) \\ & +\beta(1-q)p_0 V^n(y,0) + \beta(1-q)p_1 V^n(y,1). \end{aligned} \tag{C.66}$$

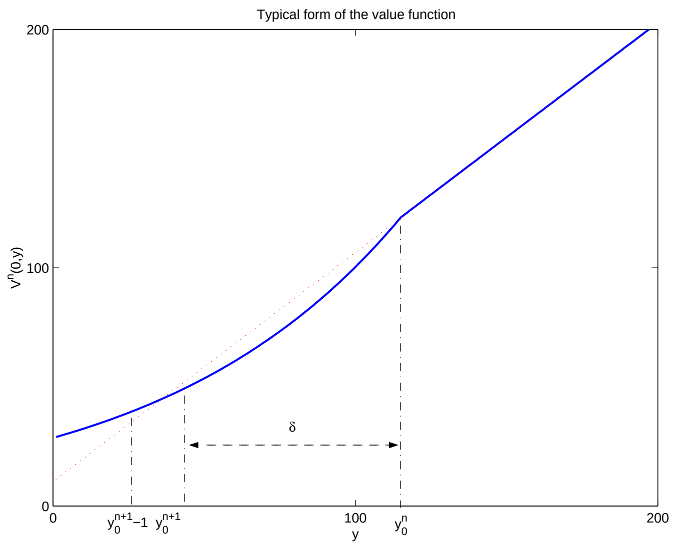

Figure C.4: Typical shape of the value function along the y axis.

Now, let's denote by y_0^n the threshold value on the $x = 0$ axis defined by policy $d^n(.,.)$. Based on the previous results we have

$$V^n(0, y) = V^n(0, y_0^n) + a(y - y_0^n) \;\; for \;\; y \geq y_0^n \tag{C.67}$$

and

$$V^n(0, y) = V^n(0, y_0^n)c^{-(y - y_0^n)} \;\; for \;\; y < y_0^n. \tag{C.68}$$

The new threshold y_0^{n+1} associated with policy $d^{n+1}(.,.)$ is found by applying the above operator to $V^n(.,.)$. We need to show that if y_0^{n+1} is the smallest y for which $d^{n+1}(0, y) = 1$ then $d^{n+1}(0, y) = 1$ for all larger values of y as well i.e., the threshold property in the y direction for $x = 0$ remains valid during the Policy Iteration steps. If we denote by y_0^{n+1} the new threshold along the y axis, we have

$$\begin{aligned} V^n(0, y_0^{n+1} - 1) \;\; &> \;\; -\nu + qy_0^{n+1} - q + qV^n(0, 0) \\ &+ \;\; \beta(1 - q)[p_0 V^n(y_0^{n+1} - 1, 0) + p_1 V^n(y_0^{n+1} - 1, 1)] \end{aligned} \tag{C.69}$$

and

$$
\begin{aligned}
cV^n(0, y_0^{n+1} - 1) &< -\nu + qy_0^{n+1} + qV^n(0,0) \\
&+ \beta(1-q)[p_0 V^n(y_0^{n+1}, 0) + p_1 V^n(y_0^{n+1}, 1)] \\
&= -\nu + qy_0^{n+1} + qV^n(0,0) \\
&+ \beta(1-q)[p_0 V^n(y_0^{n+1} - 1, 0) + p_1 V^n(y_0^{n+1} - 1, 1)] \\
&+ \beta p_0 (1-q) a + \beta p_1 (1-q) a \\
&= -\nu + qy_0^{n+1} + qV^n(0,0) \\
&+ \beta(1-q)[p_0 V^n(y_0^{n+1} - 1, 0) + p_1 V^n(y_0^{n+1} - 1, 1)] \\
&+ \beta(1-q)a.
\end{aligned} \tag{C.70}
$$

The new threshold value y_0^{n+1} can be smaller, equal, or larger than y_0^n depending on the $d^n(.,.)$ policy. Let's investigate the case where $y_0^{n+1} - 1 < y_0^n$ but it is still larger then x_0. In that case we have

$$
V^n(0, y_0^n) = V^n(y_0^n, 0) \tag{C.71}
$$

and

$$
V^n(0, y_0^{n+1} - 1) = V^n(y_0^{n+1} - 1, 0) \tag{C.72}
$$

where $\delta \triangleq y_0^n - y_0^{n+1}$. The right hand side of equation C.69 can then be written as

$$
\begin{aligned}
& -\nu + qy_0^{n+1} - q + qV^n(0,0) + \beta(1-q)[p_0 V^n(y_0^{n+1} - 1, 0) + p_1 V^n(y_0^{n+1} - 1, 1)] \\
= \; & -\nu + qy_0^n + qV^n(0,0) + \beta(1-q)[p_0 V^n(y_0^n, 0) + p_1 V^n(y_0^n, 1)] \\
& -q - q\delta - \beta(1-q)[p_0(\delta+1)a + p_1(\delta+1)a] \\
= \; & V^n(0, y_0^n) - (\delta+1)\left(q + \beta(1-q)a\right) \\
= \; & V^n(0, y_0^n) - a(\delta+1).
\end{aligned} \tag{C.73}
$$

Therefore, equations C.69 and C.70 can be written as

$$V^n(0, y_0^{n+1} - 1) > V^n(0, y_0^n) - a(\delta + 1) \tag{C.74}$$

and

$$cV^n(0, y_0^{n+1} - 1) < V^n(0, y_0^n) - a\delta. \tag{C.75}$$

If we define $K \triangleq V^n(0, y_0^n)$, the above equations become

$$Kc^{-(\delta+1)} > K - a(\delta + 1) \tag{C.76}$$

and

$$Kc^{-\delta} < K - a\delta. \tag{C.77}$$

These equations have a simple geometric interpretation. They show that the distance of the new threshold value from the old threshold is the largest integer point before the intersection of the extension of the linear portion of the value function (with slope $-a$) to the smaller y values and, the exponential portion of the value function as shown in figure C.4. In other words, the improved policy $d^{n+1}(0, y)$, assigns the idle policy to all y's where the exponential curve is above the linear extension, and the active policy for all y's where it is under the linear function. Since $Kc^{-\delta}$ is a convex function of δ, it has at most two intersection points with any straight line including the $K - a\delta$ function. Therefore, the line remains above the exponential function for all $(0, y)$ points where $y_0^{n+1} < y < y_0^n$ i.e., the improved policy is 1 for all those points and 0 for all points with $y < y_0^{n+1}$. We only need to show that the improved policy is also 1 for $y \geq y_0^n$. For this we need to have

$$\begin{aligned} \beta p_0 V^n(0, y) + \beta p_1 V^n(0, y + 1) &< -\nu + qy + qV(0, 0) \\ &+ \beta(1 - q)p_0 V^n(y, 0) + \beta(1 - q)p_1 V^n(y, 1) \\ &= V^n(0, y). \end{aligned} \tag{C.78}$$

or

$$V^n(0,y) > \frac{\beta a p_1}{1-\beta}. \tag{C.79}$$

It is easy to observe that in this case we have (figure C.4),

$$V^n(0,y_0^n) - a > V(0,y_0^n-1) = \frac{V^n(0,y_0^n)}{c} \tag{C.80}$$

or

$$V^n(0,y_0^n) > \frac{ac}{c-1} = \frac{a(1-\beta p_0)}{1-\beta} > \frac{a\beta p_1}{1-\beta}. \tag{C.81}$$

Hence, the improved policy is also 1 for $y \geq y_0^n$ and it is therefore of threshold type along the $x=0$ axis. The optimal threshold along the $x=0$ axis (i.e. y_0^∞) will eventually be at the point where the straight line with slope $-a$ will be tangent to the exponential curve (or the closest integer approximation). The proof for the $y_0^{n+1} > y_0^n$ case is almost trivial using the above reasonings and we will not discuss it here.

The second property that we need to show to remain valid during the Policy Iteration stages is the threshold property along each of the vertical axes with $0 < x < x_0$. In other words, we need to show that the improved policy $d^{n+1}(x,y)$ for every x as above, will have a threshold value y_x^{n+1} where $d^{n+1}(x,y) = 0$ for $y < y_x^{n+1}$ and is 1 for all other y. The proof for a typical case where $y_x^n > y_0^n$ is very similar to the above discussions with only the parameter a being replaced by b. As we showed in the previous sections where we tried to compute the index function, for $y_x^n < y_0^n$ the exponential function will be replaced by a polynomial function and the argument becomes more and more lengthy and complicated and since this case is not very frequent, we will not discuss it here.

Property 4 is the last property we would like to discuss in this section. As before, we need to show that if this property holds for $d^n(.,.)$, it will also hold for $d^{n+1}(.,.)$.

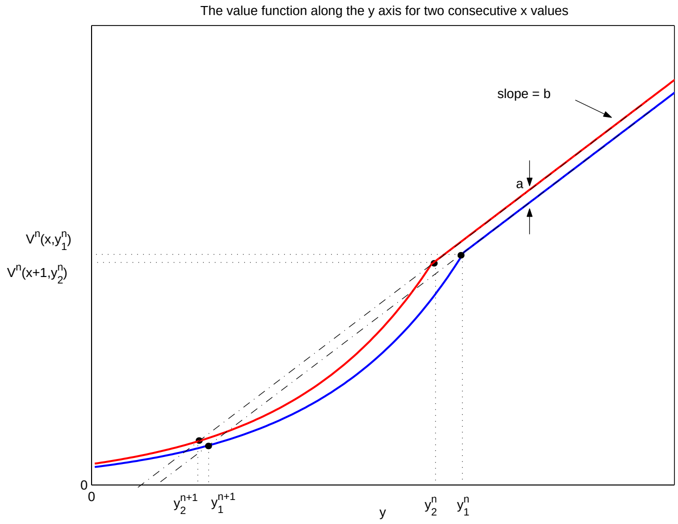

Figure C.5: The value function along the y axis for two consecutive x values.

Let's assume the threshold points corresponding to two successive x_1 and $x_2 = x_1 + 1$ values under the d^n policy to be y_1^n and y_2^n, respectively. By assumption we have

$$y_1^n - \frac{1}{\beta a} - 1 \leq y_2^n \leq y_1^n. \tag{C.82}$$

Therefore,

$$\begin{aligned} V^n(x+1, y_2^n) &\geq V^n(x+1, y_1^n) - b(\frac{1}{\beta a} + 1) \\ &= V^n(x, y_1^n) + a - b(\frac{1}{\beta a} + 1). \end{aligned} \tag{C.83}$$

or

$$V^n(x+1, y_2^n) \geq V^n(x, y_1^n) - b. \tag{C.84}$$

Now, the mechanism of finding the corresponding threshold values for the improved policy d^{n+1} is according to the Policy Iteration rule and can be explained by the same geometric discussion we presented before for each of y_1^{n+1} and y_2^{n+1} values as shown in figure C.5. For $y \geq y_1^n$, both $V^n(x, y)$ and $V^n(x+1, y)$ functions are straight lines with slope b and with $V^n(x+1,y) = V^n(x,y) + a$. However, $V^n(x,y)$ switches to the exponential curve at $y = y_1^n$ while $V^n(x+1,y)$ does it at $y = y_2^n$. The new threshold values for each of these functions are the intersections of the linear extensions of the top portion of each function with the exponential portion, as described before. The goal is to show that the new threshold values y_1^{n+1} and y_2^{n+1} satisfy the above inequality relationship. We showed in equation C.83 that $V^n(x+1, y_2^n)$, which is the beginning point of the exponential part for the $V^n(x+1,y)$ function (from above), is within certain range of the $V^n(x, y_1^n)$ value. Let's define

$$\delta_1 = y_1^n - y_1^{n+1} \tag{C.85}$$

and

$$\delta_2 = y_2^n - y_2^{n+1} \tag{C.86}$$

As we showed before each of the δ_i; $i = 1, 2$ values are the solutions of the

$$K_i c^{-\delta_i} = K_i - b\delta_i \tag{C.87}$$

equation with $K_1 = V^n(x, y_1^n)$ and $K_2 = V^n(x+1, y_2^n)$. It is easy to show using the behavior of $\frac{\partial \delta}{\partial K}$ and equation C.83 that, the difference between δ_1 and δ_2 values is bounded and keeps the resulting $y_1^{n+1} - y_2^{n+1}$ value within the limits specified by this property. In other words, the new threshold values for consecutive xs maintain the distance bound.

We finish our discussion of the properties of the optimal policy by reminding that the above arguments are guidelines capturing the most frequent forms of the switching form and a complete proof requires a more detailed and lengthy analysis.

BIBLIOGRAPHY

[1] A. C. Gilbert A. Feldmann and W. Willinger. Data networks as cascades: Investigating the multifractal nature of internet wan traffic. *Computer Communication Review, Vol. 28*, pp42-55, 1998.

[2] P. Abry and D. Veitch. Wavelet analysis of long-range dependent traffic. *IEEE transactions on Information Theory, Vol. 44*, pp2-15, 1998.

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

[4] 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.

[5] J. S. Bendat and A. G. Piersol. *Random Data, analysis and measurement procedures.* John Wiley & Sons, 1986.

[6] J. Beran. *Statistics for long-memory processes.* Chapman & Hall, 1994.

[7] 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.

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

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

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

[11] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

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

[13] D. Heyman et. al. A new method for analyzing feedback-based protocols with applications to engineering web traffic over the internet. *Performance Evaluation Review*, Vol. 25, No. 1, 1997.

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

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

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

[17] M. Taqqu et. al. Proof of a fundamental result in self-similar traffic modeling. *ACMCCR: Computer Communication Review*, 27, 1997.

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

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

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

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

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

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

[24] A. Hordijk. Insensitivity for stochastic networks. *Mathematical Computer Performance Reliability, eds G. Lazeolla, Elsevier Pub.*, pp77-94, 1984.

[25] D. Iglehart and A. P. Lalchandani. Diffusion approximations for complex repair systems. *Naval Research Logistics*, Vol. 38, pp817-838, 1991.

[26] 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.

[27] 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.

[28] N. I. Johnson and S. Kotz. *Continuous univariate distributions-1.* Hoghton Mifflin, 1970.

[29] M. Crovella K. Park, G. Kim. On the relationship between file sizes, transport protocols, and self-similar network traffic. *Proc. SPIE int. conp. perf. cont. network systems*, Nov. 1997.

[30] 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.

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

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

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

[34] M. Krunz and A. Makowski. Modeling video traffic using m/g/infinity input processes: A compromise between markovian and lrd models. *IEEE Journal on Selected Areas in Communications (JSAC), Vol. 16*, pp733-748, 1998.

[35] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic methods in stochastic modeling.* ASA-SIAM, 1999.

[36] 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.

[37] A. A. Alzaid M. A. Al-Osh. First-order integer-valued autoregressive (inar(1)) process. *J. of Time Series Analysis, Vol. 8*, No.3, pp261-275, 1987.

[38] E. McKenzie. Some arma models for dependent sequences of poisson counts. *Adv. Appl. Prob., Vol. 20*, pp822-835, 1988.

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

[40] 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.

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

[42] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover, 1995.

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

[44] 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.

[45] Kihong Park and Walter Willinger. *Self-Similar traffic and performance evaluation*. Wiley Inter-Science, 2000.

[46] V. Paxon. Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM transactions on Networking, Vol. 12*, pp316-336, 1994.

[47] S. I. Pederson and M. E. Johnson. Estimating model discrepancy. *Technometrics, Vol. 32*, pp305-314, 1990.

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

[49] H. Kobayashi Q. Ren. Diffusion approximation modeling for markov modulated bursty traffic and its applications to bandwidth allocation in atm networks. *IEEE JSAC*, Vol. 16, No. 5, pp679-691, Jun 1998.

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

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

[52] R. Schassberger. Insensitivity of steady-state distributions of generalized semi-markov processes. part i. *The Annals of Probability*, Vol.5, No.1, pp87-99, 1977.

[53] R. Schassberger. Insensitivity of steady-state distributions of generalized semi-markov processes. part ii. *The Annals of Probability*, Vol.6, No.1, pp85-93, 1978.

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

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

[56] S. Floyd V. Paxon. Wide area traffic: the failure of poisson modelling. *IEEE/ACM Transactions on Networking, Vol. 3*, pp226-244, 1995.

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

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

[59] 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.

[60] W. Willinger W. E. Leland, M. S. Taqq and D. V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[61] 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.

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

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