SCHEDULING IN WIRELESS CELLULAR DATA NETWORKS

Nomesh Bolia

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research (Operations Research).

Chapel Hill
2009

Approved by,

Vidyadhar Kulkarni, Advisor

Serhan Ziya, Committee Member

Nilay Argon, Committee Member

Jasleen Kaur, Committee Member

Haipeng Shen, Committee Member

ABSTRACT

NOMESH BOLIA: Scheduling in Wireless Cellular Data Networks
(Under the direction of Professor Vidyadhar Kulkarni)

This thesis studies the performance of scheduling policies in a wireless cellular data network. We consider a cell within the network. The cell has a single base station serving a given number of users in the cell. Time is slotted and the base station can serve at most one user in a given time slot. The users are mobile and therefore the data transfer rate available to each user changes from time slot to time slot depending on the distance from the base station and the terrain of the user.

There are two conflicting objectives for the base station: maximize the data throughput per time slot, and maintaining "fairness". To maximize the data throughput, the base station would like to serve the user with the highest available data rate, but this can lead to starvation of some users. To ensure "fairness", no user should be unserved for a "long" time, i.e., users should be served in a round-robin manner. Although this problem has been studied in the literature to some extent, existing methods to do this are ad-hoc. Our goal is to derive policies that have a sound theoretical basis, and at the same time are computationally tractable, are easy to implement, are fair to all the users and beneficial for the service providers.

We formulate the problem of finding an optimal scheduling policy as a Markov Decision Process (MDP) and prove some characteristics of the optimal policy. Since solving the MDP to optimality is infeasible, given the huge size of the problem, we develop heuristic policies called "index policies". These policies are based on a closed form "index" for every user that depends only its own current state. We derive this index using a policy improvement approach based on Markov Decision Processes. We also compare their performance with existing policies through simulation. We develop such index policies in two settings: when every user always has ample data waiting for it to be served (the infinitely backlogged case), and when data arrives for every user in every time slot according to some distribution (the external data arrival case).

Further, we consider the case of users entering and leaving the cell as well, but only from a simulation perspective.

ACKNOWLEDGEMENTS

I am sincerely grateful to my advisor Professor Vidyadhar G. Kulkarni for introducing me to the area of stochastic modeling and for his direction and guidance over the past three years. Words are not enough to express my gratitude for his dedication and encouragement, which made my dissertation research a truly joyful and rewarding experience. He provided me with an environment conducive to self-motivated research and one in line with my own interests. His approach of providing mentoring on matters ranging from very little ones in the beginning to bigger ones like the broad direction of the dissertation helped me immensely. Besides knowledge, one important thing he taught and showed me is the persistence to strive for perfection. His advice of thinking about the problem even while not actually working on it, and deriving the pleasure that only such intellectual pursuits can provide inspired me to be a scholar and shall influence and benefit me for the rest of my life.

I would also like to thank my committee members, Professor Serhan Ziya, Professor Nilay Argon, Professor Jasleen Kaur and Professor Haipeng Shen, for their helpful suggestions on my research.

Special thanks to Professor Kevin Glazebrook of Lancaster University Management School, Dr. Siamak Sorooshyari of Alcatel-Lucent and Professor George Shanthikumar of University of California, Berkeley for discussions and valuable feedback that helped improve the quality of this work. Further, I also thank one of the reviewers of one of our papers whose persistence led us to identify an error in our simulation in the dynamic cell case for infinitely backlogged queues. I would also like to thank Professor Paul H. Zipkin from Duke University and Professor Eda Kemahlioglu-Ziya for their enlightening courses on inventory management and supply chain management which greatly broadened my horizons in the area of stochastic modeling. Big thanks to Professor E. Michael Foster and Professor Serhan Ziya for their advice and support to me during my research assistantship with them. This research assistantship gave me a glimpse of how OR models can be applied to a subjective problem such as appropriate rehabilitation policy for victims of substance abuse. Professor Foster's ability to interpret OR in the context of health policy problem inspired me to apply models in a relevant way to real life situations. My sincere thanks to Professor Gabor Pataki for introducing me to the area of deterministic modeling and Professor Edward Carlstein for his encouraging help and priceless advice on my career development. I have learned a number of effective methods of teaching and accompanying evaluation from Professor Carlstein.

Thanks to the faculty and staff of our department and my fellow graduate students and others at UNC who have helped me throughout the years. They made my experience in Chapel Hill a beautiful and unforgettable one.

Last, but not the least, I would like to thank my parents and my sister for their love and support over the years and for bearing with my neglect of family duties. This acknowledgement would be absolutely incomplete without a warm mention of my friends, and mentors in the various socio-cultural activities with which I have been involved. I would again like to strongly thank my advisor Prof Vidyadhar Kulkarni without whose invaluable guidance and mentorship this dissertation would not have been possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless Cellular networks have long been used for voice communication effectively. The proliferation of cell phone usage in the past decade across the world is probably one of the most visible signs of the advancement of technology. With improvements in the technology, data transfer over the internet has become an important application of wireless cellular systems. In fact the revenues from mobile data services reached \$188.7 billion in 2008, representing a 24% year-on-year increase according to data sourced from Informa Telecoms & Media's latest report [1]. This also means that mobile operators now generate approximately one fifth of their revenue from data services; a development that is considered significant, given that a general slowdown in voice revenues is a cause of concern for mobile operators. Further, at the end of 2008 40% of the data revenue was from non-SMS services pointing to the emergence of ever-newer applications of data services that inevitably require high speed data transmission. An example of such an application is high speed (broadband) internet surfing on cell phones. With these futuristic applications in mind, we study ways to improve the performance of high speed data transmission in wireless cellular networks in this thesis. We begin with a brief overview of the cellular technology [2].

1.1 Overview of Cellular Technology

Cellular network systems facilitate mobility in communication. These systems achieve mobility by transmitting data through radio waves. Cellular networks derive their name from cells, i.e., small geographical areas that cover the entire region the cellular network intends to serve. Each cell is serviced by one radio transceiver (transmitter/receiver) called the base station. The cellular structure of the network enables frequency reuse. Cells, a certain distance apart, can reuse the same frequencies ensuring efficient usage of limited radio resources [3]. Communication in a cellular network is full duplex, i.e., communication is attained by sending and receiving messages on two different frequencies and hence at the same time.

1.1.1 History and Present State of Cellular Radio Networks

The first car-based telephone was set up in St. Louis, Missouri, USA in 1946. The system used a single radio transmitter on top of a tall building. A single channel (frequency) was used for transmission, therefore requiring a button to be pushed to talk, and released to listen [3]. Such a system, referred to as a half duplex system (as opposed to a full duplex system like the current cellular networks), is still used by modern day CB radio systems utilized by police and taxi operators. In the 1960s, the system was improved to a two-channel system called the improved mobile telephone system (IMTS) [3]. Since frequencies were limited, the system could not support many users.

Cellular radio systems, implemented for the first time in the advanced mobile phone system (AMPS), support more users by allowing reuse of frequencies. AMPS is an analog system, and a part of first generation (1G) cellular radio systems. In contrast, second generation systems are digital. In the USA, two standards were introduced for second generation systems: IS-95 (popularly known as CDMA, acronym for Code Division Multiple Access - the technology used for digital data transfer) and IS-136 (popularly known as D-AMPS and based on Time Division Multiple Access, i.e., TDMA - again, another technology like CDMA for communication) [3, 4]. Europe consolidated to one system called the global system for mobile communications (GSM, based on TDMA) [4]. Even in the US, most mobile operators working with the TDMA technology have migrated to GSM. Japan uses a system called personal digital cellular (PDC) that works on a technology similar to GSM.

Today Cellular radio is the fastest growing segment of the communications industry [3]. According to GSMA, an international mobile communications industry group, the total number of cell phone subscribers recently crossed the four billion mark and is expected to reach six billion by 2013 [5].

Current cellular radio systems are in their second generation (2G). The third generation of cellular systems (3G systems) will allow different systems to interoperate in order to attain global roaming across different cellular radio networks as well as allow new applications such as high speed internet surfing, multimedia messaging and video conferencing [6]. The International Telecommunication Union (ITU) has been doing research on 3G systems since the mid 1980s. Their version of a 3G system is called international mobile telecommunications - 2000 (IMT-2000).

European countries are researching 3G systems under the auspices of the European Community [6]. Their system is referred to as the universal mobile telecommunication system (UMTS), having the same goals as the IMT-2000 system. 3G systems have the following major objectives:

- Use of common global frequencies for all cellular networks and worldwide roaming.
- High transmission rates for data based services.
- Efficient bandwidth utilization schemes.

1.1.2 The Working of the Wireless Cellular System

In this section, we briefly explain how a wireless celluar network works [2]. In the rest of this section, a cell phone or any other device that can connect to a cellular radio network will be referred to as a mobile station in keeping with the literature on the subject.

A cellular network consists of both land and radio based sections. Such a network is commonly referred to as a PLMN - public land mobile network [3]. The network is composed of the following entities:

- Mobile station (MS): A device used to communicate over the cellular network.
- Base station transceiver (BST): A transmitter/receiver used to transmit/receive signals over the radio interface section of the network.
- Mobile switching center (MSC): The heart of the network which sets up and maintains calls made over the network.

- Base station controller (BSC): Controls communication between a group of BSTs and a single MSC.
- Public switched telephone network (PSTN): The land based section of the network.

Figure 1.1 illustrates how these entities are related to one another within the network. The BSTs and their controlling BSC are often collectively referred to as the base station subsystem (BSS). As explained before, the cellular topology of the network is a result of limited radio bandwidth. In order to use the radio spectrum efficiently, the same frequencies are reused in nonadjacent cells. A geographic region is divided up into cells. Each cell has a BST that transmits data via a radio link to MSs within the cell. A group of BSTs are connected to a BSC. A group of BSCs are in turn connected to a mobile switching center via microwave links or telephone lines. The MSC connects to the public switched telephone network, which switches calls to other mobile stations or land based telephones.

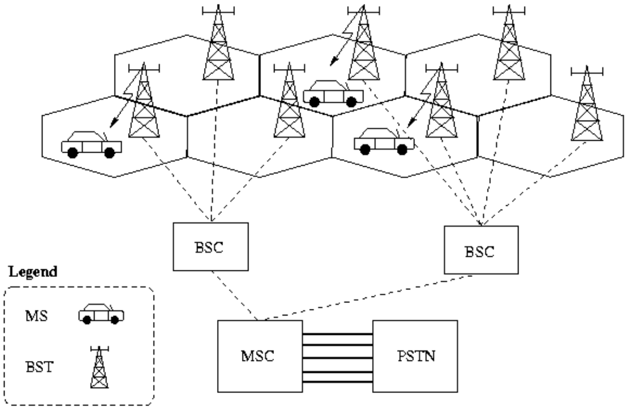


Figure 1.1: The components of a cellular network and their relation to each other

The following description of one mobile station placing a call to another mobile station best explains the underlying technology of a cellular network system: a mobile station places a call by sending a call initiation request to its nearest base station. This request is sent on a special channel, the reverse control channel (RCC). The base station sends the request, which contains the telephone number of the called party, to the MSC. The MSC validates the request and uses the number to make a connection to the called party via the PSTN. It first connects itself to the MSC of the called party, then the MSC instructs the base station and mobile station that placed the call to switch to voice channels. The mobile station that placed the call is then connected to the called station [7].

The steps explained above happen fast enough that the user does not experience any noticeable delay between placing a request for a call and the call being connected. The available frequency is accessed by different users in a cell using one of the two technologies mentioned in section 1.1.1: TDMA and CDMA. In the TDMA technology, a frequency band is divided into time slots. Each user gets the radio in this band all to itself for the entire time slot in which it is served. This is possible in 2G cellular systems (and wasn't in the 1G analog systems) because voice data that has been converted to digital information is compressed so that it takes up significantly less transmission space. The GSM standard uses TDMA for voice transfer. CDMA takes an entirely different approach to the use of the available bandwidth. After digitizing data, CDMA spreads it out over the entire available bandwidth. Multiple calls are overlaid on each other on the channel, with each assigned a unique sequence code (hence the "code" in CDMA). Thus data is sent in small packets over a number of discrete frequencies available for use at any time in the specified range.

As described in the beginning of this chapter, however, voice communication is not the only service sought by users. Data applications such as web browsing, downloading files from the internet, multimedia messaging are catching up fast and might soon overtake voice communication as the prime revenue generator. The 2G systems are inadequate to handle the requirements of such bandwidth intensive and rate sensitive data applications. The next generation of wireless cellular systems that support such applications and are currently the state-of-the-art in the industry (still being developed and deployed in newer markets) use the 3G technology.

1.2 Third Generation - High Speed Data Networks

The third generation, or 3G as it is popularly called, technology is the latest in mobile communications. 3G networks have potential transfer speeds of up to 3 Mbps (about 15 seconds to download a 3-minute MP3 song). For comparison, the fastest 2G phones can achieve up to 144Kbps (about 6 minutes to download a 3-minute song). 3G's high data rates are ideal for downloading information from the Internet and sending and receiving large, multimedia files. 3G phones are like mini-laptops and can accommodate broadband applications like video conferencing, receiving streaming video from the Web, sending and receiving faxes and instantly downloading e-mail messages with attachments. 3G comprises several cellular access technologies. The three most common are:

- WCDMA (UMTS) - Wideband Code Division Multiple Access; versatile and complicated implementation, hence technically challenging.
- TD-SCDMA - Time-division Synchronous Code-division Multiple Access; currently being developed by Chinese Academy of Telecommunications Technology and Siemens.
- CDMA2000 - primarily developed to work with existing 2G CDMA carriers; technology implemented by Qualcomm and currently offered by Verizon wireless and Sprint Nextel among almost seventy service providers all over the world.

In this thesis we analyze and attempt to enhance the performance of an implementation of the CDMA2000 technology called Evolution-Data Optimized (EV-DO). It has been adopted by many mobile phone service providers around the world, particularly, but not only, those previously employing CDMA networks. An EV-DO channel has a bandwidth of 1.25 MHz, the same bandwidth size as IS-95 [8]. The end user purchases an EV-DO modem (often referred to as an “aircard”) that receives the signal and allows connection to the internet. The possible download speeds vary from 38 Kbps to 2400 Kbps (or 3000 Kbps in a revised implementation of EV-DO) depending on the user conditions and distance from the base station. The back-end network is entirely packet-based and employs time multiplexing for data transfer. Thus, time is slotted (with a slot length of 1.67 milliseconds) and in every time slot each MS sends a pilot signal to the BST. The strength of the pilot signal depends on the distance of the MS from the base station, the terrain and other environmental conditions. There are eleven (or thirteen in the revised implementation of EV-DO) potentially available data rates to users for which the physical infrastructure is designed. Using the pilot signal the BST determines the rate at which data can be transmitted to a user if it is chosen to be served. The BST can serve at most one user in a time slot.

The first requirement of achieving such high speeds is better infrastructure that includes a combination of more available bandwidth, more powerful signals and taller base station towers. However, to be successful commerically the new technology should supplement the existing 2G systems and work within the existing cellular framework. Hence the mechanism of data transfer broadly remains the same - using the BST, MSC and the PSTN. One reason for the popularity of EV-DO among service providers is its complete backward compatibility with CDMA (IS-95) and the fact that it can be deployed alongside a wireless carrier's voice services. This makes EV-DO the most widely used 3G technology currently.

1.3 The Role of Scheduling

Electronics and communication engineers world over are thus working on faster, better (quality) and cheaper technology to cater to the growing data services market. However better engineering and technology alone do not ensure effective utilization of the available resources. We require efficient algorithms to ensure data flow from the base station, the fundamental transmission unit of cellular networks, to various users fairly and efficiently. In particular we require effective scheduling policies that determine which users should be served in a given time slot among all those present in the wireless cell. There are two conflicting objectives for the base station: maximize the data throughput per time slot, and maintain "fairness". To maximize the data throughput, the base station would like to serve the user with the highest available data rate, but this can lead to starvation of some users. To ensure "fairness", no user should be unserved for a "long" time, i.e., users should be served in a round-robin manner. This is the conflict that we seek to resolve in this thesis. The goal is to determine scheduling policies that address this conflict, are easy to implement and improve upon the policies that exist currently.

We focus on the downlink (base station to mobile) channel throughout, since in many applications such as web browsing, most of the data flow occurs in that direction. However all the ideas presented here can be applied to data transfer in the uplink direction as well.

1.4 Literature Review

The problem of scheduling users for data transmission in a wireless cell has been considered in the literature mostly in the last decade and a half. One of the most widely used algorithms that takes advantage of multiuser diversity (users having different and time-varying rates at which they can be served data) while at the same time being fair to all users is the PFA of Tse [9]. It is described in detail in section 2.2. In the setting of infinitely backlogged queues, the PFA performs well and makes good use of the multiuser diversity. This has been demonstrated in [10] where Jalali et al show using simulation that the throughout per cell of the wireless network increases as the number of users goes up. A drawback of using this algorithm, however, is its underlying assumption of unlimited data waiting to be served for each user. It has been proven to be unstable when there is external data arrival [11] for the users. This instability in the external data arrival regime is expected because the PFA doesn't take the data queue length of users into account while making the scheduling decision. Algorithms that consider the queue length in making the decision have also been proposed in the literature. Many such algorithms have the **Max-Weight Algorithm (MWA)** as their motivation. The MWA, variously referred to as the Differential Backlog algorithm, the Backpressure algorithm and the Load Balancing algorithm, was introduced by Awerbuch and Leighton in [12, 13] and by Tassiulas and Ephremides in [14, 15]. Tassiulas and Ephremides introduce the algorithm for a multihop radio network and prove its stability in [14]. They consider an extension to the case of randomly varying connectivity in the network in [15]. A significant amount of work has since appeared on proving the stability of algorithms similar to the MWA such as Neely et al [16]. They consider power allocation in a satellite that transmits data to different ground locations each having a channel of its own. Andrews, Jung and Stoylar [17] prove the stability of the MWA for dynamic networks. They consider the problem of combined packet routing and scheduling in communication networks that have high loads and can have dynamically varying connectivity along any edge of the network. The MWA has also been studied in other situations such as scheduling input-queued crossbar switches [18], load balancing tasks in a network of processors [19] and maximizing the total utility of traffic injected into the network [20].

All the above work assumes that the channel condition between the base station and the mobile user is governed by a stationary stochastic process such as an erdogic Markov chain. The MWA has been proved to be stable under this assumption in a variety of settings as described above. Andrews and Zhang [21] focus on scheduling algorithms that perform well for nonstationary wireless channels. They show that the MWA performs extremely poorly for nonstationary channels with an example of a setting under the standard EV-DO infrastructure where the MWA produces queues that are exponential in the number of users. They present the Quadratic Tracking algorithm in [22] and prove that the bound on the queue size for this algorithm is linear in the number of users. In [21] they improve the algorithm and show how it can be implemented in practice with some approximations where needed.

There also exists literature that deals in algorithms similar to the MWA for scheduling in wireless networks with time varying channel rates. Andrews et al [23] consider a variant of the MWA where they also take into account the head of the line packet delay in addition to each queue length. They call it the Modified Least Weighted Delay First (MLWDF) algorithm and prove its stability. Shakkottai and Stolyar [24] present another variant of the MWA that they call the Exponential Rule and prove its stability. We describe each of these algorithms including the MWA briefly, but precisely, in section 6.2. Further, in [25] they prove that in a heavy traffic limit and under some more conditions, the exponential rule minimizes $\max_u a_u Q_u(n)$ where a_u is some positive number, and $Q_u(n)$ is the queue length in time slot n of user u. A related area of reserch is "Generalized processor sharing" (GPS) where a multitude of users share the capacity of congested communications links in a fair manner. Mainly starting with the PhD dissertation of Parekh [26], there was tremendous activity and interest in this area [27, 28]. Several algorithms such as Weighted Fair Queuing [29], Start-time Fair Queuing [30] and Stochastic Fair Queuing [31] have been proposed to keep the queue lengths "'fair" for every user. In this setting, however, throughput is not a matter of concern since the available data rate for a user is bounded above only by the entire capacity of a link, and the data transmission rate does not change due to user mobility. In the wireless cell considered in this thesis, user mobility and the consequent multi-user diversity gain in throughput motivates us to look for alternative models and scheduling algorithms.

1.5 Our Contributions

We consider two types of cells according to user movement between cells. We first derive our scheduling policy assuming a fixed number of users in the cell, and no user movement in or out of the cell (the users are mobile within the cell). We call such a cell for which the number of users is fixed a "static cell". Then we extend the results of a static cell to the case when users can enter and leave the cell. We refer to such a cell with incoming and outgoing users as a "dynamic cell". As discussed in section 1.4, algorithms for scheduling in the static cell have been studied quite extensively in the wireless networks literature. We extend this work in two directions: First, the existing algorithms do not explicitly attempt to optimize any system wide objective function. We develop a Markov Decision Process (MDP) framework to find scheduling policies that optimize such an objective function. This objective function captures the conflict between throughput maximization and fairness to users effectively using appropriate costs and rewards. The policy improvement algorithm (PIA) is then used on the MDP formulation to derive "index policies" that involve computing an index for each user dependent on the current state of that user. In any given time slot the index policy serves the user with the highest index. The index has a sound theoretical basis, and we develop a closed form expression for it so that it is computationally efficient. We expect better performance from such a policy given its origins in a sound optimization framework and demonstrate the same through simulations. Secondly, we develop analytical results when possible and also look at simulations in the dynamic cell to get an insight into the performance of scheduling policies in a more realistic environment. To the best of our knowledge, there doesn't exist any study - numerical, simulation-based, or analytical - for dynamic cells.

The thesis is divided into two parts, each of which considers one of the two cases according to the data to be served: "infinitely backlogged" data queues and data queues fed by "externally arriving" data. The scheduling problem when there is always ample data to be served to every user is referred to as the "infinitely backlogged" queues case. This is a realistic assumption in the high congestion regime, where every user has high rate of data arrival and thus always has data to receive. The objective function that the MDP maximizes in this setting is the long run net reward per time slot. The net reward in each time slot is the reward accrued from data transmitted to the user scheduled to be served minus the penalty incurred by users that are not served. Thus our policy aims to strike a good balance between the two conflicting objectives: maximizing throughput and maintaining fairness. Part I of this thesis gives a precise account of these issues and the way we address them. Chapter 2 describes the model of underlying channel processes and develops an MDP model for the scheduling problem. Chapter 3 discusses some characteristics of the optimal policy of the MDP. Since the MDP is too complicated to be solved optimally, we derive several index policies to schedule users in chapter 4. These policies are based on the standard policy improvement method for Markov Decision Processes. The proportional fair algorithm, introduced in (PFA) [9], is currently used in practice in these settings. It has good "fairness" properties [32] but is not based on any systematic "optimization" procedure to maximize throughput. Therefore we expect that the policies we develop using our proposed methodology will exhibit better performance. We conclude part I with a performance analysis of our index policies and the PFA. We demonstrate through simulation that our index policies result in significantly better performance.

In Part II we consider the case where data arrives for every user randomly. In this case the throughput for every user is fixed in the steady state, and hence the goal of the MDP is to minimize the long run total weighted data queue length in a time slot across all users. We derive the scheduling policy as follows: we describe the model of underlying channel processes and develop an MDP model for the scheduling problem in Chapter 6. Chapter 7 discusses some characteristics of the optimal policy of the MDP. As in part I the MDP is too complicated to be solved optimally, hence we derive some index policies to schedule users in chapter 8. We conclude part II with a performance analysis of our index policies and an existing algorithm.

Part I

Infinitely Backlogged Queues

List of Notations for part I

(in the order of appearance)

N - Total number of users

u - Label for the users, $u = 1, 2, \ldots, N$

R_u^n - Channel rate of user u

R^n - The vector $[R_u^n : u = 1, 2, \ldots, N]$

Q_u - Exponentially filtered average data rate updated according to equation 2.1

τ - Key Parameter of the PFA, acts as damping coefficient in equation 2.1, PFA throughput increases as τ decreases

X_u^n - State of user u at time n

M - The number of states in the state space of the DTMC $\{X_u^n,\, n \geq 0\}$ for $u = 1, 2, \ldots, N$

P^u - The Transition probability matrix of the Markov chain $\{X_u^n,\, n \geq 0\}$; has elements $[p^u_{i_u, j_u}]$

$X^n = [X_1^n, \ldots, X_N^n]$ - State vector of all users

$i = [i_1, i_2, \ldots, i_N]$ - A realized value of the state vector X^n

Y_u^n - "Starvation age" (or "age") of the user u at time n

$Y^n = [Y_1^n, \ldots, Y_N^n]$ - The age vector at time n

$t = [t_1, t_2, \ldots, t_N]$ - A realized value of the age vector Y^n

$\mathcal{A}$ - Action space $\{1, 2, \ldots, N\}$ in any state (i, t)

$v(n)$ - User served in the n^{th} time slot

Ω - State space of the DTMC $\{X_u^n,\, n \geq 0\}$; is same for all users $u = 1, 2, \ldots, N$

r - A constant vector of data rates $= [r_1, r_2, \ldots, r_M]$; when $X_u^n = k$, $R_u^n = r_k$

$D_l(y)$ - Cost of not serving user l of age y in slot n

$V_T(i, t)$ - Optimal reward starting from state $[X^0, Y^0] = [i, t]$ at time 0 over time periods $0, 1, 2, \ldots, T$

g - The long run average throughput

$w(i, t)$ - Bias function starting in state (i, t)

q - Initial policy vector $= [q_1, q_2, \ldots, q_N]$

g_q - The constant g under the initial policy q

$w_q(i, t)$ - The bias function $w(i, t)$ under the initial policy q

$\pi^u = [\pi_1^u, \ldots, \pi_M^u]$ - Steady state distribution of the Markov chain $\{X_u^n : n \geq 0\}$

$\phi_u(q_u)$ - Long run cost per slot for user u under policy q

A_u - Mean reward earned by user u if served in every slot

K_u - Parameter of the LIP for user u so that $D_u(n) = K_u n$

$I_u(i, t)$ - The index for user u in state (i, t)

L_q - Lagrangian used to compute the optimal initial policy

θ - Lagrangian multiplier for optimizing L_q

B - Long run expected throughput per time slot

ζ - Long run expected starvation age of a user

ρ_d - Long run probability that a user is starved for more than d time slots

$\hat{B}$ - Estimate of B obtained from simulation

$\hat{\zeta}$ - Estimate of ζ obtained from simulation

$\hat{\rho_d}$ - Estimate of ρ_d obtained from simulation

K - The constant K_u assumed the same (K) for all users u

$N(t)$ - Number of users at time t in the cell in the dynamic case

λ - The arrival rate of users in the dynamic cell

a - Sojourn time of a user is exponentially distributed with mean a in a dynamic cell

Chapter 2

The Model

2.1 Motivation

We start by considering a fixed set of N mobile data users in a wireless cell served by a single base station. As noted in section 1.3 we focus on the downlink channel. The base station maintains a separate queue of data for each user. Time is slotted and in each time slot the base station can transmit data to exactly one user. Let R_u^n $(u = 1, 2, \ldots, N; n = 0, 1, \ldots)$ be the channel rate of user u during time slot n, i.e., the amount of data that can be transmitted to user u during time slot n by the base station. We assume that the base station knows at all time slots n a vector $R^n = [R_1^n, R_2^n, \ldots, R_N^n]$. How this information is gathered depends on the system in use. An example of a resource allocation system widely known and used in practice is the CDMA2000 1xEV-DO system [33] described briefly in section 1.2. A good description of how this information is generated is also provided in [33]. The algorithms presented in this chapter do not require the details of this mechanism of information transfer. We simply assume that $\{R^n, n \geq 0\}$ is a stochastic process that accounts for the random variation in data rates due to user mobility and other factors such as user terrain. A good framework for resource allocation and related issues in this (and more general) setting can be found in [34].

There are two objectives to be fulfilled while scheduling the data transfer. The first is to obtain a high data transfer rate. This can be achieved by serving a user u in slot n whose channel rate R_u^n is the highest, i.e., following a myopic policy. However if we follow the myopic policy, we run the risk of severely starving users whose channel rate is low for a long time.

The second objective is to ensure that none of the users is severely *starved*. Thus these are conflicting objectives and any good algorithm tries to achieve a “good” balance between the two. An algorithm that seeks to allocate resources to maximize system throughput under some Quality of Service (QoS) constraints that ensure a certain level of fairness to each user is presented in [35]. We shall comment on this algorithm later in section 5.4. The key features of the problem are a stochastic evolution of the data rate available to users, choosing a user to serve, a reward from serving that user in the form of data served, and a penalty in the form of unserved hence unsatisfied users. Markov Decision Processes (MDP) models are frequently used to determine optimal decisions (such as the user to serve) in a stochastic environment. Further we know that this type of reward structures can be easily incorporated into such models. A problem with using such a model is the curse of dimensionality if the model is to be solved to optimality. However previous experience [36] suggests that using only one step of the policy improvement approach can yield policies that are nearly optimal and do not suffer from this curse of dimensionality. Derivation of such policies does not need the solution to the corresponding high-dimensional MDP. This motivates us to develop our MDP and policy improvement based approach in this chapter. To our knowledge, no past work deals with this problem using an MDP based approach to derive implementable policies.

The rest of the chapter is organized as follows. In section 2.2 we describe a popular algorithm called Proportional Fair Algorithm (PFA) currently in use in the 1xEV-DO system. In section 2.3 we formulate the bandwidth allocation problem as an MDP under the assumption of a fixed number of users in the cell.

2.2 Proportional Fair Algorithm

The Proportional Fair Algorithm [32] is currently used in the 1xEV-DO system. In this part of the thesis we consider the case where every user has infinitely backlogged queues. This is a realistic assumption in heavy traffic regime, where every user always has data to receive. The PFA algorithm implicitly makes this assumption by not considering the current queue length in choosing which user to serve, and this makes comparison of our policy (derived with an assumption of infinitely backlogged queues) with PFA meaningful.

The PFA aims to optimize a given function of the throughput achieved by all the users. A commonly used objective function is the Proportional Fair metric $\sum_u \log Q_u$, where Q_u is a given measure of the long term throughput achieved by user u. A useful characteristic of this metric is that although it is strictly increasing in the throughput of each user, it prevents any user from being starved since $\log Q_u \to -\infty$ as $Q_u \to 0$. The PFA is characterized by a single constant $\tau \in (0,1)$ as explained below. Assume that there are a fixed number N of users in the cell. Let $v(n)$ be the user served in slot n. For each user u, define $Q_u(0) = 1$, and compute $Q_u(n)$ recursively as follows:

$$Q_u(n+1) = \begin{cases} (1-\tau)Q_u(n) + \tau R_u^n & \text{if } u = v(n) \\ (1-\tau)Q_u(n) & \text{if } u \neq v(n), \end{cases} \tag{2.1}$$

see [32]. Mathematically, τ acts as a damping coefficient and $Q_u(n)$ represents an exponentially filtered average service rate. The constant $t_c = \frac{1}{\tau}$ can also be taken to be a measure of the time a user can remain unserved [10]. Clearly $Q_u(n)$ represents an exponentially filtered average service rate. The PFA algorithm chooses to serve user $v(n)$ in slot n where

$$v(n) = \arg\max_u \frac{R_u^n}{Q_u(n)}. \tag{2.2}$$

It can be proved that this algorithm maximizes $\sum_u \log Q_u(n+1)$ - $\sum_u \log Q_u(n)$ for each n [32]. The PFA can also handle users of more than one type by assigning different values of τ to different types of users. It can be seen that users with higher values of τ will be served more frequently. In the next section we describe our MDP model to make this scheduling decision optimally.

2.3 Formulation as MDP

In this section we start with a stochastic model for $\{R^n,\, n \geq 0\}$ and formulate the scheduling problem as an MDP. Let X_u^n be the state of user u at time n. This represents all the various factors such as the position of the user in the cell, the propagation conditions etc that determine the data rate received by u in time slot n. We assume that $\{X_u^n,\, n \geq 0\}$ is an irreducible Discrete

Time Markov chain (DTMC) on state space $\Omega = \{1, 2, \ldots, M\}$ with Probability Transition Matrix (PTM) $P^u = [p^u_{i_u,j_u}]$. We make this assumption to make the analysis tractable [32]. For the sake of notational convenience, particularly in chapters 3 and 7, we assume, without loss of generality, that

$$r_1 \leq r_2 \leq \ldots \leq r_N \tag{2.3}$$

Further, as indicated in section 5.3.2, a set of $M = 11$ fixed data rates is what is available to users in an actual system [33]. Let r_k be the fixed data rate (or channel rate) associated with state $k \in \Omega$ of the DTMC. When $X^n_u = k$, the user u can receive data from the base station at rate $R^n_u = r_k$. Thus for all $u \in \{1, 2, \ldots, N\}$, the state space of the Markov chain $\{R^n_u : n \geq 0\}$ is $r = [r_1, r_2, \ldots, r_N]$. Let N be the total number of users in the cell (assumed constant in this section) and let $X^n = [X^n_1, \ldots, X^n_N]$ be the state vector of all the users. Since each component of $\{X^n, n \geq 0\}$ is an independent DTMC on Ω, it is clear that $\{X^n,\ n \geq 0\}$ itself is a DTMC on Ω^N. We assume the users behave independently of each other and that each user has ample data to transmit.

Let Y^n_u be the "starvation age" (or simply "age") of the user u at time n, defined as the time elapsed (in number of slots) since the user u was served most recently. Thus, the age of the user is zero at time $n+1$ if it is served in the n^{th} time slot. Furthermore, for $m \geq 1$, if the user was served in time slot n and it is not served for the next m time slots, its age at time $n+m$ is $m-1$. Let $Y^n = [Y^n_1, \ldots, Y^n_N]$ be the age vector at time n. The base station serves exactly one user in each time slot. In this and the following sections let $v(n)$ be the user served in the n^{th} time slot. It should be noted here that the expression for $v(n)$ given by (2.2) is used only for the PFA, it does not define $v(n)$ as used in this section. The age variables change according to

$$Y^{n+1}_u = \begin{cases} Y^n_u + 1 & \text{if } u \neq v(n) \\ \\ 0 & \text{if } u = v(n) \end{cases} \tag{2.4}$$

The "state of the system" at time n is given by $[X^n, Y^n] \in \Omega^N \times Z^N$, where $Z = \{0, 1, 2, \ldots\}$. The "state" is thus a vector of $2N$ components and we assume that it is known at the base station in each time slot. After observing $[X^n, Y^n]$ the base station decides to serve one of the

N users in the time slot n. We need a reward structure in order to make this decision optimally. We describe such a structure below. If we serve user u in the n^{th} time slot, we earn a reward of $R_u^n = r_{X_u^n}$ for this user and none for the others. In addition, there is a cost of $D_l(y)$ if user l of age y is not served in slot n. Clearly, we can assume $D_l(0) = 0$ since there is no starvation at age zero. The net reward of serving user u at time n is

$$R_u^n - \sum_{l \neq u} D_l(Y_l^n). \tag{2.5}$$

We assume that there is no cost in switching from one user to another from slot to slot. This is not entirely true in practice, but including switching costs in the model will make the analysis intractable. For convenience we use the notation

$$W_u^n = \sum_{l \neq u} D_l(Y_l^n).$$

The problem of scheduling a user in a given time slot can now be formulated as a Markov Decision Process (MDP). The decision epochs are $\{1, 2, \ldots\}$. The state at time n is $[X^n, Y^n]$ with Markovian evolution as described above. The action space in every state is $\mathcal{A} = \{1, 2, \ldots, N\}$ where action u corresponds to serving the user u. The reward in state $[X^n, Y^n]$ corresponding to action u is $R_u^n - W_u^n$. Let the transition probability under action u from (i, s) to (j, t) ($i, j \in \Omega^N$ and $s, t \in Z^N$) be denoted by $p((j,t)|(i,s),u)$. It is given by

$$p((j,t)|(i,s),u) = \begin{cases} p^1_{i_1,j_1} p^2_{i_2,j_2} \cdots p^N_{i_N,j_N} = p_{ij}, \\ \text{if } t_u = 0 \text{ and } t_l = s_l + 1 \text{ for } l \neq u \\ \\ 0 \quad \text{otherwise.} \end{cases} \tag{2.6}$$

The state space of this MDP, i.e., $\Omega^N \times Z^N$, is very large which makes the derivation of the optimal policy extremely hard and unusable in practice. The advantage of our analysis is that it avoids having to solve the MDP equations. Instead it uses just one step of the policy improvement algorithm that avoids the curse of dimensionality and produces simple scheduling policies that have minimal parameter requirements and are unaffected by the size of this state space. Thus we see that although we start with formulating the problem as an MDP, our final index policy is independent of the transition probabilities $p^u_{i_u,j_u}$ (u=1,2,...,N). We do however use $p^u_{i_u,j_u}$ to simulate the underlying Markov chain to do a performance analysis of our proposed policy and the arguments used to obtain the value of $p^u_{i_u,j_u}$ are described in detail in section 5.3.2. The method described in [37] can also be used to estimate $p^u_{i_u,j_u}$ based on engineering considerations, if needed.

Let $V_T(i,t)$ be the optimal reward starting from state $[X^0, Y^0] = [i,t]$ at time 0 over time periods $0, 1, 2, \ldots, T$. If user u is served at time 0, the age vector in the next time slot is given by

$$t^u = (t_1 + 1, \ldots, t_{u-1} + 1, 0, t_{u+1} + 1, \ldots, t_N + 1). \tag{2.7}$$

We also define

$$W_u(t) = \sum_{l \neq u} D_l(t_l). \tag{2.8}$$

A standard Dynamic Programming (DP) argument then yields the following Bellman equation

$$V_T(i,t) = \max_{u=1,2,\ldots,N} \left[r_{i_u} - W_u(t) + \sum_j p_{ij} V_{T-1}(j, t^u) \right], \tag{2.9}$$

where p_{ij} is as defined in (2.6). We wish to determine the action $u = u(i,t)$ that maximizes $\lim_{T\to\infty} V_T(i,t)/T$, i.e., the long run average reward. It is well known [38] that such a policy $\{u(i,t) : (i,t) \in \Omega^N X Z^N\}$ exists if there is a constant g (also called the gain) and a bias function $w(i,t)$ satisfying

$$g + w(i,t) = \max_u \{ r_{i_u} - W_u(t) + \sum_j p_{ij} w(j, t^u) \}. \tag{2.10}$$

The intuitive explanation of g and the bias function will be made clear in equation 4.3 of section 4.2. Here we end with the result that any u that maximizes $r_{i_u} - W_u(t) + \sum_j p_{ij} w(j, t^u)$ over all $u \in \{1, \ldots, N\}$ is an optimal action $u(i,t)$ in state (i,t).

Chapter 3

Monotonicity of the Optimal Policy

We discussed in chapter 2 that solving the MDP to optimality is infeasible. However, we can derive some important characteristics of the optimal policy. In this chapter, we consider two monotonicity properties of the optimal policy. We will see in chapter 4 that our suggested index policies too possess these basic properties of the optimal policy.

3.1 Monotonicity in Age

The penalty accrued for each user in a given time slot is an increasing function of its current age. Hence we expect the likelihood of the optimal policy serving any given user to increase with its age, i.e., if the optimal policy serves a user u in the state $[i, t]$, it will serve user u in state $[i, t + e_u]$ as well, where e_u denotes an N-dimensional vector with the u^{th} component 1 and all other components 0. Theorem 3.1.2 states and proves this monotonicity of the optimal policy for discounted reward. Then we show that standard MDP theory [38] implies the result holds in the case of average reward as well. We use the following notation: For any real valued function $f(i, t)$ defined on $\Omega^N \times Z^N$, $f \downarrow t$ denotes that f decreases in every component of t.

Let $V_\alpha(i, t)$ be the total discounted reward with a discounting rate α starting in state $[i, t]$. In the rest of this section and section 3.2, we drop the subscript α from $V_\alpha(\cdot, \cdot)$ for notational convenience. Then following equation 2.9, the standard Bellman equation for the discounted reward model is

$$V(i, t) = \max_{u=1,2,\ldots,N} \left[r_{i_u} - W_u(t) + \alpha \sum_j p_{ij} V(j, t^u) \right] \tag{3.1}$$

Equivalently, standard value iteration equations of (3.1) are given by

$$V_{k+1}(i,t) = \max_{u=1,2,\dots,N} \left[r_{i_u} - W_u(t) + \alpha \sum_j p_{ij} V_k(j, t^u) \right], \qquad k \geq 0. \tag{3.2}$$

For notational convenience, let

$$\begin{aligned} \sum_j p_{ij} V(j,t) &= h(i,t), \\ \sum_j p_{ij} V_k(j,t) &= h_k(i,t), \end{aligned} \tag{3.3}$$

yielding

$$\begin{aligned} V(i,t) &= \max_{u=1,2,\dots,N} \left[r_{i_u} - W_u(t) + \alpha h(i,t^u) \right], \\ V_{k+1}(i,t) &= \max_{u=1,2,\dots,N} \left[r_{i_u} - W_u(t) + \alpha h_k(i,t^u) \right], \qquad k \geq 0. \end{aligned} \tag{3.4}$$

Let $dec(i,t) \in \mathcal{A}$ be the optimal decision made (i.e., the user served) in state $[i,t]$. Then, $dec(i,t) = \arg\max_{u=1,2,\dots,N} \left[r_{i_u} - W_u(t) + \alpha h(i,t^u) \right]$. Further, let

$$dec_k(i,t) = \arg \max_{u=1,2,\dots,N} \left[r_{i_u} - W_u(t) + \alpha h_k(i,t^u) \right] \tag{3.5}$$

be the optimal decision at the k^{th} step of the value iteration scheme given by (3.2).

We will need the following result to prove theorem 3.1.2.

Theorem 3.1.1. $V(i,t) \downarrow t$

Proof. Following standard methods in MDP theory [38], we can choose $V_0(i,t) = 0$ for all $[i,t] \in \Omega^N \times Z^N$ to initialize the value iteration equations of (3.2). Therefore, $h_0(i,t) \downarrow t$. We will prove the theorem using induction on k. Assume $h_k(i,t) \downarrow t$ for some $k \geq 0$. This induction hypothesis holds at $k = 0$. Under this assumption, we prove $V_{k+1}(i,t) \downarrow t$. It is enough to prove that

$$V_{k+1}(i,t) - V_{k+1}(i, t + e_1) \geq 0, \tag{3.6}$$

since the proof for all components other than 1 follows similarly. We consider four cases:

Case 1: $dec_k(i,t) = 1$ and $dec_k(i, t + e_1) = 1$. From (3.4),

$$\begin{aligned} &V_{k+1}(i,t) - V_{k+1}(i, t+e_1) \\ &= [r_{i_1} - W_1(t) + \alpha h_k(i, t^1)] - [r_{i_1} - W_1(t+e_1) + \alpha h_k(i, (t+e_1)^1)] \\ &\geq 0, \end{aligned} \tag{3.7}$$

since $W_v(t + e_v) = W_v(t)$ and $(t + e_v)^v = t^v$ using equations 2.7 and 2.8.

Case 2: $dec_k(i,t) = 1$ and $dec_k(i, t+e_1) = u \neq 1$. From (3.4), and using $W_u(t+e_1) \geq W_u(t)$ and $(t + e_1)^u = t^u + e_1$, we have

$$\begin{aligned} &V_{k+1}(i,t) - V_{k+1}(i, t+e_1) \\ &= [r_{i_1} - W_1(t) + \alpha h_k(i, t^1)] - [r_{i_u} - W_u(t+e_1) + \alpha h_k(i, (t+e_1)^u)] \\ &\geq [r_{i_1} - r_{i_u}] + [W_u(t) - W_1(t)] + \alpha \left[h_k(i, t^1) - h_k(i, (t^u + e_1))\right] \\ &\geq [r_{i_1} - r_{i_u}] + [W_u(t) - W_1(t)] + \alpha \left[h_k(i, t^1) - h_k(i, t^u)\right] \\ &\geq 0. \end{aligned} \tag{3.8}$$

The last inequality holds because $dec_k(i,t) = 1$.

Case 3: $dec_k(i,t) = u \neq 1$ and $dec_k(i, t + e_1) = u$. From (3.4),

$$\begin{aligned} &V_{k+1}(i,t) - V_{k+1}(i, t+e_1) \\ &= [r_{i_u} - W_u(t) + \alpha h_k(i, t^u)] - [r_{i_u} - W_u(t+e_1) + \alpha h_k(i, (t+e_1)^u)] \\ &\geq [W_u(t+e_1) - W_u(t)] + \alpha \left[h_k(i, t^u) - h_k(i, (t^u + e_1))\right] \\ &\geq 0. \end{aligned} \tag{3.9}$$

Case 4: $dec_k(i,t) = u \neq 1$ and $dec_k(i, t+e_1) = v$. From (3.4), and using $W_v(t+e_1) \geq W_v(t)$ and $(t + e_1)^v = t^v + e_1$, we have

$$\begin{aligned}
&V_{k+1}(i,t) - V_{k+1}(i,t+e_1) \\
&= [r_{i_u} - W_u(t) + \alpha h_k(i,t^u)] - [r_{i_v} - W_v(t+e_1) + \alpha h_k(i,(t+e_1)^v)] \\
&\geq [r_{i_u} - r_{i_v}] + [W_v(t) - W_u(t)] + \alpha\,[h_k(i,t^u) - h_k(i,(t^v+e_1))] \qquad (3.10) \\
&\geq [r_{i_u} - r_{i_v}] + [W_v(t) - W_u(t)] + \alpha\,[h_k(i,t^u) - h_k(i,t^v)] \\
&\geq 0.
\end{aligned}$$

The last inequality holds because $dec_k(i,t) = u$.
Clearly, cases 1-4 are exhaustive and thus equations 3.7 through 3.10 prove that $V_{k+1}(i,t) \downarrow t$. From equation 3.3, $V_{k+1}(i,t) \downarrow t \Longrightarrow h_{k+1}(i,t) \downarrow t$, thus completing our induction argument. Since $V_k(i,t) \downarrow t$ for each $k \geq 0$ and $V_k(i,t) \to V(i,t)$ as $k \to \infty$, we have

$$V(i,t) \downarrow t, \qquad (3.11)$$

as required. □

Now we move on to the main theorem of this chapter that says that the decision to serve a user in any time slot is monotone in age.

Theorem 3.1.2. *$dec(i,t) = v \Longrightarrow dec(i,t+e_v) = v$.*

Proof. Since $dec(i,t) = v$ we have,

$$r_{i_v} - W_v(t) + \alpha h(i,t^v) \geq r_{i_u} - W_u(t) + \alpha h(i,t^u), \qquad u \in \mathcal{A}. \qquad (3.12)$$

To prove $dec(i,t+e_v) = v$, we need to prove

$$[r_{i_v} - r_{i_u}] + [W_u(t+e_v) - W_v(t+e_v)] + \alpha\,[h(i,(t+e_v)^v) - h(i,(t+e_v)^u)] \geq 0, \qquad (3.13)$$

which follows from (3.12), and the results that $W_v(t+e_v) = W_v(t)$, $W_u(t+e_v) \geq W_u(t)$ (using equation 2.8) and $h(i,(t+e_v)^u) < h(i,t^u)$ (using theorem 3.1.1 and equation 3.3). □

3.2 Monotonicity in Rate

The MDP model in chapter 2 has been formulated to maximize the long term net reward. The net reward over one time slot in a given state $[i, t]$ equals the data rate of the user that is chosen to serve minus the penalty accrued by all other users. We expect the optimal policy to be monotone in the rate that can be potentially available to the users. In particular, we expect that if the optimal policy serves user v in state $[i, t]$, then it will serve v in state $[i + e_v, t]$ as well. We prove this in theorem 3.2.1 under the assumption that $\{X^n : n \geq 0\}$ are i.i.d. Let $f(\cdot)$ be the probability mass function of the environment state vector $X \in \Omega^N$.

Theorem 3.2.1. *Suppose $\{X^n : n \geq 0\}$ are i.i.d. and $v \in \mathcal{A}$ is fixed. Then $dec(i, t) = v \implies dec(i + e_v, t) = v$.*

Proof. Since $\{X^n : n \geq 0\}$ are i.i.d., we get

$$h(i, t) = \sum_{j : j \in \Omega^N} f(j) V(j, t) \tag{3.14}$$

Following (2.9), the value function $V(i, t)$ under the optimal policy is given by

$$V(i, t) = \max_{u=1,2,\ldots,N} \left[r_{i_u} - W_u(t) + \alpha h(i, t^u) \right] \tag{3.15}$$

Hence for $u \in \mathcal{A}$

$$dec(i, t) = v \implies [r_{i_v} - r_{i_u}] + [W_u(t) - W_v(t)] + \alpha \left[h(i, t^v) - h(i, t^u) \right] \geq 0. \tag{3.16}$$

To prove $dec(i + e_v, t) = v$, we need to prove

$$[r_{i_v+1} - r_{i_u}] + [W_u(t) - W_v(t)] + \alpha \left[h(i + e_v, t^v) - h(i + e_v, t^u) \right] \geq 0, \tag{3.17}$$

which follows from (3.14) and (3.16) since $h(i, t)$ is independent of i. □

However, if $\{X^n : n \geq 0\}$ is a DTMC, the above proof does not work. A key step in the proof above is

$$h(i + e_v, t^v) - h(i + e_v, t^u) = h(i, t^v) - h(i, t^u). \tag{3.18}$$

Correspondingly, in the DTMC case, we require

$$h(i+e_v, t^v) - h(i+e_v, t^u) \geq h(i, t^v) - h(i, t^u). \tag{3.19}$$

We expect (3.19) to hold only when for each $u \in \mathcal{A}$, the Markov chain $\{X_u^n : n \geq 0\}$ possesses a special property called stochastic monotonicity [39]. Let $f(\cdot, \cdot)$ be a function defined on $\Omega^N \times Z^N$ and $f(i,t) \uparrow i$ denotes that $f(\cdot, \cdot)$ increases in every component of i. The main consequence of stochastic monotonicity of $\{X_u^n : n \geq 0\}$ is the result that

$$V(i,t) \uparrow i \Longrightarrow h(i,t) \uparrow i. \tag{3.20}$$

Although we are unable to furnish a complete proof of the monotonicity in rate of the optimal policy for a Markovian evolution of the packet arrival process $\{A^n : n \geq 0\}$, from our analysis we expect (3.20) to be a necessary condition for the rate monotonocity (3.21).

Now we state the monotonocity of decisions in rate formally for the total discounted reward case in Conjecture 3.2.2.

Conjecture 3.2.2. *If the Markov chain $\{X^n : n \geq 0\}$ is stochastically monotone, then*

$$dec(i,t) = v \Longrightarrow dec(i+e_v, t) = v \tag{3.21}$$

3.3 Average Reward Criterion

In this section we extend the results of sections 3.1 and 3.2 to the average reward criterion. Define a subset S of the state space $\Omega^N \times Z^N$ by

$$S = \{[i,t] \in \Omega^N \times Z^N : t_u \neq t_v, \quad u, v = 1, 2, \ldots, N\} \tag{3.22}$$

Consider any stationary policy $\{f(i,t) : \Omega^N \times Z^N \mapsto \mathcal{A}\}$ of the original MDP introduced in section 2.3. Let $\{(X^n, Y^n) : n \geq 0\}$ be the DTMC induced by f. Then we have the following lemma.

Lemma 3.3.1. *S is a closed communicating class of $\{(X^n, Y^n) : n \geq 0\}$.*

Proof. Let $(X^n, Y^n) \in S$ for some $n \geq 0$. Since $\{Y^n : n \geq 0\}$ evolves according to (2.4) and we serve exactly one user in every time slot, $[X^{n+1}, Y^{n+1}] \in S$. Further, since $\{X^n : n \geq 0\}$ is a finite and irreducible DTMC, S is closed and communicating, as required. □

We note that as a result of lemma 3.3.1 and the evolution of the age vector $\{Y^n : n \geq 0\}$, any state $[i,t] \in \left(\Omega^N \times Z^N\right) \backslash S$ is transient. Therefore, we restrict ourselves to proving monotonicity of the optimal policy on S. Let $[w(i,t) : (i,t) \in S]$ be the bias vector satisfying (2.10). To prove that the monotonicity in age is valid (over S) for the average reward criterion, we need to prove that for $[i,t] \in S$,

$$\begin{aligned} & r_{i_v} - W_v(t) + \sum_j p_{ij} w(j, t^v) \geq r_{i_u} - W_u(t) + \sum_j p_{ij} w(j, t^u) \Longrightarrow \\ & r_{i_v} - W_v(t+e_v) + \sum_j p_{ij} w(j, (t+e_v)^v) \geq r_{i_u} - W_u(t+e_v) + \sum_j p_{ij} w(j, (t+e_v)^u). \end{aligned} \tag{3.23}$$

To do this, we choose a fixed integer T and for each $u \in \mathcal{A}$ set

$$D_u(t) = \infty, \quad t > T, \quad u \in \mathcal{A}. \tag{3.24}$$

Now, consider two systems:

- ***System A*:** The MDP model described in section 2.3 with state space restricted to S and with the extra condition (3.24).
- ***System A′*:** Identical to System A except that any user with age T has to be served. Therefore, the state space of this system is finite and is given by

  $$S' = \{[i,t] \in S : t_u \leq T, u \in \mathcal{A}\}, \tag{3.25}$$

  and the transition probabilities, reward structure are the same as that of System A. Clearly, as $T \uparrow \infty$, $S' \uparrow S$

Our goal is to prove that even for the average reward criterion, the optimal policy is monotone in age in System A. We will show in theorem 3.3.2 that the monotonicity in age for the average reward criterion holds for all fixed T in System A'. Further, since Systems A and A'

are equivalent in the total optimal discounted reward sense of (3.31), we will conclude that monotonicity in age for the average reward criterion holds for System A.

Theorem 3.3.2. *The optimal policy for the average reward criterion is monotone in age in System A', i.e. for $[i,t] \in S'$*

$$dec(i,t) = v \Longrightarrow dec(i,t+e_v) = v. \tag{3.26}$$

Proof. Consider System A'. The state space S' is finite and using (2.5), the one step reward is bounded below by $C_L = r_1 - ND(T)$ and above by $C_U = r_N$. Thus the absolute value of the one step reward is bounded by $F = \max\{|C_L|, C_U\}$. Let $V_\alpha^{'}(i,t)$ be the optimal total discounted reward of System A' starting in state $[i,t] \in S'$. Then $V_\alpha^{'}(i,t)$ satisfies the standard Bellman equation given by (3.1). Using results in chapter 3 of [40], for a fixed $[k,m] \in S'$,

$$|V_\alpha^{'}(i,t) - V_\alpha^{'}(k,m)| < C < \infty \quad \text{for } [i,t] \in S', \tag{3.27}$$

where C is a positive constant. Then from Ross [41], there exists a constant g' and bias function $w'(i,t)$ satisfying (2.10) and given by

$$\begin{aligned} g' &= \lim_{\alpha \to 1} \left[V_\alpha^{'}(k,m)(1-\alpha) \right] \\ w'(i,t) &= \lim_{\alpha \to 1} \left[V_\alpha^{'}(i,t) - V_\alpha^{'}(k,m) \right]. \end{aligned} \tag{3.28}$$

Theorem 3.1.2 implies that

$$\begin{aligned} & r_{i_v} - W_v(t) + \alpha \sum_j p_{ij} V_\alpha^{'}(j, t^v) \geq r_{i_u} - W_u(t) + \alpha \sum_j p_{ij} V_\alpha^{'}(j, t^u) \Longrightarrow \\ & r_{i_v} - W_v(t+e_v) + \alpha \sum_j p_{ij} V_\alpha^{'}(j, (t+e_v)^v) \geq r_{i_u} - W_u(t+e_v) + \alpha \sum_j p_{ij} V_\alpha^{'}(j, (t+e_v)^u). \end{aligned} \tag{3.29}$$

Subtracting $V_\alpha^{'}(k,m) = \sum_j p_{ij} V_\alpha^{'}(k,m)$ on both sides of both the inequalities in (3.29) and taking the limit as $\alpha \to 1$, we get

$$\begin{aligned} & r_{i_v} - W_v(t) + \sum_j p_{ij} w'(j, t^v) \geq r_{i_u} - W_u(t) + \sum_j p_{ij} w'(j, t^u) \Longrightarrow \\ & r_{i_v} - W_v(t+e_v) + \sum_j p_{ij} w'(j, (t+e_v)^v) \geq r_{i_u} - W_u(t+e_v) + \sum_j p_{ij} w'(j, (t+e_v)^u), \end{aligned} \tag{3.30}$$

using (3.28). Equation 3.30 implies (3.26), as required. □

Thus the optimal policy of System A' is monotone in age for every T. Let $V_\alpha(i,t)$ be the optimal total discounted reward of System A starting in state $[i,t] \in S$. From the definition of Systems A and A', it is clear [40] that

$$V_\alpha^{'}(i,t) = V_\alpha(i,t) \text{ for } [i,t] \in S'. \tag{3.31}$$

From equations 3.31 and 3.27 through 3.30 it is clear that System A is monotone in age over S' constructed using any fixed T. Since $S' \uparrow S$ as $T \uparrow \infty$, we can conclude that the optimal policy of the MDP introduced in section 2.3 is monotone in age over S for the average reward criterion.

Partial results for the monotonicity in rate of the optimal policy for the total discounted reward case of the MDP of section 2.3 are presented in theorem 3.2.1 and conjecture 3.21. Using results similar to the results mentioned above in this section, we can show that the rate monotonicity continues to hold for the average reward criterion.

Now, since solving the MDP to optimality is infeasible, we derive a heuristic policy based on the policy improvement algorithm. Such policies have been termed “index policies” in the literature [42, 43] for reasons that become apparent in chapter 4. As we expect from any reasonable policy that acts as a surrogate for the optimal policy, the index policy is monotone in rate as well as age in the sense described in this chapter.

Chapter 4

Index Policy

We develop the index policy as an approximation to the optimal policy using one step of policy improvement algorithm. We give an overview of our approach in the next section.

4.1 Policy Improvement Approach

In this chapter we use a policy-improvement approach to develop a heuristic scheduling policy. We first describe the standard policy improvement algorithm [38].

1. Let π^0 be an arbitrary policy that chooses action $\pi^0(i,t) \in \mathcal{A}$ in state (i,t). Set $n = 0$.

2. Policy Evaluation Step: Solve the equations

$$g_n + w_n(i,t) = r_{i_u} - W_u(t) + \sum_j p_{ij} w_n(j, t^u), \ (i,t) \in \Omega^N X Z^N$$

   for g_n and $\{w_n(i,t), i \in \Omega^N, t \in Z^N\}$ where $u = \pi^n(i,t)$ and n denotes the number of iterations so far.

3. Policy Improvement Step: Let

$$\pi^{n+1}(i,t) = \arg\max_{u \in \mathcal{A}} \{r_{i_u} - W_u(t) + \sum_j p_{ij} w_n(j, t^u)\}. \tag{4.1}$$

   If $\pi^n(i,t)$ maximizes the Right Hand Side (RHS), choose $\pi^{n+1}(i,t) = \pi^n(i,t)$.

4. If $\pi^{n+1} \neq \pi^n$, set $n = n + 1$ and go to step 2. Else, STOP. π^{n+1} is the optimal policy.

Under certain conditions [38], one can show that this algorithm terminates in a finite number of steps.

Next we describe how a heuristic policy can be developed using just one policy improvement step. It should be noted here that applying the standard policy improvement method that involves using the policy improvement step multiple times until a terminal condition is satisfied is not feasible in our problem because of the large state space. Further, as we see in the numerical results, for a wide range of the model parameters the throughput using just one policy improvement step yields a throughput close to the maximal throughput possible (obtained using the myopic policy).

This suggests using one policy improvement step alone suffices to get a policy close to the optimal policy. In each time slot, given the state (i, t) of the process, the aim is to compute for each user u an index (i.e., a real number) that depends solely on the current state (i_u, t_u) of that user. The heuristic scheduling policy then serves the user with the maximum index in each time slot. Such policies are referred to as Index Policies [44, 45, 36] and as we shall see in the context of this problem, perform very well. A major contribution of this thesis is the derivation of a closed form expression of the index for each user. Further, although this index will be developed under our current assumptions of Markovian evolution of the system and a constant number of users, we will see that our Index policy does not use the parameters of the Markovian structure. The method of developing such an index policy involves choosing an "appropriate" initial policy and modifying it by a single step of policy improvement algorithm of the MDP. We discuss an appropriate initial policy in the next subsection.

4.2 Initial Policy

Consider a stationary state-independent policy that serves user u with probability q_u in any time slot. Here $q_1, \ldots, q_N$ are fixed numbers such that $q_u > 0$, $\sum_u q_u = 1$. Note that $q_u, u = 1, 2, \ldots, N$ only give us the initial policy that we use to formulate the ultimate index policy. Let

$$q = [q_1, q_2, \ldots, q_N],$$

g_q be the long run reward and

$$w_q = \{w(i,t), (i,t) \in \Omega^N X Z^N\}$$

be the bias vector for this policy satisfying the equation

$$g_q + w_q(i,t) = \sum_{u=1}^{N} q_u \left\{ r_{i_u} - W_u(t) + \sum_j p_{ij} w_q(j, t^u) \right\}. \tag{4.2}$$

We also refer to this initial policy as the *randomized policy* or the *policy* q. Let $V_T^q(i,t)$ be the total reward in periods 0 through T starting in state (i,t) corresponding to this initial policy q. We use $o(T)$ to denote terms that go to zero as T approaches ∞. Then standard MDP theory [38] yields

$$V_T^q(i,t) = g_q T + w_q(i,t) + o(T). \tag{4.3}$$

Thus for a fixed policy q and state (i,t) one can think of $V_T^q(i,t)$ as an approximately linear function of T with slope g_q and intercept $w_q(i,t)$.

The computation of g_q is fairly straightforward and we give the main result in theorem 4.2.1. This result will be used in subsection 4.4 for choosing the optimal q_u, $u = 1, \ldots, N$. Computing $w_q(i,t)$ is not simple. However, as we shall see in the next subsection, we do not need to compute $w_q(i,t)$ per se, but only the difference $w_q(j,t^u) - w_q(j, t+e)$, where $e = (1, \ldots, 1) \in Z^N$.

Let $\pi^u = [\pi_1^u, \ldots, \pi_M^u]$ be the steady state distribution of the Markov chain $\{X_u^n : n \geq 0\}$, $u = 1, 2, \ldots, N$. Then, since M is finite and the DTMC is irreducible, it is well known [46] that π^u exists and is the unique solution to

$$\begin{aligned} \pi^u &= \pi^u P^u \\ \sum_{m=1}^{M} \pi_m^u &= 1. \end{aligned}$$

We define

$$\phi_u(q_u) = \sum_{k=1}^{\infty} D_u(k)(1-q_u)^k q_u, \tag{4.4}$$

and

$$A_u = \sum_m \pi_m^u r_m. \tag{4.5}$$

Thus $\phi_u(q_u)$ is the long run cost per slot for user u under the randomized policy and A_u denotes the mean reward earned by user u if he is served in every slot. Then, the following theorem gives an expression for g_q.

Theorem 4.2.1. *We have*

$$g_q = \sum_{u=1}^{N} [-\phi_u(q_u) + A_u q_u]. \tag{4.6}$$

Proof. Since all the users are independent, the total average net reward g_q is just the sum of the average net reward accrued to each user. We first derive the total average reward earned and then the total average cost incurred. Consider user u. In each time slot user u is served with probability q_u. Clearly, the total average reward earned per unit time in steady state is

$$\sum_{u=1}^{N} A_u q_u. \tag{4.7}$$

Note that the age process $\{Y_u^n : n \geq 0\}$ is a regenerative process with $Y_u^n = 0$ as the regeneration point. Let S_u denote the length of a regenerative cycle and C_u the cost accrued in one regenerative cycle. Then, average cost is given by [46]

$$\frac{\mathrm{E}[C_u]}{\mathrm{E}[S_u]}. \tag{4.8}$$

Now the randomized policy implies that

$$\Pr[S_u = k] = (1 - q_u)^{k-1} q_u, \tag{4.9}$$

for $k = 1, 2, \ldots$ and hence

$$\mathrm{E}[S_u] = \frac{1}{q_u}. \tag{4.10}$$

Recall that $D_u(0) = 0$. Using $C_u = \sum_{k=0}^{S_u - 1} D_u(k)$, we get

$$\mathrm{E}[C_u] = \sum_{k=2}^{\infty} [D_u(1) + \ldots + D_u(k-1)](1 - q_u)^{k-1} q_u$$

which can be simplified to

$$\mathrm{E}[C_u] = \sum_{k=1}^{\infty} D_u(k)(1-q_u)^k. \tag{4.11}$$

Using (4.4), (4.8), (4.10) and (4.11), the total expected starvation cost per unit time in steady state is

$$\sum_{u=1}^{N} \phi_u(q_u) \tag{4.12}$$

yielding (4.6) as the total net reward per unit time in steady state. □

Corollary 4.2.2. *Suppose $D_u(n) = K_u n$ for $n \geq 0$. Then $\phi_u(q_u)$ is given by*

$$\phi_u(q_u) = \frac{(1-q_u)K_u}{q_u}. \tag{4.13}$$

Proof. Putting $D_u(t_u) = K_u t_u$ in (4.4), we get

$$\phi_u(q_u) = K_u \sum_{k=1}^{\infty} k(1-q_u)^k q_u,$$

which reduces to (4.13), as required. □

We thus have an initial randomized scheduling policy based on the probability vector q (defined above) and an expression for the long run reward per time slot for such a policy. In the next section we apply the policy improvement step (step 3 in the policy improvement algorithm described in section 4.1) once to obtain a better policy. We see later for the examples considered in section 5.3 that even this one-step improved policy, for appropriate values of the parameters K_u, gives us throughput that is close to the maximum possible throughput.

4.3 Policy Improvement Step

For a given (i, t) the policy improvement step seeks to maximize

$$r_{i_u} - W_u(t) + \sum_j p_{ij} w_q(j, t^u)$$

over all $u \in \mathcal{A}$. This is equivalent to maximizing

$$I_u(i,t) = r_{i_u} - W_u(t) + \sum_l D_l(t_l) \\ + \sum_j p_{ij}\left[w_q(j,t^u) - w_q(j,t+e)\right], \tag{4.14}$$

over all $u \in \mathcal{A}$ since for a given (i,t), the additional term $\sum_l D_l(t_l) - \sum_j p_{ij} w_q(j, t+e)$ does not depend on u. The improved policy then serves the user with the highest index $I_u(i,t)$. To compute $I_u(i,t)$, we need an expression for

$$w_q(j,t^u) - w_q(j,t+e), \tag{4.15}$$

which we derive in the next theorem.

Theorem 4.3.1. *Given the initial policy q and the reward structure in section 2.3, $I_u(i,t)$ is given by*

$$I_u(i,t) = r_{i_u} + D_u(t_u) + \sum_{k=1}^{\infty}(1-q_u)^{k-1}[D_u(t_u+k+1) - D_u(k)]. \tag{4.16}$$

Proof. We only need an expression for the difference in biases given in (4.15). Suppose we follow the randomized policy and consider two sample paths of the $\{(X^n, Y^n), n \geq 0\}$ process under this policy denoted by $\{(X^{n,m}, Y^{n,m}), n \geq 0\}$ for $m = 1, 2$. We assume that $Y^{0,1} = t^u$, $Y^{0,2} = t + e$ and

$$X^{0,1} = X^{0,2} = j; \quad X^{n,1} = X^{n,2} \text{ for } n \geq 0. \tag{4.17}$$

Let $v^m(n)$ be the user served in slot n sample path m. We assume that

$$v^1(n) = v^2(n) \text{ for } n \geq 0. \tag{4.18}$$

Equations (4.17) and (4.18) state the manner in which the two sample paths are coupled. Now fix a $u \in \mathcal{A}$ and assume the user u has been served at time 0. Let S denote the time when we next serve the user u, i.e., $S = \min\{n > 0 : v^1(n) = u\}$. Then recall that $Y_v^{n,m}$ denotes the age of user v in time slot n along the sample path $m \in \{1,2\}$ and we have for $n \leq S$

$$\begin{aligned} Y_v^{n,m} &= t_v + 1 + n \text{ for } v \neq u, m = 1,2, \\ Y_u^{n,1} &= n, \\ Y_u^{n,2} &= t_u + 1 + n, \end{aligned} \tag{4.19}$$

and for $n > S$

$$Y^{n,1} = Y^{n,2}. \tag{4.20}$$

From (4.17), there is no difference in the rewards accrued by the two sample paths. Let C_n^m, $m = 1,2$ be the cost incurred by the path m in time slot n, and $C_n = C_n^1 - C_n^2$ be their difference. Then (4.19), (4.20) and

$$C_n = \sum_{v=1}^{N} \left[D_v(Y_v^{n,1}) - D_v(Y_v^{n,2}) \right], \tag{4.21}$$

imply that

$$C_n = \begin{cases} D_u(t_u + n + 1) - D_u(n) & \text{for } n \leq S \\ \\ 0 & \text{for } n > S. \end{cases} \tag{4.22}$$

Hence, given $S = k$,

$$w_q(j, t^u) - w_q(j, t + e) = \sum_{n=1}^{k} C_n$$

Then, using equation 4.9 we have

$$\begin{aligned} w_q(j, t^u) - w_q(j, t + e) = \sum_{k=1}^{\infty} q_u (1 - q_u)^{k-1} [\{D_u(t_u + 2) + \ldots \\ \ldots + D_u(t_u + k + 1)\} - \{D_u(1) + \ldots + D_u(k)\}]. \end{aligned}$$

Rearranging and simplifying, we get

$$w_q(j, t^u) - w_q(j, t + e) = \sum_{k=1}^{\infty} (1 - q_u)^{k-1} [D_u(t_u + k + 1) - D_u(k)]. \tag{4.23}$$

Further, using $W_u(t) = \sum_{l \neq u} D_l(t_l) = \sum_l D_l(t_l) - D_u(t_u)$, and (4.23), $I_u(i,t)$ is given by

$$r_{i_u} + D_u(t_u) + \sum_{k=1}^{\infty}(1-q_u)^{k-1}[D_u(t_u+k+1) - D_u(k)],$$

as required. □

Starting with the randomized policy the policy improvement step thus yields the policy that chooses action u in state (i,t) that maximizes $\{I_u(i,t), u = 1, 2, \ldots, N\}$. Thus the improved policy is an index policy. Specifically, we shall study the index policy with linear incremental cost, i.e., $D_u(t_u) = K_u t_u$. The index for this policy is particularly simple and is given by the next corollary.

Corollary 4.3.2. *Suppose $D_u(n) = K_u n$ for $n \geq 0$. Then the index is given by*

$$I_u(i,t) = r_{i_u} + K_u t_u (1 + \frac{1}{q_u}) + \frac{K_u}{q_u}. \tag{4.24}$$

Proof. Putting $D_u(n) = K_u n$ in (4.16), we get

$$I_u(i,t) = r_{i_u} + K_u t_u + K_u(t_u + 1)\sum_{k=1}^{\infty}(1-q_u)^{k-1},$$

which simplifies to (4.24), as required. □

An index policy based on the index (4.24) is called a Linear Index Policy (LIP). Clearly, the initial policy q is arbitrary. This immediately begs the question: what q should one use? There are two possible answers:

1. Choose $q_u = \frac{1}{N}$ for all u. In this case,

   $$I_u(i,t) = r_{i_u} + K_u t_u (N+1) + K_u N. \tag{4.25}$$

   We call this policy Uniform Linear Index Policy (ULIP).

2. Choose the q that maximizes the long run average reward g_q of the policy. We discuss the methods of doing that in the next section and call the resulting policy Optimal Linear Index Policy (OLIP).

It is immediately clear from equation 4.24 that any LIP satisfies theorem 3.1.2 and conjecture 3.2.2 and is therefore monotone in age and available data rate in the way described in chapter 3. Thus we can conclude, at least on the basis of monotonicity, that these index policies are reasonable surrogates for the optimal policy.

4.4 Optimizing the Average Reward

In this section we see how an optimal value of q that maximizes g_q can be obtained. We describe a Lagrangian based algorithm to solve the following problem:

$$\text{maximize } g_q$$

$$\begin{aligned} \text{subject to } \sum_{u=1}^{N} q_u &= 1, \\ q_u &\geq 0, \;\; u = 1, 2, \ldots, N. \end{aligned}$$

We solve the problem without the non-negativity constraints first and find that they are automatically satisfied. Define

$$L_P = \sum_{u=1}^{N} -\phi_u(q_u) + A_u q_u + \theta(1 - \sum_{u=1}^{N} q_u),$$

where θ is the Lagrangian multiplier. Differentiate w.r.t. q_u $(u = 1, 2, \ldots, N)$ to get

$$-\phi_u^{'}(q_u) + A_u = \theta. \tag{4.26}$$

It is easy to see that $\phi_u(\cdot)$ is decreasing and $\phi_u^{'}(\cdot)$ is an increasing function of q_u and that

$$\phi_u^{'}(q_u) = -\sum_{k=0}^{\infty}(k+1)(1-q_u)^k\{D_u(k+1) - D_u(k)\}.$$

In our specific case of linear incremental penalty, i.e., $D_u(n) = K_u n$, $\phi_u(q_u) = K_u(1 - q_u)/q_u$ from (4.13), we get

$$\phi_u^{'}(q_u) = \frac{-K_u}{q_u^2}.$$

Using this, (4.26) yields

$$q_u(\theta) = \sqrt{\frac{K_u}{\theta - A_u}} \tag{4.27}$$

where q_u is written as $q_u(\theta)$ to emphasize its dependence on θ. But $\sum_{u=1}^{N} q_u(\theta) = 1$. So the optimal $q_u(\theta)$ subject to the given constraints can be obtained by solving for the optimal θ (call it θ^*) from

$$F(\theta) = \sum_{u=1}^{N} \sqrt{\frac{K_u}{\theta - A_u}} = 1 \tag{4.28}$$

and putting $\theta = \theta^*$ in (4.27). It is clear that $F(\theta)$ is a decreasing function of θ and we can solve (4.28) by bisection methods [47] if we can identify a θ_l and θ_h such that $F(\theta_l) > 1$ and $F(\theta_h) < 1$. The following theorem gives such θ_l and θ_h.

Theorem 4.4.1. *Let*

$$\theta_l = max_u(K_u + A_u) \tag{4.29}$$

and

$$\theta_h = max_u(K_u N^2 + A_u). \tag{4.30}$$

Then, assuming there are at least two users u_1, u_2 such that $K_{u_1} > 0, K_{u_2} > 0$, $F(\theta_l) > 1$ and $F(\theta_h) < 1$.

Proof. Let

$$v = \text{argmax}(K_u + A_u). \tag{4.31}$$

Then $q_v(\theta_l) = 1$, $q_u(\theta) > 0$ for u satisfying $K_u > 0$ and therefore $F(\theta_l) > 1$ since $K_u > 0$ for at least one $u \neq v$. Now from (4.30), for each u we have

$$\sqrt{\frac{K_u}{\theta_h - A_u}} < \frac{1}{N} \tag{4.32}$$

yielding $F(\theta_h) < 1$, as required. □

In the next chapter, we describe the performance analysis methodology used to compare our suggested index policy with the PFA.

Chapter 5

Performance Analysis of LIP and PFA

5.1 Introduction

As mentioned in the introduction the allocation algorithm strives to find the balance between high throughput and low starvation age. In this section we compare the LIP and PFA algorithm. First we show that the space and time complexity of the PFA and ULIP are the same. The state vector of a user in the PFA algorithm is given by the $[Q_u(n), R_u^n]$ and it takes a constant time to update it as seen from equation 2.1. Further, the actual scheduling step involves a maxima over N entities, which is a log$N$ operation. Thus the time complexity of the PFA algorithm is constant + log$N$ while space complexity is $2N$. For the ULIP algorithm (the one we recommend and test), the state vector of a user is given by (Y_u^n, R_u^n), which can again be updated in a constant time, see equation 2.4. The computation of the index of equation 4.25 is a constant time operation. The actual scheduling step involves a maxima over N indices, which is a log$N$ operation. Thus, as with PFA, the time complexity of the ULIP is constant + log$N$ while space complexity is $2N$.

Next we evaluate the performance of an allocation algorithm by studying the following performance measures:

$$\begin{aligned} B &= \text{long run expected throughput per time slot,} \\ \zeta &= \text{long run expected starvation age of a user,} \\ \rho_d &= \text{long run probability that a user is starved for longer} \\ &\quad\ \text{than } d \text{ time slots.} \end{aligned}$$

Having high throughput is important for the service provider to ensure full utilization of the existing infrastructure and maximize profits. It is easy to see that both the long run expected starvation age ζ and the long run probability of a user starving more than a predetermined number of slots ρ_d are measures of the consistency of service, or fairness to users, and are directly related to customer satisfaction. Therefore both ζ and ρ_d are measures of Quality of Service (QoS) to individual users. In this and the next section "QoS level" refers to the (actual or estimated) value of either ζ or ρ_d. We compare the LIP and PFA by plotting throughput against these two QoS levels for each of the two algorithms. Using the probability of delay of packets, that we are estimating by ρ_d, as a QoS has been widely prevalent in the literature [48].

5.2 Characteristics of the LIP

In this section we study these three performance measures for the LIP and PFA algorithms. The mean throughput B is clearly controlled using the parameter $\tau \in [0, 1]$ (introduced in section 2.2) in the case of PFA and $K_u \in [0, \infty)$ in the case of LIP. As an illustration of the effect of τ and K_u on the throughput, we present three theorems corresponding to the extreme values of these parameters. For this section and the next, we assume that all the users have the same PTM P with limiting distribution $\pi = [\pi_1, \dots \pi_M]$, and that $\{X^n : n \geq 0\}$ is aperiodic. Furthermore, we assume that $K_u = K$ for all $u \in 1, 2, \dots, N$ yielding $q_u = \frac{1}{N}$ for OLIP, since in this setting, using (4.27), $q_u = \frac{1}{N}$ optimizes the average reward of the randomized policy. Thus under our assumptions of the user processes and user parameters LIP is the same as ULIP and OLIP. The characteristics and hence performance of the LIP are known for special cases of the constants $K_u, u = 1, 2, \dots, N$ as described below. For general values of K_u, we use simulation in section 5.3 to study the performance of the LIP.

Theorem 5.2.1. *Let $\tau \to 1$ in PFA and $K \to \infty$ in LIP. Then, in the limit, both PFA and LIP converge to the round-robin service rule, i.e., every user is served once every N slots and the limiting throughput is*

$$B = \sum_{k=1}^{M} \pi_k r_k. \tag{5.1}$$

Furthermore, in this limiting regime,

$$\zeta = \frac{N-1}{2},$$

$$\rho_d = \begin{cases} 0 & \textit{if} \quad d > N-1 \\ \frac{N-1-d}{N} & \textit{if} \quad 0 \leq d \leq N-1. \end{cases} \tag{5.2}$$

Proof. First, consider LIP with $K \to \infty$. Then, from (4.25), $I_u(i,t) = r_{i_u} + Kt_u(1+N)$. As $K \to \infty$, the r_{i_u} term becomes insignificant. For sufficiently large K, an equivalent index is $K(1+N)t_u$, i.e., the LIP serves the user with the lowest age. This leads to the round-robin service rule.

Next, setting $\tau = 1-h$ and letting $h \to 0$ in (2.1) and (2.2) yields

$$\frac{R_u^n}{Q_u(n)} = \begin{cases} \frac{R_u^n}{h^{(n-Y_u^n)} R_u^{(n-Y_u^n)}} & \text{if } u \text{ not served at } n-1 \\ \frac{R_u^n}{R_u^{n-1}} & \text{if } u \text{ served at } n-1. \end{cases} \tag{5.3}$$

Clearly the quantity $R_u^n/Q_u(n)$ for the user served in slot $n-1$ remains finite in slot n, but that for other users (not served in slot $n-1$) can become arbitrarily large (as $h \to 0$) in slot n. Thus in the limit as $h \to 0$, the PFA doesn't serve the same user in two consecutive slots. Furthermore, for sufficiently small h, the quantities $\frac{R_u^n}{Q_u(n)}$ in (5.3) are in the same order as the ages Y_u^n for $u \in \mathcal{A}$. Thus, PFA too serves the user with the lowest age yielding the round-robin service rule.

Now, since all users have the same PTM P, $R_{v(n)}^n = r_k$ with probability π_k for $k \in \{1, 2, \ldots N\}$ as $n \to \infty$. Further, since the state space of X^n is finite, $\{R_{v(n)}^n, n \geq 0\}$ is ergodic, we get equation 5.1 [49], as desired.

Finally, as a consequence of the round-robin rule, in any given slot n, the age vector for all the N users is either $(0, 1, \ldots, N-1)$ or a permutation of this sequence. Therefore,

$$\zeta = \frac{1}{N}[1 + 2 + \ldots + N - 1] = \frac{N-1}{2}.$$

Similarly in any time slot, there is no user with age greater than d for $d > N-1$ and the number of users with age greater than d for $0 \leq d \leq N-1$ is equal to $N-1-d$, yielding (5.2), as required. $\square$

The maximum possible throughput is achieved by using what we call the "myopic policy" that serves user $v(n) = \text{argmax}_u R_u^n$ in time slot n. The next theorem gives an expression for the throughput achieved by the myopic policy. For the sake of notational convenience, let

$$\begin{aligned} \alpha_k &= \pi_1 + \pi_2 + \ldots + \pi_k, \text{ and} \\ \gamma_k &= (\alpha_k)^N - (\alpha_{k-1})^N \end{aligned}$$

Theorem 5.2.2. *Let $\tau \to 0$ in PFA and $K \to 0$ in LIP. Then both PFA and LIP converge to the myopic policy and the throughput of the myopic policy is given by*

$$B = \sum_{k=1}^{M} \gamma_k r_k. \tag{5.4}$$

Proof. $K \to 0$ in (4.25) immediately shows that the LIP reduces to the myopic policy. Similarly, letting $\tau \to 0$ in (2.2) shows that the PFA approaches the myopic policy.

We know that for $n \to \infty$, $Pr[X_u^n = k] = \pi_k$, $1 \leq k \leq M$, for $u \in \{1, \ldots, N\}$. Let

$$U^n = \max\{X_1^n, X_2^n, \ldots, X_N^n\}.$$

Then,

$$Pr[U^n \leq k] = Pr[X_1^n \leq k] Pr[X_2^n \leq k] \ldots Pr[X_N^n \leq k] = (\alpha_k)^N.$$

Therefore,

$$\gamma_k = Pr[U^n = k] = Pr[U^n \leq k] - Pr[U^n \leq k-1] = (\alpha_k)^N - (\alpha_{k-1})^N.$$

Clearly,

$$B = \sum_{k=1}^{M} \lim_{n \to \infty} Pr[U^n = k] r_k,$$

yielding

$$B = \sum_{k=1}^{M} \gamma_k r_k, \tag{5.5}$$

as required. □

In section 5.3, we also consider the *dynamic population* case where the users arrive in the cell according to a Poisson process with rate λ and remain in the cell for a generally distributed time with mean a. Let $N(t)$ be the number of users in a dynamic cell at time t. Then $\{N(t), t \geq 0\}$ is the number of customers in an $M/G/\infty$ queue and the LIP index of (4.25) is modified for this case as follows:

$$I(u,t) = r_{i_u} + Kt_u(N(t)+1). \tag{5.6}$$

Using the results for the $M/G/\infty$ queue [46] we see that the number of users in steady state is a Poisson random variable with parameter λa. The following theorem gives an expression for the myopic policy throughput in this setting. We need the following notation:

$$s_k = e^{-\lambda a(1-\alpha_k)} - e^{-\lambda a(1-\alpha_{k-1})}.$$

Theorem 5.2.3. *The throughput of the myopic policy in the dynamic population case is given by*

$$B = \sum_{k=1}^{M} s_k r_k. \tag{5.7}$$

Proof. Let N represent the number of users in the cell in steady state. It is known that N is a Poisson random variable with parameter λa. The myopic policy chooses the user to be served in a given time slot independent of the starvation age. Therefore, from (5.5),

$$\text{E}[\text{Throughput}|N=n] = \sum_{k=1}^{M} \left[(\alpha_k)^n - (\alpha_{k-1})^n\right] r_k.$$

Since $B = \text{E}[\text{E}[\text{Throughput}|N]]$, we have

$$B = \sum_{k=1}^{M} \left(\text{E}\left[(\alpha_k)^N\right] - \text{E}\left[(\alpha_{k-1})^N\right]\right) r_k,$$

which yields (5.7) using $\text{E}\left[z^N\right] = e^{-\lambda a(1-z)}$ from standard probability theory [50]. □

The performance of LIP can be compared with that of PFA using plots of B vs ζ and B vs ρ_d for both the policies as τ and K vary. The policies implied by values of the LIP parameters $(K_u, u = 1, 2, \ldots, N)$ and PFA parameters (τ) other than at extreme values are too complex to yield to analytical expressions for B, ζ or ρ_d. Since analytical results are not available except for the extreme values of these parameters we resort to simulation to estimate these quantities. This is done in the next section.

5.3 Simulation Results

We now use simulation to estimate B, ζ and ρ_d for the LIP and PFA for a given K and τ. The code for the simulation has been written using the C programming language. We begin by formulating the estimators for these parameters below.

5.3.1 The Estimators

In this and the following sections we use $\hat{B}$ to denote the estimator of B, etc. We consider the static case first; i.e., the number of users in the cell is a constant N. Let L be the number of independent sample paths simulated and T be the number of slots in each path. Let $R_{v(n)}^{n,l}$ be the throughput in the n^{th} slot of the l^{th} sample path, $l = 1, 2, \ldots, L$ and $n = 1, 2, \ldots, T$. Then the estimator $\hat{B}$ of B is given by

$$\hat{B} = \frac{1}{L} \sum_{l=1}^{L} \left(\frac{1}{T} \sum_{n=1}^{T} R_{v(n)}^{n,l} \right). \tag{5.8}$$

Let $Y_u^{n,l}$ be the age of the user u in the n^{th} slot in sample path l, $l = 1, 2, \ldots, L; n = 1, 2, \ldots, T$. Then we define the estimator $\hat{\zeta}$ of ζ as

$$\hat{\zeta} = \frac{1}{L} \sum_{l=1}^{L} \left[\frac{1}{T} \sum_{n=1}^{T} \left(\frac{1}{N} \sum_{u=1}^{N} Y_u^{n,l} \right) \right]. \tag{5.9}$$

Similarly the estimator $\hat{\rho_d}$ of ρ_d is given by

$$\hat{\rho_d} = \frac{1}{L} \sum_{l=1}^{L} \left[\frac{1}{T} \sum_{n=1}^{T} \left(\frac{1}{N} \sum_{u=1}^{N} \mathbf{1}\{Y_u^{n,l} > d\} \right) \right], \tag{5.10}$$

where

$$\mathbf{1}\{Y_u^{n,l} > d\} = \begin{cases} 1 & \text{if } Y_u^{n,l} > d \\ \\ 0 & \text{otherwise.} \end{cases} \tag{5.11}$$

In the *dynamic population* case the estimators $\hat{B}$, $\hat{\zeta}$ and $\hat{\rho_d}$ are obtained by replacing N with $N(t)$ $(t = 1, 2, \ldots, T)$ and at the same time excluding contribution from the time slots for which $N(t) = 0$. Thus the estimators in the *dynamic population* case are given by

$$\begin{aligned} \hat{B} &= \frac{1}{L}\sum_{l=1}^{L}\left(\frac{1}{T}\sum_{n=1}^{T} R_{v(n)}^{n,l}\right), \\ \hat{\zeta} &= \frac{1}{L}\sum_{l=1}^{L}\left[\frac{1}{T}\sum_{t=1,N(t)>0}^{T}\left(\frac{1}{N(t)}\sum_{u=1}^{N(t)} Y_u^{t,l}\right)\right], \\ \hat{\rho_d} &= \frac{1}{L}\sum_{l=1}^{L}\left[\frac{1}{T}\sum_{t=1,N(t)>0}^{T}\left(\frac{1}{N(t)}\sum_{u=1}^{N(t)} \mathbf{1}\{Y_u^{t,l} > d\}\right)\right]. \end{aligned}$$

It is worth noting that all these estimators are for long-run performance measures, and so we need to collect samples only from the stationary region of the Markov chain $\{X^n, n \geq 0\}$ (and of $\{N(t), t \geq 0\}$ in the dynamic population case). To ensure this, both in the constant and dynamic population case, we start the simulation in the stationary distribution of the $\{X^n, n \geq 0\}$ and the $\{N(t), t \geq 0\}$ process. We plot estimate of B vs estimates of ζ and ρ_d for both the policies, and see that for any given QoS level, the LIP produces higher throughput.

5.3.2 Simulation Parameters

We use the following set of available data rates (kbps) [33]: r ={38.4, 76.8, 102.6, 153.6, 204.8, 307.2, 614.4, 921.6, 1228.8, 1843.2, 2457.6}. Thus, each Markov chain $\{X_u^n, n \geq 0\}$ has $M = 11$ states. We use $d = 100$, $T = 10^5$ and

$$P = \gamma I + \frac{1-\gamma}{M-1}(D - I),$$

where I is an $M \times M$ identity matrix and D is an $M \times M$ matrix with all entries equal to 1. This implies that the Markov chain stays in a given state for *Geometric*$(1-\gamma)$ number of time slots and then moves to one of the remaining $M-1$ states with equal probability. Note that P is doubly stochastic, and hence $\pi_k = 1/M = 1/11$. Further, the length of a time slot is $1.67 * 10^{-3}$ seconds [10]. We choose $\gamma = 0.9999$ implying that on the average the Markov chain stays in one state for 10^4 time slots, i.e., 16.7 seconds, before changing states. To get the required plots we vary K for LIP and τ for the PFA. The actual ranges are mentioned in the plots.

5.3.3 Constant Number of Users

We take samples from $L = 100$ sample paths of the Markov chain. We conduct the simulation for various values of N and display the plots of $\hat{B}$ vs $\hat{\zeta}$ in figures 5.1(a) and 5.1(b). We obtain the plots by running the LIP and PFA algorithms for a range of values of K (between 0 and 500) and τ (between 0 and 0.5) respectively. Each point on the LIP and PFA plot corresponds to a unique value of K and τ respectively. Both K and τ decrease as we move from left to right on the curves. As expected, $\hat{\zeta}$ and $\hat{B}$ decrease with τ and K. The points with the minimum $\hat{\zeta}$ correspond to $K = 500$ and $\tau = 0.5$, with both the curves approaching the throughput for the round-robin policy (calculated to be 722.62 using (5.1)) as proved in theorem 5.2.1. Similarly, the points with the maximum $\hat{\zeta}$ correspond to $K = 0$ and $\tau = 1.25 * 10^{-10}$. The LIP throughput and the PFA throughput both converge to the throughput for the myopic policy (using (5.4), calculated to be 2121.31 for $N = 10$ and 2452.34 for $N = 50$) as proved in theorem 5.2.2. The LIP plot reaches close to its maximum value at a much lower mean starvation age (than the PFA) corresponding to low values of K (0.05 for $N = 10$ and 0.003 for $N = 50$) and does not increase significantly by reducing K further. As is clear from figure 5.1, LIP outperforms PFA significantly over the entire range of $\hat{\zeta}$. Thus no matter what QoS level we choose for the users, using LIP always results in better system throughput. Also we see that the PFA throughput converges to the myopic policy throughput (this is the maximum throughput achievable) at a much slower rate than the LIP throughput. This enhanced performance of the LIP is expected because the LIP is the result of a method that starts with the objective of maximizing throughput while penalizing user starvation at the same time. The PFA on the other hand has not been derived with such an objective in mind. Further, we also carried out simulations with transition probability matrices having lower values of γ (implying transitions to other states occur at a faster rate on an average). The improvement of LIP over PFA does go down as γ decreases, i.e., as the average time spent in a state decreases. However its worth noting that in practice γ would only be higher than we have used for simulations (users will not change state every 16.7 seconds or lesser) and so the LIP will outperform the PFA.

Similarly figure 5.2 shows the plots of $\hat{B}$ vs $\hat{\rho_d}$ for different values of N. As in the plots of figure 5.1, each point on the LIP and PFA curves of the plots of figure 5.2 represents a unique value of K and τ respectively. Both K and τ increase as we move from left to right on the curves. We see in both figures 5.2(a) and 5.2(b) that for any given value of $\hat{\rho_d}$, the LIP throughput is significantly higher than that of PFA. Thus LIP demonstrates better performance at all QoS levels. Further, even at very high levels of $\hat{\rho_d}$, the PFA throughput does not always approach the myopic policy throughput (which, as noted above, is the maximal throughput). On the other hand, using LIP gives us an option of achieving a throughput close to the myopic policy throughput for a high QoS level. The substantial domination of the LIP plot for both the QoS measures clearly demonstrates the superiority of LIP over PFA. Next, we consider the case when users can enter or leave the cell.

5.3.4 Poisson Arrival of Users

In this section, we assume users arrive according to a Poisson process with rate λ. Once in the cell, sojourn time of a user is exponentially distributed with mean a. Thus, in steady state the number of users is a Poisson random variable with mean $N_{\text{avg}} = \lambda a$. The LIP index is given by (5.6). We then serve the user with the maximum value of index $I(u,t)$. For PFA, we initialize Q_u to 1 for a newly arriving user and $R_u^n/Q_u(t)$ is maximized among all the $N(t)$ users present in the cell at the beginning of the time slot t. Again, we consider $L = 100$ sample paths of the process $\{(X^t, Y^t), t \geq 0\}$ with T time slots in each sample path. The graphs in figures 5.3(a) to 5.3(c) and 5.4(a) to 5.4(c) show the same qualitative behavior as in the case of constant number of users. We explain them in more detail below.

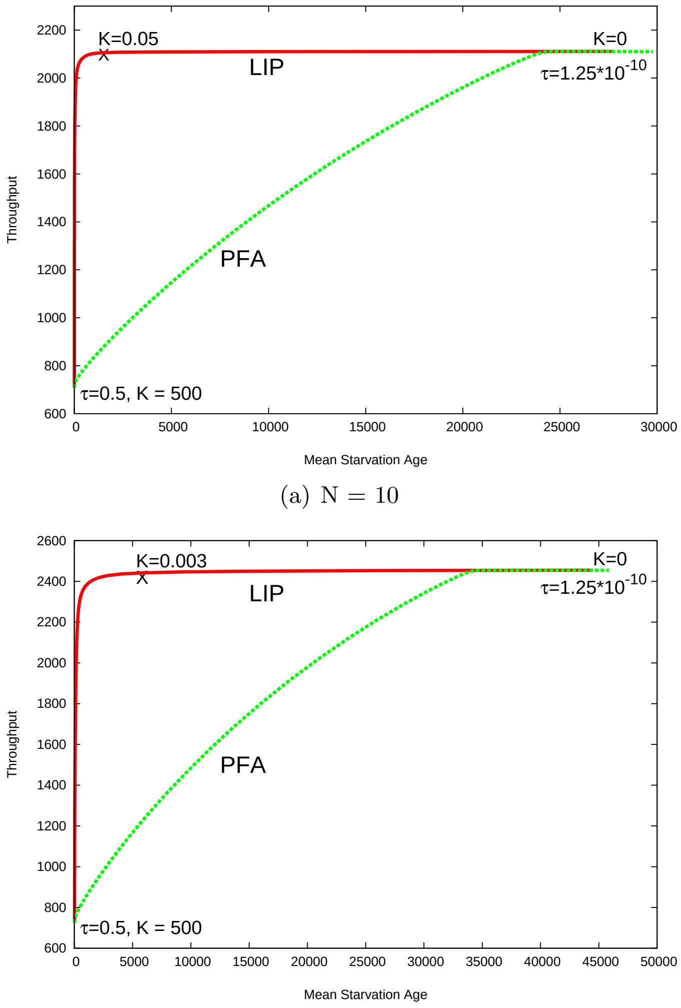
Considering the mean age measure of QoS first, we show in figure 5.3 plots of $\hat{B}$ vs $\hat{\zeta}$. As in subsection 5.3.3, the bottom left point corresponds to $\tau = 0.5$ and $K = 500$ approaching the round-robin policy (theorem 5.2.1), and the top right point corresponds to $\tau = 2 * 10^{-11}$ and $K = 0$ corresponding to the myopic policy (theorem 5.2.2). The throughput corresponding to $K = 0.06$ for $N_{\text{avg}} = 10$ and to $K = 0.01$ for $N_{\text{avg}} = 50$ is very close to the myopic policy throughput as indicated in figure 5.3. The theoretical value computed for the round-robin policy remains the same at 722.62. The values of the myopic policy throughput, however, are now given by theorem 5.2.3, and we compute them to be 2078.31 for $N_{\text{avg}} = 10$ and 2451.01 for $N_{\text{avg}} = 50$. For any given level of the QoS (any value of $\hat{\zeta}$) of practical interest, the LIP throughput is considerably more than the PFA throughput. At very high values of $\hat{\zeta}$ the LIP throughput and the PFA throughput get close to each other, but this region is not of any practical interest since we get almost the same throughput as the myopic policy throughput from the LIP at much lesser $\hat{\zeta}$ values.

Next we consider ρ_d as the QoS measure and plot $\hat{B}$ vs $\hat{\rho_d}$ for various combinations of N_{avg} in figure 5.4. On both the LIP and PFA curves, K and τ vary as in the plots of figure 5.3. The bottom left point now corresponds to $\tau = 0.5$ and $K = 500$ approaching the round-robin policy (theorem 5.2.1), and the top right point corresponds to $\tau = 2 * 10^{-9}$ and $K = 0.01$ approaching the myopic policy (theorem 5.2.2). Across the entire achievable range of QoS, LIP throughput is much better than the PFA throughput.

5.4 Summary

The graphs corresponding to both the QoS measures demonstrate the significantly better performance of the LIP algorithm. Another algorithm for resource scheduling in the setting of this chapter is suggested in [35]. However for the identical users case considered in this section their algorithm reduces to the myopic policy. Hence we do not need to study it separately.

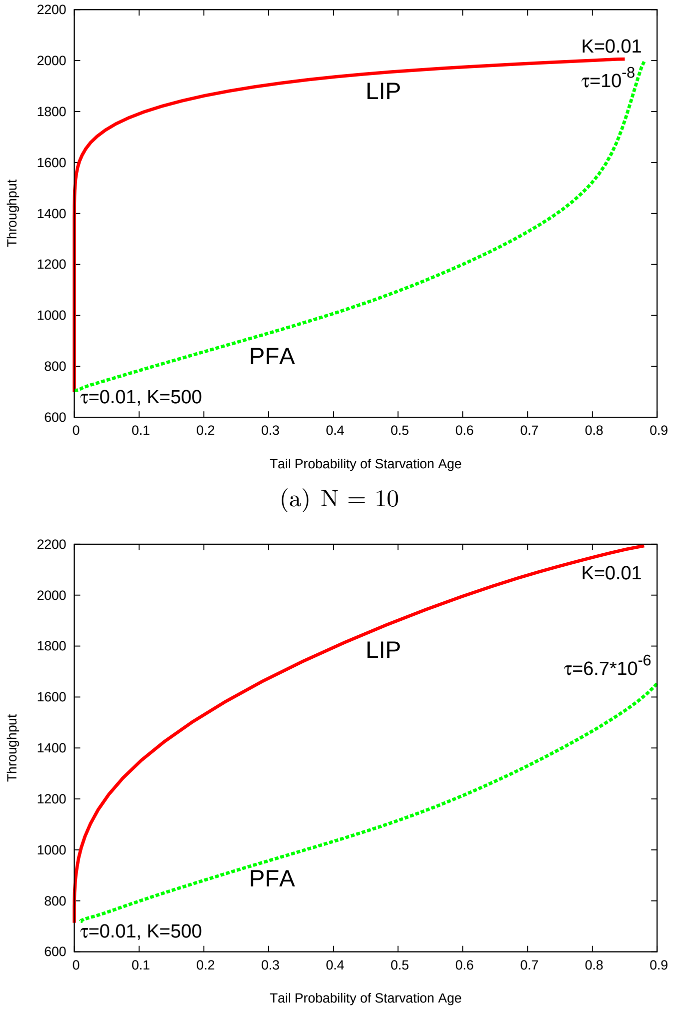
Thus we have used the MDP formulation and the policy improvement approach in this part of the dissertation to develop an index policy for the data transfer problem in a wireless telecommunication cell with infinitely backlogged queues. The index is a simple, intuitive, closed form expression and we have demonstrated its superior performance over the existing PFA over a wide parameter space. In part II we consider the scheduling problem when data arrives for each user in each time slot according to some distribution.



(a) N = 10

(b) N = 50

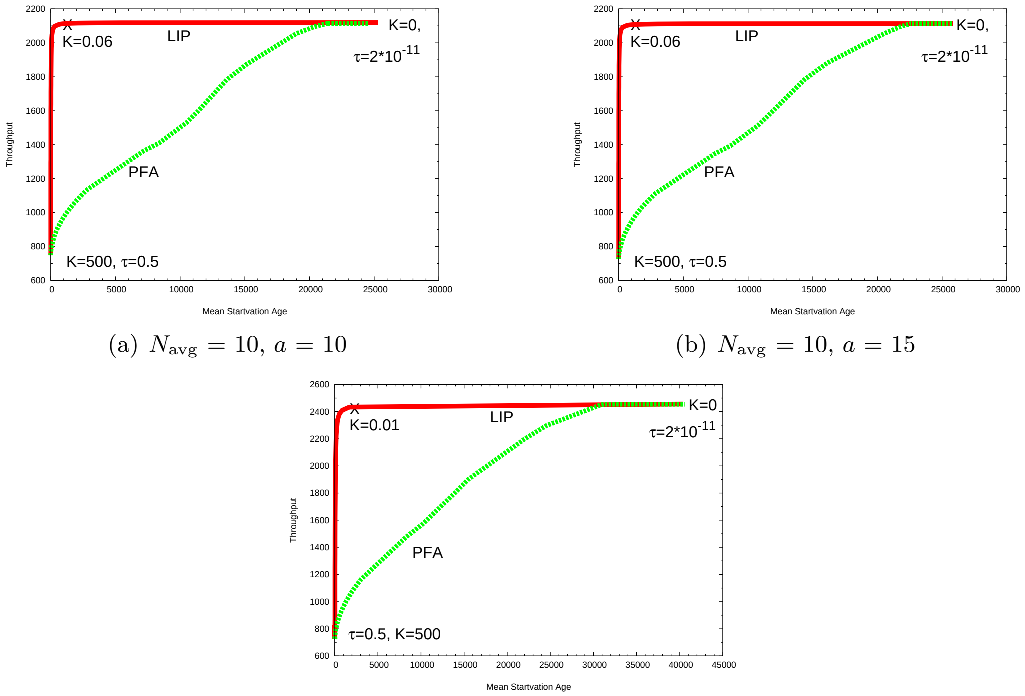
Figure 5.1: Throughput Vs Mean Starvation Age: Constant N. Each point on the LIP plot corresponds to a value of K and on the PFA plot to a value of τ. K varies from 0 to 500, with the throughput corresponding to $K = 0.05$ for $N = 10$ and $K = 0.003$ for $N = 50$ matching the myopic policy throughput closely. τ varies from $1.25 * 10^{-10}$ to 0.5.



(a) N = 10

(b) N = 50

Figure 5.2: Throughput Vs Probability of Starvation for greater than d slots for LIP and PFA: Constant N. Each point on the LIP plot corresponds to a value of K and on the PFA plot to a value of τ. K varies from 0.01 to 500. The throughput for $K = 0.01$ matches the myopic policy throughput closely in both cases. τ varies from 10^{-8} to 0.01 for $N = 10$, and from $6.7{*}10^{-6}$ to 0.01 for $N = 50$.

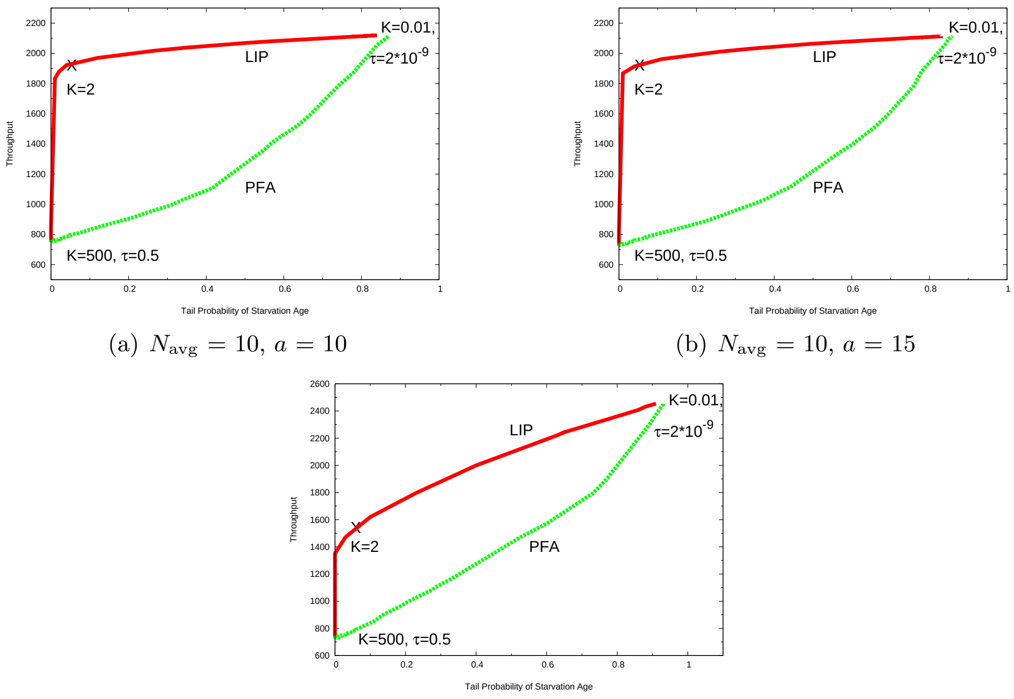


(a) $N_{\text{avg}} = 10$, $a = 10$

(b) $N_{\text{avg}} = 10$, $a = 15$

(c) $N_{\text{avg}} = 50$, $a = 10$

Figure 5.3: Throughput Vs Mean Starvation Age: Varying N (Poisson arrivals). Each point on the LIP plot corresponds to a value of K and on the PFA plot to a value of τ. K varies from 0 to 500, with the throughput for $K = 0.06$ or lesser matching the myopic policy throughput closely. τ varies from $2 * 10^{-11}$ to 0.5.



(a) $N_{\text{avg}} = 10$, $a = 10$

(b) $N_{\text{avg}} = 10$, $a = 15$

(c) $N_{\text{avg}} = 50$, $a = 10$

Figure 5.4: Throughput Vs Probability of Starvation for greater than d slots for LIP and PFA: Varying N (Poisson arrivals). τ varies from $2 * 10^{-9}$ to 0.5. K varies from 0.01 to 500.

Part II

External Data Arrival

List of Notations for part II

(in the order of appearance)

N - Total number of users

u - Label for the users, $u = 1, 2, \ldots, N$

R_u^n - Channel rate of user u

R^n - The vector $[R_u^n : u = 1, 2, \ldots, N]$

X_u^n - State of user u at time n

M - The number of states in the state space of the DTMC $\{X_u^n,\, n \geq 0\}$ for $u = 1, 2, \ldots, N$

P^u - The Transition probability matrix of the Markov chain $\{X_u^n,\, n \geq 0\}$; has elements $[p_{i_u, j_u}^u]$

$X^n = [X_1^n, \ldots, X_N^n]$ - State vector of all users

$i = [i_1, i_2, \ldots, i_N]$ - A realized value of the state vector X^n

Y_u^n - Queue length of the user u at time n

$Y^n = [Y_1^n, \ldots, Y_N^n]$ - The Queue length vector at time n

A_u^n - Number of packets arrivals for user u at time n

$A^n = [A_1^n, \ldots, A_N^n]$ - The packet arrival vector at time n

$y = [y_1, y_2, \ldots, y_N]$ - A realized value of the queue length vector Y^n

$v(n)$ - User served in the n^{th} time slot

Ω - State space of the DTMC $\{X_u^n,\, n \geq 0\}$; is same for all users $u = 1, 2, \ldots, N$

Z - The set of all non-negative integers $\{0, 1, 2, \ldots\}$ r - A constant vector of data rates $= [r_1, r_2, \ldots, r_M]$; when $X_u^n = k$, $R_u^n = r_k$

$V_D(i, y, a)$ - Optimal cost starting from state $[X^0, Y^0, A^0] = [i, y, a]$ at time 0 over time periods $0, 1, 2, \ldots, D$

g - The long run average throughput

$w(i, y, a)$ - Bias function starting in state (i, y, a)

q - Initial policy vector $= [q_1, q_2, \ldots, q_N]$

g_q - The constant g under the initial q-policy

$w_q(i, y, a)$ - The bias function $w(i, y, a)$ under the initial q-policy

$\pi^u = [\pi_1^u, \ldots, \pi_M^u]$ - Steady state distribution of the Markov chain $\{X_u^n : n \geq 0\}$

K_u - Parameter of the IP for user u so that holding cost in one time slot for y_u packets $= K_u y_u$

$I_u(i_u, y_u, a_u)$ - The index for user u in state (i_u, y_u, a_u)

B - Long run expected throughput per time slot

ξ - Sum of long run expected queue length of all users

$\hat{B}$ - Estimate of B obtained from simulation

$\hat{\xi}$ - Estimate of ξ obtained from simulation

$G(Y)$ - The Lyapunov function defined on $Y \in Z^N$ $N(n)$ - The number of users at time t in the cell in the dynamic case

λ - The arrival rate of users in the dynamic cell

λ^p - The arrival rate for packets of a user (equal for all users) in the static cell

a - Sojourn time of a user is exponentially distributed with mean a in a dynamic cell

Chapter 6

The Model

6.1 Motivation

In this chapter we consider the case where there is external data arrival for each user in each time slot. The base station maintains a separate data queue for every user in the cell. Let Y_u^n be the number of packets in the queue for user u at the beginning of time slot n, A_u^n be the number of packets the user u receives during the time slot n. For the sake of notational convenience, in this chapter and the rest of part II, we use Y_u^n to denote the data queue length of user u in slot n (whereas, recall that in all of part I, Y_u^n meant the starvation age of user u in slot n). Thus $Y_u^n + A_u^n$ is the number of packets available for transfer if the user u is served in the n^{th} slot. We assume that at all times n the base station also keeps track of the queue length vector $Y^n = [Y_1^n, Y_2^n, \ldots, Y_N^n]$ and data arrival vector $A^n = [A_1^n, A_2^n, \ldots, A_N^n]$ in addition to the data rate vector R^n. Thus $\{Y^n, n \geq 0\}$ is the queue length process and $\{A^n, n \geq 0\}$ the packet arrival process both of whose components evolve stochastically. Let S_u^n be the amount of data actually transferred in slot n if user u is chosen to receive data in that time slot. Clearly, $S_u^n = \min(R_u^n, Y_u^n + A_u^n)$.

In this setting the system wide objective function of interest to us will not be the average throughput per time slot, since any scheduling algorithm that restricts the queue lengths from increasing indefinitely without any bounds (i.e., any algorithm that induces a stationary distribution of data queue length for all users) results in the same throughput. To see this, consider the data queue of any user: in steady state, whatever data enters the queue has to leave it, thus implying a fixed throughput that depends on the average rate of data arrival.

In the absence of throughput maximization, a useful measure of overall customer satisfaction is the total length of the data queues of all users. The service provider attempts to empty all data queues as quickly as possible, so that it can serve all the demand of data with the smallest delay possible. Thus a reasonable objective is to minimize the data queue length (across all users) remaining after the chosen user u is served. Therefore, if K_l, $l \in \mathcal{A}$ is the cost of holding one packet of data for one time slot in the queue of user l, we seek to minimize

$$E[K_u \max(Y_u^n + A_u^n - R_u^n, 0) + \sum_{l \neq u} K_l(Y_l^n + A_l^n)] = E[\sum_l K_l(Y_l^n + A_l^n) - K_u S_u^n].$$

Note that this objective prevents any user from being starved indefinitely, since the data queues of users not served keep on increasing, eventually causing every user to be served. This objecive function also provides an incentive to the base station to maximize its data transmission in any given time slot (since doing that reduces the data queue length the most). Thus it addresses the basic scheduling conflict well.

As in the infinitely backlogged case, the key features of this scheduling problem are: stochastic evolution of the data rates available to the users and the length of their data queues and penalties incurred by the system for users that are not served during n. As noted earlier, it is well known that Markov Decision Processes (MDP) can be used to determine optimal decisions (that is, which user to serve in this case) in such settings. Further, using a single step of the policy improvement approach can yield index policies [44, 45] that are nearly optimal and do not suffer from the curse of dimensionality.

6.2 Existing Algorithms

In this section we describe three existing algorithms to solve the scheduling problem described in section 6.1. Recall that $v(n)$ denotes the user chosen for service during time slot n. Each algorithm presented below can be seen as an "index policy", i.e., every user is assigned an "index" based on its current channel conditions and queue length in every time slot. The user that gets served in a given time slot is the user with the largest index value.

1. Max-Weight Algorithm (MWA): This algorithm is used and cited most often and is in some sense the starting point of the others presented below. As mentioned in section 1.4 this algorithm was first introduced by Tassiulas and Ephremides in [14, 15] and Awerbuch and Leighton in [12, 13]. In time slot n, it serves the user u ($u \in \mathcal{A}$) for whom the product of the available data rate and the total data available to transmit is maximized. Mathematically,

$$v(n) = \arg\max_u R_u^n (Y_u^n + A_u^n). \tag{6.1}$$

A stability condition, i.e., condition(s) under which queue lengths are bounded, for this algorithm is derived in [14].

2. Generalized Max-Weight Algorithm: Andrews et al [23] present and prove the stability of another algorithm that is a generalization of the max-weight algorithm using additional parameters as follows:

$$v(n) = \arg\max_u \gamma_u R_u^n (Y_u^n + A_u^n)^{\beta}, \tag{6.2}$$

where γ_u and β are arbitrary positive constants. We call this the Generalized Max-Weight Algorithm (GMW). The authors in [23] also prove the stability of a more generalized version of GMW where they replace the queue length $Y_u^n + A_u^n$ in equation 6.2 with a linear combination of the queue length and the head-of-the-line packet delay.

3. The Exponential Rule: Introduced by Shakkottai and Stolyar [24], this rule is another generalization of the max-weight algorithm. The index for every user has more parameters specific to that user as well as information about queue length of other users. Let γ_u, a_u, $u = 1, 2, \ldots, N$ and β, $\eta \in (0, 1)$ be positive constants. Then the exponential rule serves the user $v(n)$ in time slot n such that

$$v(n) = \arg\max_u \gamma_u R_u^n exp\left(\frac{a_u Y_u^n}{\beta + (\bar{Y}^n)^{\eta}}\right), \tag{6.3}$$

where $\bar{Y}^n = (1/N) \sum_u a_u Y_u^n$. This algorithm has the property that for each time n, it minimizes $\max_u a_u Y_u^n$ [25]. The constants β, η and the arbitrary parameters a_u, γ_u give more flexibility to the scheduling algorithm. However, no method is specified to select these parameters.

The scheduling algorithm based on each of the above indices is proven to be stable. We derive our index based on an MDP formulation (with associated costs) which attempts to minimize the long run average cost described in section 6.3. We evaluate its performance using simulation. Further, we do not actually solve the MDP to optimality to compute the index, thus making the index policy computationally easy to implement. As described in detail in sections to follow, the MDP formulation just gives us a starting point and a sound theoretical framework for the derivation of the ultimate index policy. Furthermore, we prove that the index computed using our approach generates a stable scheduling policy.

6.3 MDP Formulation

We formulate the problem as an MDP in this section. We start with a model for the data rate process $\{R^n,\ n \geq 0\}$, the data arrival process $\{A^n,\ n \geq 0\}$ and the queue length process $\{Y^n, n \geq 0\}$ of the users. We use the same model for the data rate process as in section 2.3, but we again present it here for ready reference. In systems currently used in practice such as CDMA2000 1xEV-DO system [33], the base station serves users at one of the M data rates $r_1, r_2, \ldots, r_M$. Each of these M data rates corresponds to the "environmental state" of a given user. This "environmental state" of the user takes into account factors such as distance from the base station and topography. As in the infinitely backlogged case, let X_u^n be the environmental state of user u during the time slot n. For ready reference we again present the model of X_u^n described in section 2.3: $\{X_u^n,\ n \geq 0\}$ is an irreducible Discrete Time Markov chain (DTMC) on state space $\Omega = \{1, 2, \ldots, M\}$ with Transition Probability Matrix (TPM) $P^u = [p^u_{i_u, j_u}]$. For example, a set of $M = 11$ fixed data rates is available to users in an actual system [33]. During any time slot n, for every user u the underlying DTMC $\{X_u^n,\ n \geq 0\}$ determines the data rate R_u^n as follows: for $k \in \Omega$ $X_u^n = k \Longrightarrow R_u^n = r_k$. Further, let $X^n = [X_1^n, \ldots, X_N^n]$ be the environmental state vector of all the users. We assume that all users are independent of each other and thus each component of $\{X^n, n \geq 0\}$ is an independent DTMC on Ω. Hence it is clear that $\{X^n,\ n \geq 0\}$ itself is a DTMC on Ω^N.

Next we consider the data model for each user. The base station maintains a separate queue for every user u. Let A_u^n be the number of packets that arrive at the base station for user u during time slot n. We assume that the arrival process $\{A_k^n, n \geq 0\}$ is independent of

$\{A_l^n, n \geq 0\}$ for $k \neq l$. Further we assume that for every u, $\{A_u^n,\ n \geq 0\}$ is an irreducible DTMC on state space $Z = \{0, 1, 2, \ldots\}$ with TPM $H^u = [h^u_{i_u, j_u}]$. Since each component of $\{A^n, n \geq 0\}$ is an independent DTMC on Z, $\{A^n,\ n \geq 0\}$ itself is a DTMC on Z^N. An example of a Markovian packet arrival process is when $\{A_u^n, n \geq 0\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with the probability distribution $h^u = [h_1^u, h_2^u, \ldots]$. In this case every row of the TPM H^u is equal to h^u. In this and the following sections we also refer to A_u^n as the "packet arrival state" of user u in time slot n.

We can now describe the evolution of the queue length process $\{Y^n, n \geq 0\}$. Recall that, for every user u, Y_u^n is the number of data packets in the queue for user u at the beginning of time slot n. Therefore $\{Y_u^n, n \geq 0\}$ changes according to

$$Y_u^{n+1} = \begin{cases} Y_u^n + A_u^n & \text{if } u \neq v(n) \\ Y_u^n + A_u^n - \min(R_u^n, Y_u^n + A_u^n) & \text{if } u = v(n). \end{cases} \tag{6.4}$$

The "state" of user $u \in \mathcal{A}$, given by the vector $[X_u^n, Y_u^n, A_u^n] \in \Omega \times Z \times Z$, thus has three components: the "environmental state" X_u^n, the "queue length state" Y_u^n and the "data arrival state" A_u^n. The "state of the system" at time n is then given by $[X^n, Y^n, A^n] \in \Omega^N \times Z^N \times Z^N$. It is thus a vector of $3N$ components and we assume that it is known to the base station in each time slot.

Unless the data queues of all users are empty, the base station serves exactly one user in every time slot after observing the system state in that time slot. We need a cost structure to make this decision optimally and we propose it below. We pay a cost K_u to hold one packet for one time slot for user u. This cost structure is thus equivalent to choosing linear incremental penalty $D_l(y_l) = K_l y_l$ in the infinitely backlogged case. The queue length of every user $l \neq u$ is $Y_l^n + A_l^n$. For user u, however, the queue length (after transmission) equals $(Y_u^n + A_u^n - R_u^n)^+$ where $(x)^+ = \max(x, 0)$ for any real number x. The total holding cost W_u^n in time slot n is then given by

$$\begin{aligned} W_u^n &= \sum_{l \neq u} K_l \cdot (Y_l^n + A_l^n) + K_u \cdot (Y_u^n + A_u^n - R_u^n)^+ \\ &= \sum_{l} K_l \cdot (Y_l^n + A_l^n) - K_u \min\left(Y_u^n + A_u^n, R_u^n\right). \end{aligned} \tag{6.5}$$

Thus the total cost incurred by serving user u in time slot n is W_u^n. As in the infinitely backlogged case, we assume that there is no cost in switching from one user to another from slot to slot. This is not entirely true in practice, but including switching costs in the model makes the analysis intractable.

Having described the system state, its evolution and the cost structure, we can now model the problem of scheduling a user to serve in a given time slot as an MDP. The decision epochs are the time slots $\{1, 2, \ldots\}$. The state at time n is $[X^n, Y^n, A^n]$ with Markovian evolution described above. The action space in every state is $\mathcal{A}$ where action u corresponds to serving user u. The cost in state $[X^n, Y^n, A^n]$ for action u is W_u^n. The transition probability $p((j,z,b)|(i,y,a),u)$ under action u from (i,y,a) to (j,z,b) ($i, j \in \Omega^N$, $y, z \in Z^N$, $a, b \in Z^N$) is given by

$$p((j,z,b)|(i,y,a),u) = \begin{cases} p^1_{i_1,j_1} \ldots p^N_{i_N,j_N} h^1_{a_1,b_1} \ldots h^N_{a_N,b_N} = p_{ij} h_{ab} & \text{if } z_u = (y_u + a_u - r_{i_u})^+ \text{ and} \\ & z_l = y_l + a_l \text{ for } l \neq u \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

Next we describe the value functions that form the basis of the PIA. Let $V_D(i,y,a)$ be the optimal cost starting from state $[X^0, Y^0, A^0] = [i, y, a]$ at time 0 over time periods $0, 1, 2, \ldots, D-1$. If user u is served at time 0 the queue length vector in the next time slot is

$$(y+a,i)^u = (y_1 + a_1, \ldots, y_{u-1} + a_{u-1}, (y_u + a_u - r_{i_u})^+, y_{u+1} + a_{u+1}, \ldots, y_N + a_N). \tag{6.7}$$

For the sake of notational convenience, we define

$$W_u(i, y+a) = \sum_l K_l \cdot (y_l + a_l) - K_u \min\left(y_u + a_u, r_{i_u}\right). \tag{6.8}$$

A standard Dynamic Programming (DP) argument then yields the following Bellman equation

$$V_D(i, y, a) = \min_{u=1,2,\ldots,N} \left[W_u(i, y+a) + \sum_{(j,b)} p_{ij} h_{ab} V_{D-1}(j, (y+a, i)^u, b) \right]. \tag{6.9}$$

The goal of the scheduling policy is to determine the action $u = u(i, y, a)$ that minimizes the long run average cost $\lim_{D\to\infty} V_D(i, y, a)/D$ given the state $(i, y, a) \in \Omega^N \times Z^N \times Z^N$ of the system. It is well known from standard MDP theory [38] that such a policy $\{u(i, y, a)\}$ exists if there is a constant g and a bias function $w(i, y, a)$ satisfying

$$g + w(i, y, a) = \min_u \{W_u(i, y+a) + \sum_{(j,b)} p_{ij} h_{ab} w(j, (y+a, i)^u, b)\}. \tag{6.10}$$

The physical significance of g and the bias function is similar to that in part I and easy to see. Furthermore, any u that minimizes $W_u(i, y+a) + \sum_{(j,b)} p_{ij} h_{ab} w(j, (y+a, i)^u)$ over all $u \in \{1, \ldots, N\}$ is an optimal action $u(i, y, a)$ in state (i, y, a).

Chapter 7

Monotonicity of the Optimal Policy

We discussed in chapter 6 that solving the MDP to optimality is infeasible. However, as in part I, we expect the optimal policy to possess several monotonicity properties. In this chapter we define monotonicity in data queue length, rate and the number of packet arrivals using a framework similar to that of chapter 3. We prove monotonicity in data queue length of the optimal policy for the expected total discounted cost. We also prove partial results about monotonicity in rate and packet arrivals. We will see that our suggested index policies too possess these characteristics of the optimal policy.

7.1 Monotonicity in Data Queue Lengths

The penalty accrued for each user in a given time slot is an increasing function of its current data queue length. Hence we expect the likelihood of the optimal policy serving any given user to increase with its queue length, i.e., if the optimal policy serves a user u in the state $[i, y, a]$, it will serve user u in state $[i, y + e_u, a]$ as well, where e_u denotes an N-dimensional vector with the u^{th} component 1 and all other components 0. Theorem 7.1.2 states and proves this monotonicity of the optimal policy for the total discounted cost.

Let $V(i, y, a)$ be the total discounted cost starting in state (i, y, a). For notational convenience, let

$$h(i, y, a) = \sum_{j,b} p_{ij} h_{ab} V(j, y, b). \tag{7.1}$$

Then following equation 6.9, the standard Bellman equation for the total discounted cost model with a discounting rate α is

$$V(i,y,a) = \min_{u=1,2,\ldots,N} \left[W_u(i,y+a) + \alpha h(i,(y+a,i)^u,a)\right], \tag{7.2}$$

where, $(y+a,i)^u$ is as in (6.7). Let $dec(i,y,a) \in \mathcal{A}$ be the optimal decision made (i.e., the user served) in state $[i,y,a]$. Then,

$$dec(i,y,a) = \arg\min_{u=1,2,\ldots,N} \left[W_u(i,y+a) + \alpha h(i,(y+a,i)^u,b)\right]. \tag{7.3}$$

For a given queue length vector $y \in Z^N$, a fixed positive integer t and fixed $u \in \mathcal{A}$, define ${}_ty^u \in Z^N$ by

$${}_ty^u = [y_1, y_2, \ldots, y_{u-1}, (y_u - t)^+, y_{u+1}, \ldots, y_N]. \tag{7.4}$$

We will need the following lemma to prove theorem 7.1.2.

Lemma 7.1.1. *Let r, s be fixed positive integers. Then*

$$V(j, {}_t(y+e_v)^v, b) - V(j, {}_ty^v, b) - V(j, {}_s(y+e_v)^u, b) + V(j, {}_sy^u, b) \leq 0. \tag{7.5}$$

Proof. Clearly, without loss of generality, we can choose $v = 1$ and $u = 2$ and hence it is enough to prove that

$$V(j, {}_t(y+e_1)^1, b) - V(j, {}_ty^1, b) - V(j, {}_s(y+e_1)^2, b) + V(j, {}_sy^2, b) \leq 0 \tag{7.6}$$

We prove (7.6) below using a technique inspired by Chen and Kulkarni [51] and based on the coupling method [52, 53].

To make the exposition clear, we use the following self-explanatory notations for the four system states in (7.5):

$$\begin{aligned}
(j, {}_t(y+e_1)^1, b) &= (j; (y_1 - t + 1)^+, y_{-1}; b), \\
(j, {}_ty^1, b) &= (j; (y_1 - t)^+, y_{-1}; b), \\
(j, {}_s(y+e_1)^2, b) &= (j; y_1 + 1, {}^sz_{-1}; b),
\end{aligned}$$

$$(j, {}_sy^2, b) \;=\; (j; y_1, {}^sz_{-1}; b) \tag{7.7}$$

where,

$$\begin{aligned} y_{-1} &= [y_2, \ldots, y_N] \\ {}^sz_{-1} &= [(y_2 - s)^+, y_3 + a_3, \ldots, y_N + a_N]. \end{aligned} \tag{7.8}$$

For notational convenience, define

$$D^* = V(j, {}_t(y+e_1)^1, b) - V(j, {}_ty^1, b) - V(j, {}_s(y+e_1)^2, b) + V(j, {}_sy^2, b). \tag{7.9}$$

For our proof, we define processes 1 through 4 on the same state space: Process 1 starts in state $[j; (y_1 - t + 1)^+, y_{-1}; b]$, process 2 starts in state $[j; (y_1 - t)^+, y_{-1}; b]$, process 3 in state $[j; y_1 + 1, {}^sz_{-1}; b]$ and process 4 in state $[j; y_1, {}^sz_{-1}; b]$. Each of the four processes sees the same environmental states and arrivals in every time slot. Processs 2 and 3 follow the optimal policy, while we describe the (sub-optimal) policies followed by processes 1 and 4 below. We use $v(n,k)$ to denote the user served in process $k \in \{1,2,3,4\}$ in slot n. Then consider the two exhaustive cases:

***Case 1*:** $y_1 < t$. Then, ${}_t(y+e_1)^1 = {}_ty^1 = 0$. Hence, using (7.7) and (7.8) the optimal discounted cost from processes 1 and 2 are identical yielding,

$$D^* = -V(j; y_1 + 1, {}^sz_{-1}, b) + V(j; y_1, {}^sz_{-1}; b). \tag{7.10}$$

Now, to prove $D^* \le 0$, consider processes 3 and 4. Process 3 follows the optimal policy and process 4 follows the policy ϕ described below. Let $[X^n, Y^{n,k}, A^n]$ be the state of process $k \in \{3,4\}$ in time slot n.

Let τ be the first time when the data queue of user 1 in process 3 becomes empty. Then $\tau = \min\{n \ge 1 : r_{X_1^n} > Y_1^{n,3} + A_1^n, v(n,3) = 1\}$. For $0 \le n \le \tau$ the policy ϕ serves the same user as process 3, and then follows the optimal policy. Let $V^\phi(\cdot,\cdot,\cdot)$ denote the long run discounted cost under policy ϕ. Then from (7.10),

$$\begin{aligned}
D^* &= -V(j; y_1+1, {}^s z_{-1}, b) + V(j; y_1, {}^s z_{-1}; b) \leq V^{\phi}(j; y_1, {}^s z_{-1}; b) - V(j; y_1+1, {}^s z_{-1}, b) \\
&= E\sum_{n=0}^{\tau} e^{-\alpha n}\left[-W_{v(n,3)}(X^n, Y^{n,3}+A^n) + W_{v(n,3)}(X^n, Y^{n,4}+A^n)\right] \\
&+ Ee^{-\alpha(\tau+1)}\left[-V(X^{\tau+1}, Y^{\tau+1,3}, A^{\tau+1}) + V(X^{\tau+1}, Y^{\tau+1,4}, A^{\tau+1})\right].
\end{aligned} \tag{7.11}$$

For $n < \tau$ both processes serve exactly the same number of packets to any user that is chosen to serve; at τ, process 3 serves 1 additional packet than process 4 to user 1. Therefore for $n \leq \tau$, $Y_1^{n,3} = Y_1^{n,4} + 1$ and for $n > \tau$, $Y_1^{n,3} = Y_1^{n,4}$. Thus, from (6.8),

$$\begin{aligned}
&E\sum_{n=0}^{\tau} e^{-\alpha n}\left[-W_{v(n,3)}(X^n, Y^{n,3}+A^n) + W_{v(n,3)}(X^n, Y^{n,4}+A^n)\right] \\
&= \sum_{n=0}^{\tau} e^{-\alpha n}\left[K_1\left(-Y_1^{n,3} + Y_1^{n,4}\right)\right] + e^{-\alpha\tau}\left[Y_1^{\tau,3} - Y_1^{\tau,4}\right] \\
&= -K_1\sum_{n=0}^{\tau} e^{-\alpha n} + K_1 e^{-\alpha\tau} = K_1\left(e^{-\alpha\tau} - 1\right) - K_1\sum_{n=1}^{\tau} e^{-\alpha n} \leq 0,
\end{aligned} \tag{7.12}$$

Combining equations 7.9, 7.11 and 7.12, we get (7.6), as required.

Case 2: $y_1 \geq t$. Then, using (7.7) and (7.8),

$$\begin{aligned}
D^* &= V(j; y_1+1-t, y_{-1}; b) - V(j; y_1-t, y_{-1}; b) \\
&- V(j; y_1+1, {}^s z_{-1}; b) + V(j; y_1, {}^s z_{-1}; b).
\end{aligned} \tag{7.13}$$

Following the approach of Case 1, we consider the four processes 1-4. Processes 2 and 3 follow the optimal policy. Process 1 and 4 follow policies ϕ_1 and ϕ_4 respectively as described below. Let $[X^n, Y^{n,k}, A^n]$ be the state of process $k \in \{1,2,3,4\}$.

Let $\tau_k, k \in \{1,3\}$ be the first time when the queue length of user 1 in process k goes to zero. Then $\tau_3 = \min\{n \geq 1 : r_{X_1^n} > Y_1^{n,3} + A_1^n, v(n,3) = 1\}$ and $\tau_1 = \min\{n \geq 1 : r_{X_1^n} > Y_1^{n,1} + A_1^n, v(n,1) = 1\}$. Since both processes 1 and 3 see the same arrivals and environmental states,

$$Y_1^{n,3} > Y_1^{n,1} \Longrightarrow \tau_3 > \tau_1. \tag{7.14}$$

According to policy ϕ_1, process 1 serves the same user as process 2 until τ_1, then follows the optimal policy and according to policy ϕ_4, process 4 serves the same user as process 3 until τ_3, then follows the optimal policy. Then from (7.2),

$$\begin{aligned}
&D^* = V(j;y_1+1-t,y_{-1};b) - V(j;y_1-t,y_{-1},b) - V(j;y_1+1,{}^sz_{-1},b) + V(j;y_1,{}^sz_{-1};b)\\
&\leq V^{\phi_1}(j;y_1+1-t,y_{-1};b) - V(j;y_1-t,y_{-1},b)\\
&+ V^{\phi_4}(j;y_1,{}^sz_{-1};b) - V(j;y_1+1,{}^sz_{-1},b)\\
&= E\sum_{n=0}^{\tau_1} e^{-\alpha n}\Big[W_{v(n,2)}(X^n,Y^{n,1}+A^n) - W_{v(n,2)}(X^n,Y^{n,2}+A^n)\Big]\\
&+ E\sum_{n=0}^{\tau_3} e^{-\alpha n}\Big[W_{v(n,3)}(X^n,Y^{n,4}+A^n) - W_{v(n,3)}(X^n,Y^{n,3}+A^n)\Big]\\
&+ Ee^{-\alpha(\tau_1+1)}\Big[V(X^{\tau_1+1},Y^{\tau_1+1,1},A^{\tau_1+1}) - V(X^{\tau_1+1},Y^{\tau_1+1,2},A^{\tau_1+1})\Big]\\
&+ Ee^{-\alpha(\tau_3+1)}\Big[-V(X^{\tau_3+1},Y^{\tau_3+1,3},A^{\tau_3+1}) + V(X^{\tau_3+1},Y^{\tau_3+1,4},A^{\tau_3+1})\Big].
\end{aligned} \tag{7.15}$$

We follow an approach similar to Case 1 to simplify (7.15) further: For $n < \tau_1$ both processes 1 and 2 serve exactly the same number of packets to any user that is chosen to serve; at τ_1, process 1 serves 1 additional packet than process 2 to user 1. Therefore for $n \leq \tau_1$, $Y_1^{n,1} = Y_1^{n,2} + 1$ and for $n > \tau_1$, $Y_1^{n,1} = Y_1^{n,2}$. Similarly, $n \leq \tau_3$, $Y_1^{n,3} = Y_1^{n,4} + 1$ and for $n > \tau_3$, $Y_1^{n,3} = Y_1^{n,4}$. Thus, from (6.8),

$$\begin{aligned}
&E\sum_{n=0}^{\tau_1} e^{-\alpha n}\Big[\Big(-W_{v(n,2)}(X^n,Y^{n,2}+A^n) + W_{v(n,2)}(X^n,Y^{n,1}+A^n)\Big)\Big]\\
&+ E\sum_{n=0}^{\tau_3} e^{-\alpha n}\Big[\Big(-W_{v(n,3)}(X^n,Y^{n,3}+A^n) + W_{v(n,3)}(X^n,Y^{n,4}+A^n)\Big)\Big]\\
&\leq E\sum_{n=0}^{\tau_1} e^{-\alpha n}\Big[K_1\Big(-Y_1^{n,2}+Y_1^{n,1}\Big)\Big]\\
&+ E\sum_{n=0}^{\tau_3} e^{-\alpha n}\Big[K_1\Big(-Y_1^{n,3}+Y_1^{n,4}\Big)\Big] + e^{-\alpha\tau_3}\Big[Y_1^{\tau_3,3}-Y_1^{\tau_3,4}\Big]\\
&= K_1\sum_{n=0}^{\tau_1} e^{-\alpha n} - K_1\sum_{n=0}^{\tau_3} e^{-\alpha n} + K_1e^{-\alpha\tau_3} = K_1\sum_{n=0}^{\tau_1} e^{-\alpha n} - K_1\sum_{n=0}^{\tau_3-1} e^{-\alpha n}\\
&- K_1\left[\sum_{n=0}^{\tau_3-1} e^{-\alpha n} - \sum_{n=0}^{\tau_1} e^{-\alpha n}\right] \leq 0,
\end{aligned} \tag{7.16}$$

where the last inequality follows from (7.14). Combining equations 7.13, 7.15 and 7.16 we get (7.6), as required. □

We now state the main result of this chapter.

Theorem 7.1.2. *The optimal policy is monotone in data queue length, i.e., $dec(i, y, a) = v \Longrightarrow dec(i, y + e_v, a) = v$.*

Proof. Since $dec(i, y, a) = v$, using (6.8) and the remark immediately following (6.10), we have for $u \in \mathcal{A}$,

$$-K_v \min(y_v+a_v, r_{i_v})+\alpha h(i, (y+a, i)^v, a) \le -K_u \min(y_u+a_u, r_{i_u})+\alpha h(i, (y+a, i)^u, a). \quad (7.17)$$

Similarly, $dec(i, y + e_v, a) = v$ if

$$-K_v \min(y_v+a_v+1, r_{i_v})+\alpha h(i, (y+a+e_v, i)^v, a) \le -K_u \min(y_u+a_u, r_{i_u})+\alpha h(i, (y+a+e_v, i)^u, a), \quad (7.18)$$

which holds if

$$h(i, (y+a+e_v, i)^v, a)-h(i, (y+a, i)^v, a)-h(i, (y+a+e_v, i)^u, a)+h(i, (y+a, i)^u, a) \le 0. \quad (7.19)$$

Lemma 7.1.1 implies that

$$V(j, (y+a+e_v, i)^v, b)-V(j, (y+a, i)^v, b)-V(j, (y+a+e_v, i)^u, b)+V(j, (y+a, i)^u, b) \le 0 \quad (7.20)$$

by putting $t = r_{i_v}$, $s = r_{i_u}$ and replacing y by $y + a$ in (7.6). Multiplying (7.20) by $p_{ij}h_{ab}$ followed by summing over $[j, b] \in \Omega^N \times Z^N$ and using (7.1) yields (7.19), as required. □

7.2 Monotonicity in Rate and Arrivals

The MDP model in chapter 6.3 has been formulated to minimize the long run cost of holding data in the queues for all users. The holding cost in a given time slot decreases with decrease in data queue lengths, i.e., with increase in the number of packets actually served. Therefore, we expect the optimal policy to be monotone in the rate that can be potentially transferred to the users and the number of arriving packets. In particular, if the optimal policy serves a user v in state $[i, y, a]$, it should serve v in state $[i + e_v, y, a]$ and $[i, y, a + e_v]$ as well. We prove this monotonicity in rate in theorem 3.2.1 under the assumption that $\{X^n : n \geq 0\}$ are i.i.d. The monotonicity in the number of arrival packets when the arrival vector in every time slot is i.i.d. can be proved similarly. Let $f(\cdot)$ be the probability mass function of the environmental state vector $i \in \Omega^N$.

Theorem 7.2.1. *Suppose $\{X^n : n \geq 0\}$ are i.i.d. and $v \in \mathcal{A}$ is fixed. Then $dec(i, y, a) = v \Longrightarrow dec(i + e_v, y, a) = v$.*

Proof. Since $\{X^n : n \geq 0\}$ are i.i.d., we get

$$h(i, y, a) = \sum_{j:j \in \Omega^N} f(j) \sum_b h_{ab} V(j, y, b) \tag{7.21}$$

Following (6.9), the value function $V(i, y, a)$ under the optimal policy is given by

$$V(i, y, a) = \min_{u=1,2,\dots,N} \left[W_u(i, y + a) + \alpha h(i, (y + a, i)^u, a)\right]. \tag{7.22}$$

Hence for $u \in \mathcal{A}$

$$dec(i, y, a) = v \Longrightarrow [[W_v(i, y + a) - W_u(i, y + a)] + \alpha \left[h(i, (y + a, i)^v, a) - h(i, (y + a, i)^u, a)\right] \leq 0. \tag{7.23}$$

To prove $dec(i + e_v, y, a) = v$, we need to prove

$$[W_v(i + e_v, y + a) - W_u(i + e_v, y + a)] + \alpha \left[h(i + e_v, (y + a, i)^v, a) - h(i + e_v, (y + a, i)^u, a)\right] \leq 0. \tag{7.24}$$

which follows from the definition (6.8) of $W_u(\cdot, \cdot)$ and from (7.21) and (7.23) since $h(i, y, a)$ is independent of i. □

Analogous to the logic of rate monotonicity in section 3.2, we assume stochastic monotonicity of $\{X^n : n \geq 0\}$ and of $\{A^n : n \geq 0\}$ for monotonicity in age and number of arrival packets respectively. We state these monotonicity results formally for the total discounted costs case in

Conjecture 7.2.2.

Conjecture 7.2.2. *1. If the Markov chain $\{X^n : n \geq 0\}$ is stochastically monotone,*

$$dec(i, y, a) = v \Longrightarrow dec(i + e_v, y, a) = v \tag{7.25}$$

2. If the Markov chain $\{A^n : n \geq 0\}$ is stochastically monotone,

$$dec(i, y, a) = v \Longrightarrow dec(i, y, a + e_v) = v \tag{7.26}$$

Further, since the optimal policy is monotone for every value of the discounting factor α, we expect it to be monotone under the average cost criterion as well. However, unlike chapter 3, we do not have a rigorous proof of this statement.

Chapter 8

Index Policy

Theoretically, the MDP equations given by (6.10) can be solved using numerical techniques such as the standard value iteration method [38] or the PIA. However, for problems with large state space such as the one in the problem under consideration and the infinitely backlogged problem, using any of these methods becomes infeasible. Therefore, in this chapter we use the PIA to develop an easily implementable heuristic scheduling policy along the lines of the infinitely backlogged case. The standard PIA [38] in this setting is given by:

1. Let π^0 be an arbitrary policy that chooses action $\pi^0(i, y, a) \in \mathcal{A}$ in state (i, y, a). Set n $= 0$.

2. Policy Evaluation Step: For $(i, y, a) \in \Omega^N \times Z^N \times Z^N$, solve the equations

$$g_n + w_n(i, y, a) = W_u(i, y + a) + \sum_{(j,b)} p_{ij} h_{ab} w_n(j, (y + a, i)^u, b),$$

   for g_n and $\{w_n(i, y, a) : i \in \Omega^N, y \in Z^N \; a \in Z^N\}$ where $u = \pi^n(i, y, a)$ and n denotes the number of iterations so far.

3. Policy Improvement Step: Let

$$\pi^{n+1}(i, y, a) = \arg\min_{u \in \mathcal{A}} \{W_u(i, y + a) + \sum_{(j,b)} p_{ij} h_{ab} w_n(j, (y + a, i)^u, b)\}. \tag{8.1}$$

   If $\pi^n(i, y, a)$ minimizes the Right Hand Side (RHS), choose $\pi^{n+1}(i, y, a) = \pi^n(i, y, a)$.

4. If $\pi^{n+1} \neq \pi^n$, set $n = n + 1$ and go to step 2. Else, STOP. π^{n+1} is the optimal policy.

Under certain conditions [38] one can show that this algorithm terminates in a finite number of steps.

Next, following the broad direction of our approach in the infinitely backlogged problem, we derive a heuristic policy based on a single step of the PIA. As noted above using the PIA multiple times until a terminal condition is satisfied to solve the equations given by (2.10) to optimality is not feasible. Instead we use the policy π^1 (derived from one iteration of the PIA) as an approximation for the optimal policy. We shall show that π^1, after some approximations, is simple to implement and validate by simulation that it improves upon the existing max-weight algorithm. In all time slots, given the system state (i, y, a), the policy evaluation step will yield an index for each user $u \in \mathcal{A}$ solely based on its state (i_u, y_u, a_u) and the means of the packet arrival process $\{A_u^n, n \geq 0\}$ and the packet departure process $\{R_u^n, n \geq 0\}$ of user u. The heuristic policy π^1 that we recommend is to serve the user whose index is minimized. Such policies are called Index Policies [44, 45, 36]. Although we use the current Markovian structure for the evolution of the environmental state of users, the final index is independent of this Markovian structure. Further, the index depends on the number of users N in a very straightforward way that allows us to generalize the index for the case when the number of users in a cell varies due to user arrival and departure. We consider the case when the number of users is an $M/G/\infty$ queue in section 8.6. Derivation of these indices involves choosing an "appropriate" initial policy and then using one step of the PIA. We discuss such an initial policy in the next subsection. The policy is "appropriate" because as we shall see, applying one step of the PIA is analytically tractable after making some approximations.

8.1 Initial Policy

We consider the following state independent stationary policy as the initial policy π^0: Serve user u with probability q_u in any time slot, where $q_1, q_2, \ldots, q_N$ are fixed numbers such that

$$q_u > 0 \quad \forall u \in 1, 2, \ldots, N \tag{8.2}$$

$$\sum_u q_u = 1. \tag{8.3}$$

Let

$$q = [q_1, q_2, \ldots, q_N], \tag{8.4}$$

g_q be the long run cost and

$$w_q = \{w(i, y, a) : (i, y, a) \in \Omega^N \times Z^N \times Z^N\}$$

be the bias vector for this policy satisfying the equation

$$g_q + w_q(i, y, a) = \sum_{u=1}^{N} q_u \left\{ W_u(i, y + a) + \sum_{(j,b)} p_{ij} h_{ab} w_q(j, (y + a, i)^u, b) \right\}. \tag{8.5}$$

We refer to this policy as the *randomized policy* or the *q-policy.* The equation 8.5 assumes stability of the queues, i.e., $\{(X^n, Y^n, A^n), n \geq 0\}$ is positive recurrent. We describe necessary and sufficient conditions for stability of the q-policy below.

Let $\pi^u = [\pi_1^u, \ldots, \pi_M^u]$ be the steady state distribution of the Markov chain $\{X_u^n : n \geq 0\}$, $u = 1, 2, \ldots, N$. Then, since M is finite and the DTMC is irreducible, it is well known [46] that π^u exists and is the unique solution to

$$\begin{aligned} \pi^u &= \pi^u P^u \\ \sum_{m=1}^{M} \pi_m^u &= 1. \end{aligned}$$

Then we know from standard DTMC theory [46] that the long run average of $\{R_u^n, n \geq 0\}$ is given by

$$\bar{R}_u = \sum_{k=1}^{M} r_k \pi_k^u. \tag{8.6}$$

We further assume that for all $u \in \mathcal{A}$ the steady state distribution of the Markov chain $\{A_u^n : n \geq 0\}$ exists and is given by $\theta^u = [\theta_0^u, \theta_1^u, \ldots]$. Let λ_u be the long run average of the number of data packets that arrive for user u in one time slot. Then, from standard DTMC theory [46]

$$\lambda_u = \sum_{k=0}^{\infty} k \theta_k^u. \tag{8.7}$$

A straightforward extension of the results on Matrix-Geometric methods for stochastic models [54] can be used to show that the randomized q-policy is stable (i.e., each user's queue is stable) if

$$\lambda_u < q_u \bar{R}_u, \qquad u = 1, 2, \ldots, N. \tag{8.8}$$

We shall assume this in the rest of the chapter.

To get an idea of the physical significance of g_q and $w_q(i, y, a)$ let us look at the long term cost under this policy. Let $V_D^q(i, y, a)$ be the total cost in periods 0 through $D - 1$ starting in state (i, y, a) under this initial q-policy. We use $o(D)$ to denote terms that go to zero as D approaches ∞. Then using standard MDP theory [38] we have

$$V_D^q(i, y, a) = g_q D + w_q(i, y, a) + o(D). \tag{8.9}$$

Thus for a fixed q-policy and state (i, y, a) we can think of $V_D^q(i, y, a)$ as an asymptotically linear function of D with slope g_q and intercept $w_q(i, y, a)$.

Now consider the N queues $\{Y_u^n, n \geq 0\}$ for $u \in \mathcal{A}$ under the following policy: In every time slot serve user u with probability q_u independent of all other users. For $u \in \mathcal{A}$, the user u queue is fed by its independent arrival process $\{A_u^n, n \geq 0\}$ as above. We call this the q_u *policy* for user u. Under the policy q_u, let $V_D^{q_u}(i_u, y_u, a_u)$ be the total cost accumulated by user u starting from state $[X_u^0, Y_u^0, A_u^0] = [i_u, y_u, a_u]$ at time 0 over time periods $0, 1, 2, \ldots, D - 1$. The mean cost incurred by user u in time slot 0 is $q_u K_u(y_u + a_u - r_{i_u})^+ + (1 - q_u) K_u(y_u + a_u)$. Thus the standard DTMC theory [46] yields

$$\begin{aligned} V_D^{q_u}(i_u, y_u, a_u) = {} & q_u K_u(y_u + a_u - r_{i_u})^+ + (1 - q_u) K_u(y_u + a_u) \\ & + \sum_{j_u, b_u} q_u p^u_{i_u, j_u} h^u_{a_u, b_u} V_{D-1}^{q_u}(j_u, (y_u + a_u - r_{i_u})^+, b_u) \\ & + \sum_{j_u, b_u} (1 - q_u) p^u_{i_u, j_u} h^u_{a_u, b_u} V_{D-1}^{q_u}(j_u, y_u + a_u, b_u). \end{aligned} \tag{8.10}$$

Let g_{q_u} be the long run cost and

$$w_{q_u} = \{w_u(i_u, y_u, a_u) : (i_u, y_u, a_u) \in \Omega \times Z \times Z\}$$

be the bias vector for the policy q_u. It is known to satisfy the equation

$$\begin{aligned} g_{q_u} + w_{q_u}(i_u, y_u, a_u) = q_u K_u (y_u + a_u - r_{i_u})^+ & \\ + q_u \left\{ \sum_{(j_u, b_u)} p^u_{i_u j_u} h^u_{a_u b_u} w_{q_u}(j_u, (y_u + a_u - r_{i_u})^+, b_u) \right\} & \\ + (1 - q_u) \left\{ K_u(y_u + a_u) + \sum_{(j_u, b_u)} p^u_{i_u j_u} h^u_{a_u b_u} w_{q_u}(j_u, y_u + a_u, b_u) \right\}. & \end{aligned} \qquad (8.11)$$

Standard MDP theory [38] yields

$$V^{q_u}_D(i_u, y_u, a_u) = g_{q_u} D + w_{q_u}(i_u, y_u, a_u) + o(D), \qquad (8.12)$$

which is the counterpart of equation 8.9 for user u. Then the following theorem gives an intuitive and useful relation between g_q, $w_q(i, y, a)$ and g_{q_u}, $w_{q_u}(i_u, y_u, a_u)$.

Theorem 8.1.1. *Let g_q be a constant and $\{w_q(i, y, a) : i \in \Omega^N, y \in Z^N a \in Z^N\}$ a function satisfying equation 8.5. Then for all $u \in \mathcal{A}$, there exist constants g_{q_u} and functions $\{w_{q_u}(i_u, y_u, a_u) : (i_u, y_u, a_u) \in \Omega \times Z \times Z\}$ that satisfy equation 8.11 and*

$$g_q = \sum_u g_{q_u} \qquad (8.13)$$

$$w_q(i, y, a) = \sum_u w_{q_u}(i_u, y_u, a_u). \qquad (8.14)$$

Proof. We prove that equation 8.5 is consistent with equations 8.11, 8.13 and 8.14. Let g_q, $w_q(i, y, a)$, $\{g_{q_u}, u = 1, 2, \ldots, N\}$ and $\{w_{q_u}(i_u, y_u, a_u), u = 1, 2, \ldots, N\}$ satisfy equations 8.5, 8.13 and 8.14. Using (8.11) in the RHS of (8.14) and rearranging, we have

$$\begin{aligned} \sum_{u=1}^N g_{q_u} + \sum_{u=1}^N w_{q_u}(i_u, y_u, a_u) = & \sum_{u=1}^N \left[q_u K_u (y_u + a_u - r_{i_u})^+ + (1 - q_u)\left(K_u(y_u + a_u)\right) \right] \\ & + \sum_{u=1}^N q_u \left(\sum_{(j_u, b_u)} p^u_{i_u j_u} h^u_{a_u b_u} w_{q_u}(j_u, (y_u + a_u - r_{i_u})^+, b_u) \right) \\ & + \sum_{u=1}^N (1 - q_u) \left(\sum_{(j_u, b_u)} p^u_{i_u j_u} h^u_{a_u b_u} w_{q_u}(j_u, y_u + a_u, b_u) \right). \end{aligned} \qquad (8.15)$$

Algebraic rearrangement of some terms yields

$$\begin{aligned}\sum_{u=1}^{N}(1-q_u)\left[\sum_{(j_u,b_u)} p^u_{i_uj_u}h^u_{a_ub_u}w_{q_u}(j_u,y_u+a_u,b_u)\right] &= \sum_{u=1}^{N}q_u\left[\sum_{l\neq u}\sum_{(j_l,b_l)} p^l_{i_lj_l}h^l_{a_lb_l}w_{q_l}(j_l,y_l+a_l,b_l)\right], \\ \sum_{u=1}^{N}(1-q_u)\{K_u(y_u+a_u)\} &= \sum_{u=1}^{N}q_u\left[\sum_{l\neq u}\{K_l(y_l+a_l)\}\right]. \qquad (8.16)\end{aligned}$$

From (8.15) and (8.16), and using (6.5) for the definition of $W_u(i,y+a)$, we have

$$\begin{aligned}&\sum_{u=1}^{N}g_{q_u}+\sum_{u=1}^{N}w_{q_u}(i_u,y_u,a_u)=\sum_{u=1}^{N}\left[q_uW_u(i,y+a)\right] \\ &+\sum_{u=1}^{N}q_u\left[\left(\sum_{(j_u,b_u)} p^u_{i_uj_u}h^u_{a_ub_u}w_{q_u}(j_u,(y_u+a_u-r_{i_u})^+,b_u)\right)+\left(\sum_{l\neq u}\sum_{(j_l,b_l)} p^l_{i_lj_l}h^l_{a_lb_l}w_{q_l}(j_l,y_l+a_l,b_l)\right)\right] \\ &\qquad (8.17)\end{aligned}$$

Since $\forall k,m\in\mathcal{A}, k\neq m$ $w_{q_m}(\cdot,\cdot,\cdot)$ is independent of $p^k_{i_kj_k}h^k_{a_kb_k}$, and from (6.6) for the definitions of p_{ij} and h_{ab}, and (6.7) for the definition of $(y+a,i)^u$, (8.17) reduces to

$$\begin{aligned}&\sum_{u=1}^{N}g_{q_u}+\sum_{u=1}^{N}w_{q_u}(i_u,y_u,a_u) \\ &\qquad =\sum_{u=1}^{N}q_u\left[W_u(i,y+a)\right]+\sum_{u=1}^{N}q_u\left[\sum_{j,b}p_{ij}h_{ab}w_q(j,(y+a,i)^u,b)\right], \quad (8.18)\end{aligned}$$

which yields (8.5) by setting

$$\begin{aligned}g_{q_u}&=q_uK_u(y_u+a_u-r_{i_u})^+ + (1-q_u)[K_u(y_u+a_u)] \quad \text{and} \\ g_q&=\sum_{u=1}^{N}g_{q_u}, \quad w_q(i,y,a)=\sum_{u=1}^{N}w_{q_u}(i_u,y_u,a_u),\end{aligned}$$

as required. □

8.2 Policy Evaluation

The next step in deriving the improved policy is using one step of the PIA. From equation 8.1 of the PIA the policy improvement step seeks to minimize

$$\min(W_u(i, y+a) + \sum_{(j,b)} p_{ij} h_{ab} w_q(j, (y+a, i)^u, b) \tag{8.19}$$

over all $u \in \mathcal{A}$. Thus we do not need to compute g_q for deriving the improved policy. In fact, as we shall see in the next subsection, we do not need to compute $w_q(i, y, a)$ either, computing an expression for the difference $w_q(j, (y+a, i)^u, b) - w_q(j, y+a, b)$ is sufficient. Computing this is a key step in the derivation of the index since the computation of $w_q(i, y, a)$ itself is intractable.

We will need the following notations to compute $w_q(j, (y+a, i)^u, b) - w_q(j, y+a, b)$. Let

$$Z_u = \min\{m \geq 0 : Y_u^m = 0\}, \quad u = 1, 2, \ldots, N, \tag{8.20}$$

and

$$T_u(i_u, y_u, a_u) = \mathrm{E}[Z_u | X_u^0 = i_u, Y_u^0 = y_u, A_u^0 = a_u]. \tag{8.21}$$

For a stable randomized policy, $T_u(i_u, y_u, a_u) < \infty$ for all $(i_u, y_u, a_u) \in \Omega \times Z \times Z$ and all $u \in \mathcal{A}$.

Consider two sample paths $\{(X^{n,m}, Y^{n,m}, A^{n,m}), n \geq 0\}$, $m = 1, 2$, of the $\{(X^n, Y^n, A^n), n \geq 0\}$ process that are coupled as follows:

$$X^{0,1} = X^{0,2} = j, \quad A^{0,1} = A^{0,2} = a, \quad Y^{0,1} = (y+a, i)^u, \quad Y^{0,2} = y+a$$

and

$$A^{n,1} = A^{n,2}, \quad X^{n,1} = X^{n,2} \qquad \text{for } n \geq 0. \tag{8.22}$$

Let $v^m(n)$ be the user served in slot n along sample path m, and

$$v^1(n) = v^2(n) \quad \text{for } n \geq 0. \tag{8.23}$$

Now we introduce the notation

$$w_q^{\Delta}(j, y, (y+a, i)^u, b) = w_q(j, (y+a, i)^u, b) - w_q(j, y+a, b). \tag{8.24}$$

The expression in (8.24) can be thought of as the expected difference of the cost accumulated along the two sample paths $\{(X^{n,m}, Y^{n,m}, A^{n,m}), n \geq 0\}$, $m = 1, 2$ described above. The queue length trajectory along both paths is complicated and hence it is not possible to compute exact closed form expression for $w_q^{\Delta}(j, y, (y+a, i)^u, b)$. The next theorem, however, presents closed form expressions for a lower and an upper bound of $w_q^{\Delta}(j, y, (y+a, i)^u, b)$.

Theorem 8.2.1. *For a stable randomized policy,*

$$\begin{aligned} -K_u \min(r_{i_u}, y_u + a_u) T_u(i_u, (y_u + a_u - r_{i_u})^+, a_u) &\geq w_q^{\Delta}(j, y, (y+a, i)^u, b) \\ -K_u \min(r_{i_u}, y_u + a_u) T_u(i_u, y_u + a_u, a_u) &\leq w_q^{\Delta}(j, y, (y+a, i)^u, b). \end{aligned} \tag{8.25}$$

Proof. From equation 8.14,

$$\begin{aligned} w_q^{\Delta}(j, y, (y+a, i)^u, b) &= \sum_l w_{q_l}(i_l, (y+a, i)_l^u, a_l) - \sum_l w_{q_l}(i_l, y_l + a_l, a_l) \\ &= w_{q_u}(i_u, (y_u + a_u - r_{i_u})^+, a_u) - w_{q_u}(i_u, y_u + a_u, a_u) \end{aligned} \tag{8.26}$$

Therefore, we only need to consider user u for computing $w_q^{\Delta}(j, y, (y+a, i)^u, b)$. Now consider path 1 and path 2. We plot the queue lengths of user u along these paths in figure 8.1. We define Z_u^m, $m = 1, 2$ to be the first time when the queue length of user u in path m goes to zero. Then

$$Z_u^1 = Z_u(j_u, (y_u + a_u - r_{i_u})^+, a_u), \quad Z_u^2 = Z_u(j_u, y_u + a_u, a_u), \tag{8.27}$$

$$T_u^1 = T_u(j_u, (y_u + a_u - r_{i_u})^+, a_u), \quad T_u^2 = T_u(j_u, y_u + a_u, a_u). \tag{8.28}$$

Let C_n^m be the cost incurred by user u along path m in time slot n and let $C_n = C_n^1 - C_n^2$ be

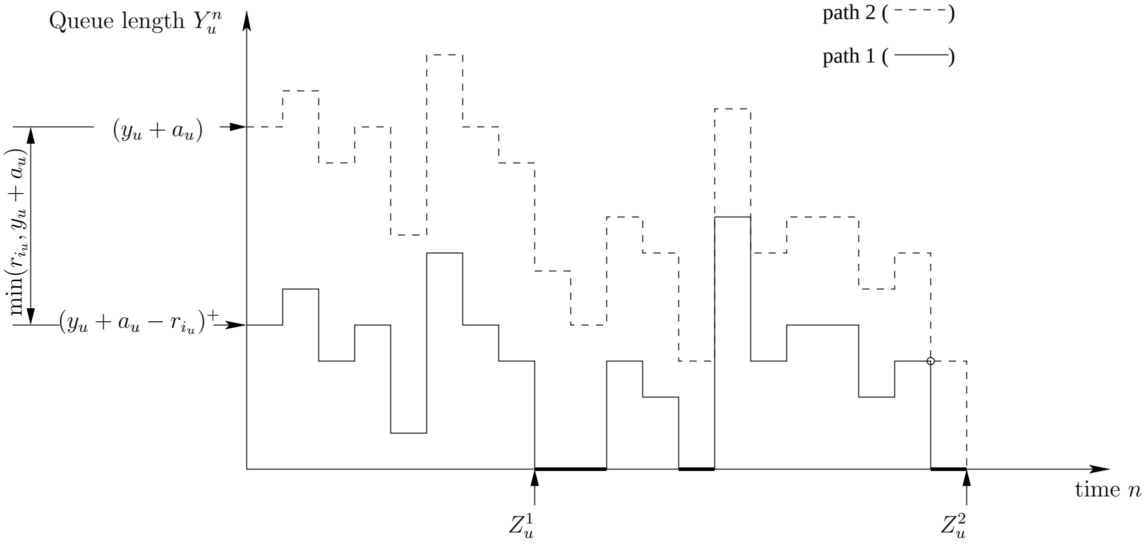


Figure 8.1: Queue lengths for user u along paths 1 and 2 for $0 \leq n \leq Z_u^2$. The queue length remains the same during a time slot. The difference $Y_u^{n,2} - Y_u^{n,1}$ remains equal to $\min(r_{i_u}, y_u + a_u)$ for $0 \leq n \leq Z_u^1$ because equal amount of data can be served along both paths. For $Z_u^2 > n > Z_u^1$, $0 \leq Y_u^{n,2} - Y_u^{n,1} \leq \min(r_{i_u}, y_u + a_u)$. The difference $Y_u^{n,2} - Y_u^{n,1}$ goes down in every slot in which there isn't enough data to serve for user u along path 1. For $n \geq Z_u^2$, $Y_u^{n,2} = Y_u^{n,1}$ and are therefore not shown in the figure.

the corresponding difference in costs. Then from equation 8.26,

$$w_q^{\Delta}(j, y, (y+a, i)^u, b) = E\left(\sum_{n \geq 0} [C_n]\right). \tag{8.29}$$

It is easy to see from the coupling of paths in equations 8.22, 8.22 and 8.23 and figure 8.1 that

$$C_n = 0, \quad n \geq Z_u^2 \tag{8.30}$$

Using equations 8.26 through 8.28 and 8.30 in equation 8.29 we get

$$w_q^{\Delta}(j, y, (y+a, i)^u, b) = E\left(\sum_{n < Z_u^2} C_n\right) \tag{8.31}$$

Clearly,

$$C_n = K_u(Y_u^{n,1} - Y_u^{n,2}). \tag{8.32}$$

Consider the queue lengths $Y_u^{n,1}$ and $Y_u^{n,2}$ in paths 1 and 2 respectively. Both get the same number of data packets in every time slot and serve the same number of packets to user u whenever enough data is available in both the queues. Thus for $n \leq Z_u^1$ the difference $Y_u^{n,2} - Y_u^{n,1}$ remains the same as $Y_u^{0,2} - Y_u^{0,1}$. For $Z_u^2 > n > Z_u^1$, the difference $Y_u^{n,2} - Y_u^{n,1}$ remains the same as $Y_u^{n-1,2} - Y_u^{n-1,1}$ when either user u is not served or user u is served and $Y_u^{n,1} \geq r_{X_u^{n,1}}$, $Y_u^{n,2} \geq r_{X_u^{n,2}}$. Also $Y_u^{n,2} - Y_u^{n,1} < Y_u^{n-1,2} - Y_u^{n-1,1}$ if $Y_u^{n,1} < r_{X_u^{n,1}}$. Thus as indicated in figure 8.1,

$$\begin{aligned} Y_u^{n,2} - Y_u^{n,1} &= \min(r_{i_u}, y_u + a_u), \quad n \leq Z_u^1, && (8.33)\\ Y_u^{n,2} - Y_u^{n,1} &< \min(r_{i_u}, y_u + a_u), \quad Z_u^2 > n > Z_u^1. && (8.34) \end{aligned}$$

Using equations 8.32, 8.33 and 8.34 in equation 8.31, and the definitions 8.21, (8.27) and (8.28) of T_u^1 and T_u^2 respectively yields (8.25), as required. □

Having computed the bias difference $w_q^{\Delta}(j, y, (y+a, i)^u, b)$, we now derive the one-step improved policy in the next subsection.

8.3 Policy Improvement Step

In this subsection we apply one step of the PIA and use some further approximations to derive an index for every user. As we shall see the index for any user $u \in \mathcal{A}$ depends only on the state of that user, and the values of these indices completely determine the one-step improved policy. Now minimizing the expression in equation 8.19 over all $u \in \mathcal{A}$ is equivalent to minimizing

$$\begin{aligned} I_u^{'''}(i, y, a) = &W_u(i, y + a) \\ &- \sum_l K_l(y_l + a_l) + \sum_{(j,b)} p_{ij} h_{ab} \left[w_q(j, (y + a, i)^u, b) - w_q(j, y + a, b) \right], \end{aligned} \tag{8.35}$$

over all $u \in \mathcal{A}$ since for a given (i, y, a), the additional term

$$-\sum_l K_l(y_l + a_l) + \sum_{(j,b)} p_{ij} h_{ab} w_q(j, y + a, b)$$

does not depend on u. The improved policy (our recommended policy) then serves the user with the highest index $I_u^{'''}(i, y, a)$. The next theorem gives an upper and lower bound for $I_u^{'''}(i, y, a)$.

Theorem 8.3.1. *$I_u^{'''}(i, y, a)$ given by (8.35) is a function only of the state of user u, i.e.,*

$$I_u^{'''}(i, y, a) = I_u^{''}(i_u, y_u, a_u), \tag{8.36}$$

and,

$$-K_u \min(r_{i_u}, y_u + a_u) T_u(i_u, y_u, a_u) \geq I_u^{''}(i_u, y_u, a_u) \geq -K_u \min(r_{i_u}, y_u + a_u) T_u(i_u, y_u + r_{i_u}, a_u). \tag{8.37}$$

Proof. Using (6.8) for the definition of $W_u(i, y + a)$ and (8.35) we have

$$I_u^{'''}(i, y, a) = -K_u \min(r_{i_u}, y_u + a_u) + \sum_{(j,b)} p_{ij} h_{ab} \left[w_q(j, (y + a, i)^u, b) - w_q(j, y + a, b) \right] \tag{8.38}$$

Thus equations 8.25 and 8.38 yield,

$$\begin{aligned} -K_u \min(r_{i_u}, y_u + a_u) \left[1 + \sum_{(j_u, b_u)} p_{i_u j_u}^u h_{a_u b_u} T_u(j_u, (y_u + a_u - r_{i_u})^+, a_u) \right] &\geq I_u^{'''}(i, y, a) & (8.39) \\ -K_u \min(r_{i_u}, y_u + a_u) \left[1 + \sum_{(j_u, b_u)} p_{i_u j_u}^u h_{a_u b_u} T_u(j_u, y_u + a_u, a_u) \right] &\leq I_u^{'''}(i, y, a). & (8.40) \end{aligned}$$

Further, we know that $\{(X_u^n, A_u^n), n \geq 0\}$ is a DTMC with state space $\Omega \times Z$ because both the components $\{X_u^n, n \geq 0\}$ and $\{A_u^n, n \geq 0\}$ are independent DTMC's with P^u and Q^u as the TPM's respectively. Therefore using standard DTMC theory [46]

$$\begin{aligned} T_u(i_u, y_u, a_u) &= 1 + \sum_{(j_u, b_u)} p_{i_u j_u}^u h_{a_u b_u} T_u(j_u, (y_u + a_u - r_{i_u})^+, a_u) & (8.41) \\ T_u(i_u, y_u + r_{i_u}, a_u) &= 1 + \sum_{(j_u, b_u)} p_{i_u j_u}^u h_{a_u b_u} T_u(j_u, y_u + a_u, a_u). & (8.42) \end{aligned}$$

Using (8.41) in (8.39) and (8.42) in (8.40) yields (8.37) and (8.36), as required. □

Thus we know the upper and lower bound for the index $I_u^{''}(i_u, y_u, a_u)$. One can use these bounds to define the Average Index (AI) given by

$$I_u^{'}(i_u, y_u, a_u) = K_u \min(r_{i_u}, y_u + a_u) \left[T_u(i_u, y_u + r_{i_u}, a_u) + T_u(i_u, y_u, a_u)\right] \tag{8.43}$$

Note that $I_u^{'}(i_u, y_u, a_u)$ has been obtained by taking the average of the upper and lower bound of $I_u^{''}(i_u, y_u, a_u)$ and multiplying the result by -1. Hence whereas the index policy based on $I_u^{'''}(i_u, y_u, a_u)$ is to serve the user u for whom $I_u^{''}(i_u, y_u, a_u)$ is *minimized*, the index policy based on $I_u^{'}(i_u, y_u, a_u)$ is to serve the user u for whom $I_u^{'}(i_u, y_u, a_u)$ is *maximized*. We will see in the simulation results that this index policy performs better than the MWA for the entire traffic range.

Now the state space of the DTMC $\{(X_u^n, Y_u^n, A_u^n), n \geq 0\}$ is $\Omega \times Z \times Z$. Exact values of the first passage times $T_u(i_u, y_u + r_{i_u}, a_u)$ and $T_u(i_u, y_u, a_u)$ can be obtained only by numerically solving a set of countably infinite equations [46] (first passage time equations). Practically they can be solved by truncating the size of the buffer Y_u^n. However that will not give us closed form expressions for the first passage times which we need to compute our index in (8.43). Hence, we use the following to approximate the first passage times. Consider the queue of user u. Under the randomized policy, on an average, $q_u \bar{R}_u$ packets depart from and λ_u packets join the queue per time slot. Thus, on an average, the number of net departures per time slot is $q_u \bar{R}_u - \lambda_u$. Hence, if there are y_u packets in the queue at time 0, the expected time when the queue length Y_u^n first hits zero can be approximated by

$$T_u(i_u, y_u, a_u) \approx \frac{y_u}{q_u \bar{R}_u - \lambda_u}. \tag{8.44}$$

Using the approximation (8.44) yields the following expressions for AI:

$$I_u(i_u, y_u, a_u) = \frac{K_u}{q_u \bar{R}_u - \lambda_u}(2y_u + r_{i_u}) \min(r_{i_u}, y_u + a_u). \tag{8.45}$$

Further, in the simulation results presented in section 9.2 we assume all users are stochastically identical (customers of the same “rate plan” for a wireless operator can be approximated to be stochastically identical, for instance). Since K_u, q_u, $\bar{R}_u$ and λ_u don’t depend on u for all stochastically identical users $u = 1, 2, \ldots, N$, our recommended index further reduces to

$$I_u(i_u, y_u, a_u) = (2y_u + r_{i_u}) \min(r_{i_u}, y_u + a_u). \tag{8.46}$$

The general expression for the index in (8.45) immediately begs the question as to the choice of numbers $q_u, u \in \mathcal{A}$. There are two alternatives:

1. Choose $q_u = 1/N$. Substituting this in (8.45) we get

   $$I_u(i_u, y_u, a_u) = \frac{K_u}{\bar{R}_u/N - \lambda_u}(2y_u + r_{i_u}) \min(r_{i_u}, y_u + a_u). \tag{8.47}$$

   as the expression for the finally recommended index. We call $I_u(i_u, y_u, a_u)$ of equation 8.47 the Uniform Index (UI). We present results corresponding to UI to make the scheduling decision in section 9.2.3.

2. Determine an expression for g_q in equation 8.5 using the average of costs. Then choose the optimal q^* that satisfies equations 8.2 through 8.4 and minimizes g_q. We know from past experience [44, 45, 36] that the choice of initial policy doesn't affect the performance of the policy governed by the index obtained from one step of the PIA. Therefore we don't pursue this approach in this chapter and refer the reader to [55] for details of this approach in a similar setting. It is also important to note that in the case of stochastically identical users, $q_u^* = 1/N, u \in \mathcal{A}$.

We can now describe our Index Policy (IP) as follows: In time slot n, if the state of the system $(X^n, Y^n, A^n) = (i, y, a)$, for each user u compute the index $I_u(i_u, y_u, a_u)$ in equation 8.45 and serve the user for whom it is maximized, i.e., if $v_{IP}(n)$ is the user served in time slot n according to the IP, then

$$\text{IP}: \quad v_{IP}(n) = \arg\max_u \frac{K_u}{q_u \bar{R}_u - \lambda_u} \min(R_u^n, Y_u^n + A_u^n) \left[2Y_u^n + R_u^n\right]. \tag{8.48}$$

In particular we use the UI given by (8.47) to yield the Uniform Index Policy (UIP). Thus, if

$v_{UIP}(n)$ is the user served in the n^{th} time slot according to the UIP, then

$$\text{UIP}: \quad v_{UIP}(n) = \arg\max_u \frac{K_u}{\bar{R}_u/N - \lambda_u} \min(R_u^n, Y_u^n + A_u^n) \left[2Y_u^n + R_u^n\right]. \tag{8.49}$$

It is immediately clear from equations 8.48 and 8.49 that both the general IP and UIP satisfy theorem 7.1.2 and conjecture 7.2.2 and are therefore monotone in data queue length, available data rate and number of arriving packets in the way described in chapter 3. Thus, as in the infinitely backlogged case, we can conclude, at least on the basis of monotonicity, that these index policies are reasonable surrogates for the optimal policy.

8.4 Stability of the Index Policy

A desirable condition that should be satisfied by any policy implementable in practice is its stability, namely the queue lengths do not grow with time. The randomized policy is stable under conditions given by (8.8). If we had an exact expression for the index $I_u^{''}(i_y, y_u, a_u)$, the Exact Index Policy (EIP) given by

$$v(n) = \arg\min_u I_u^{''}(i_y, y_u, a_u) \tag{8.50}$$

would be a policy that improves upon the randomized policy. Therefore, EIP would be stable under conditions (8.8). However, since it is infeasible to compute $I_u^{''}(i_y, y_u, a_u)$ precisely, we use the approximate index of (8.45) and hence need to prove the stability of the IP. We state the stability conditions and prove the stability of the IP in the next theorem. We use a well developed theory of stability using Lyapunov drift [56, 57, 58, 14].

Theorem 8.4.1. *If equation 8.8 holds for all $u \in \mathcal{A}$, the IP is stable.*

Proof. We follow the technique similar to the one used in [32]. Define a Lyapunov function $G(Y)$ on the set of queue length vectors $Y \in Z^N$ as follows: $G(Y) = \sum_{u=1}^{N} \delta_u Y_u^2$. Then it is enough to prove that $G(Y)$ has negative drift except in a finite set $\Lambda \in Z^N$ for some $\delta_u > 0$ $\forall u \in \mathcal{A}$.

We choose

$$\delta_u = \frac{K_u}{q_u \bar{R}_u - \lambda_u}. \tag{8.51}$$

Then from (8.8) $\delta_u > 0$ for $u = 1, 2, \ldots, N$. Recall that

$$v(n) = \arg\max_u \delta_u \min(R_u^n, Y_u^n + A_u^n) \left[2Y_u^n + R_u^n\right]. \tag{8.52}$$

For $u = 1, 2, \ldots, N$ define

$$J_u^n = \min(R_u^n, Y_u^n + A_u^n)\mathbf{1}(u = v(n)), \tag{8.53}$$

where $\mathbf{1}(u = v(n)) = 1$ if $u = v(n)$ and 0 otherwise. Therefore, (6.4) implies

$$Y_u^{n+1} = Y_u^n + A_u^n - J_u^n. \tag{8.54}$$

Squaring (8.54), then adding and subtracting $R_u^n(A_u^n - J_u^n)$ on the Right Hand Side (RHS) of the resulting equation and finally multiplying both sides by δ_u yields

$$\begin{aligned} \delta_u(Y_u^{n+1})^2 - \delta_u(Y_u^n)^2 = \delta_u J_u^n(R_u^n + J_u^n) - \delta_u A_u^n(R_u^n + J_u^n) + \delta_u(A_u^n)^2 - \delta_u A_u^n J_u^n \\ + \delta_u(2Y_u^n + R_u^n)(A_u^n - J_u^n) \end{aligned} \tag{8.55}$$

Let $R^* = \max_u r_u$. Clearly $J_u^n \leq R^*$, and since $A_u^n, R_u^n, J_u^n \geq 0$, summing equation 8.55 over all $u \in \mathcal{A}$ we get

$$\sum_u \left[\delta_u(Y_u^{n+1})^2 - \delta_u(Y_u^n)^2\right] \leq \sum_u \delta_u \left[2(R^*)^2 + (A_u^n)^2\right] + \sum_u \left[\delta_u(2Y_u^n + R_u^n)(A_u^n - J_u^n)\right]. \tag{8.56}$$

Now let ζ_u^n be the number of packets served to user u and $v_q(n)$ be the user served in time slot n under the randomized policy. Then

$$\begin{aligned} \sum_u \delta_u(2Y_u^n + R_u^n)J_u^n &= \delta_{v(n)}(2Y_{v(n)}^n + R_{v(n)}^n)\min(R_{v(n)}^n, Y_{v(n)}^n + A_{v(n)}^n) && (8.57) \\ \sum_u \delta_u(2Y_u^n + R_u^n)\zeta_u^n &= \delta_{v_q(n)}(2Y_{v_q(n)}^n + R_{v_q(n)}^n)\min(R_{v_q(n)}^n, Y_{v_q(n)}^n + A_{v_q(n)}^n) && (8.58) \\ \sum_u \delta_u(2Y_u^n + R_u^n)J_u^n &\geq \sum_u \delta_u(2Y_u^n + R_u^n)\zeta_u^n, && (8.59) \end{aligned}$$

where (8.59) follows from (8.57), (8.58), (8.52) and (8.53). Equation 8.59 implies

$$\begin{aligned}\sum_u \left[\delta_u (Y_u^{n+1})^2 - \delta_u (Y_u^n)^2\right] &\leq \sum_u \delta_u \left[2(R^*)^2 + (A_u^n)^2\right] + \sum_u \left[\delta_u (2Y_u^n + R_u^n)(A_u^n - \zeta_u^n)\right] \\ &\leq \sum_u \delta_u \left[2(R^*)^2 + (A_u^n)^2 + R_u^n A_u^n\right] - \sum_u \left[\delta_u (2Y_u^n)(\zeta_u^n - A_u^n)\right]\end{aligned} \tag{8.60}$$

Since $\{A_u^n, n \geq 0\}$ and $\{R_u^n, n \geq 0\}$ are independent of each other, $\mathrm{E}[\zeta_u^n] = q_u \bar{R}_u$ and $\mathrm{E}[A_u^n] = \lambda_u$, it follows from (8.60) that

$$\mathrm{E}\left[\sum_u \delta_u (Y_u^{n+1})^2 - \sum_u \delta_u (Y_u^n)^2 | Y^n\right] \leq \omega - 2\sum_u \left[Y_u^n \delta_u (q_u \bar{R}_u - \lambda_u)\right], \tag{8.61}$$

where the constant $\omega = 2(R^*)^2 \sum_u \delta_u + \sum_u \delta \mathrm{E}(A_u^n)^2 + \sum_u \delta_u \bar{R}_u \lambda_u$. Furthermore, letting $K^{\#} = \min_u K_u$ and using (8.51) in (8.61) we have

$$\mathrm{E}\left[\sum_u \delta_u (Y_u^{n+1})^2 - \sum_u \delta_u (Y_u^n)^2 | Y^n\right] \leq \omega - 2K^{\#} \sum_u Y_u^n. \tag{8.62}$$

Thus we have proved that for any $\alpha > 0$, the expected drift $\mathrm{E}\left[\sum_u \delta_u (Y_u^{n+1})^2 - \sum_u \delta_u (Y_u^n)^2 | Y^n\right] < -\alpha$ except in the finite set Λ given by

$$\Lambda = \left\{Y^n \in Z^N : \sum_u Y_u^n \leq \frac{\omega + \alpha}{2K^{\#}}\right\}, \tag{8.63}$$

as required. □

Thus we have proved that the IP determined by the initial q-policy is stable if $q_u \bar{R}_u > \lambda_u$ for all $u = 1, 2, \ldots, N$. In particular, the UIP is stable if for $u \in \mathcal{A}$

$$\bar{R}_u > N\lambda_u. \tag{8.64}$$

8.5 Aggregate Stability Condition

The stability conditions considered so far require $q_u \bar{R}_u > \lambda_u$ to hold for all $u = 1, 2, \ldots, N$ for the chosen q-policy. We refer to this entire set of N inequalities as the Individual Stability

Conditions (ISCS). Now consider instead the Aggregate Stability Condition (ASC) given by

$$\sum_u \frac{\lambda_u}{\bar{R}_u} < 1. \tag{8.65}$$

The ASC is clearly less restrictive than the ISCS, because any q-policy that satisfies the ISCS will satisfy (8.65). The ASC is also more useful in real life applications than the ISCS because it doesn't depend on the choice of the initial q-policy (which can be arbitrary). However, since we assume the ISCS to prove the stability of the IP, using the ASC is possible only if it ensures the ISCS for the chosen q-policy. We address this concern in the theorem below.

Lemma 8.5.1. *If the ASC (8.65) holds, $\exists$ an initial q-policy that satisfies the ISCS (8.8) for all $u \in \mathcal{A}$.*

Proof. Let $\kappa_u = \lambda_u/\bar{R}_u$. Then equation 8.65 implies

$$\sum_u \kappa_u < 1. \tag{8.66}$$

Consider the q-policy given by

$$q_u = \frac{\kappa_u}{\sum_u \kappa_u}. \tag{8.67}$$

Clearly this is a valid q-policy satisfying (8.2) and (8.3). Further, for all $u \in \mathcal{A}$, using equation 8.67 we have

$$q_u \bar{R}_u = \frac{\lambda_u}{\sum_u \kappa_u}, \tag{8.68}$$

which reduces to the ISCS (8.8) using equation 8.66. □

It should be noted here that under the assumption of stochastically identical users, κ_u is the same for every user and the initial q-policy (8.67) reduces to $q_u = 1/N$.

8.6 Variable Number of Users

In the analysis so far, no user is allowed to arrive in or depart from the cell. We call such a cell '*static*'. Next we describe a version of IP for the '*dynamic*' cell which allows for both user arrivals and departures.

In this section we allow existing users to leave and new users to join the cell. We assume users arrive according to a Poisson process with rate λ. Once in the cell, sojourn time of a user is generally distributed with mean a. Thus, in steady state the number of users is a Poisson random variable with mean $N_{\text{avg}} = \lambda a$. Let $N(n)$ be the number of users in the cell in time slot n. Then $\{N(n), n \geq 0\}$ is the number of customers in an $M/G/\infty$ queue and the index given by equation 8.47 is modified for this case as follows:

$$I_u(i_u, y_u, a_u) = \frac{K_u}{\bar{R}_u/N(n) - \lambda_u} \min(r_{i_u}, y_u + a_u) \left[2y_u + r_{i_u}\right]. \tag{8.69}$$

However, since we only look at the case of stochastically identical users in section 9.2, the UIP in this case reduces to maximizing $I_u(i_u, y_u, a_u)$ given by equation 8.46 over all users that are present in the cell. The stability of this system is guaranteed for any policy in a dynamic cell. This is easy to see because every user leaves the cell eventually bringing the corresponding queue length to zero.

In the next chapter we discuss the performance measures used to compare our recommended Index Policies and the MWA.

Chapter 9

Performance Analysis

9.1 Introduction

In this chapter we compare our algorithm to the MWA. First we show that the space and time complexity of the index policies and the MWA are the same. The state vector of a user at time n in the MWA algorithm is given by $[R_u^n, Y_u^n, A_u^n]$. The computation of the MWA index of equation 6.1 takes a constant time. The actual scheduling step involves a maxima over N entities, which is a log$N$ operation. Thus the time complexity of the MWA is constant + log$N$ while space complexity is $3N$. For the index policy (that we recommend and test), the state vector of a user is again given by $[R_u^n, Y_u^n, A_u^n]$. From equation 8.47 we see that this index can be computed in a constant time. The actual scheduling step, as in MWA, involves taking a maxima over N indices, which is a log$N$ operations. Thus the time complexity of each of the index policy UIP (as well as the more general, IP) is constant + log$N$ while space complexity is $3N$, which is the same as that of the MWA.

The following performance measures are of interest to us:

$$\begin{aligned} B &= \text{long run expected throughput per time slot,} \\ \xi &= \text{long run expected sum of queue lengths of all users.} \end{aligned}$$

Now $B = \sum_u \lambda_u$ for any stable policy and since both the MWA and UIP are stable they will achieve the same B. Hence this performance measure does not distinguish between the two policies. Therefore we concentrate on ξ. Clearly ξ measures user satisfaction because greater the value of ξ, longer is the wait for users in getting their data. Therefore it is a good measure of the Quality of Service (QoS) to users. In this and the next section “QoS level” refers to the value of ξ. We compare the UIP with MWA by comparing their achieved ξ values for a wide range of arriving packet load (λ_u). In this section and the next, we assume that all the users have the same TPM P with limiting distribution $\pi = [\pi_1, \ldots \pi_M]$, and that $\{X^n : n \geq 0\}$ is aperiodic. Furthermore, we assume that $K_u = K$ for all $u \in 1, 2, \ldots, N$ yielding $q_u^* = \frac{1}{N}$ for IP, since in this setting, as mentioned in section 8.3, $q_u = \frac{1}{N}$ optimizes the average cost of the randomized policy.

9.2 Simulation Results

We now use simulation to estimate ξ for the UIP and the MWA since it is hard to do so analytically. The code for the simulation has been written using the C programming language. We begin by formulating the estimators for these parameters below. These are the same as in [55] but are reproduced here for ready reference.

9.2.1 The Estimators

In this and the following sections we use $\hat{\xi}$ to denote the estimator of ξ. We consider the static case first; i.e., the number of users in the cell is a constant N. Let L be the number of independent sample paths simulated and T be the number of slots in each path. Let $Y_u^{n,l}$ be queue length of the user u in the n^{th} slot in sample path l, $l = 1, 2, \ldots, L; n = 1, 2, \ldots, T$. Then we define the estimator $\hat{\xi}$ of ξ as

$$\hat{\xi} = \frac{1}{L} \sum_{l=1}^{L} \left[\frac{1}{T} \sum_{n=1}^{T} \left(\sum_{u=1}^{N} Y_u^{n,l} \right) \right]. \tag{9.1}$$

In the *dynamic* cell the estimator $\hat{\xi}$ is obtained by replacing N with $N(n)$ $(n = 1, 2, \ldots, T)$ and at the same time excluding contribution from the time slot n if $N(n) = 0$. Thus the estimator for the *dynamic* cell case is given by

$$\hat{\xi} = \frac{1}{L} \sum_{l=1}^{L} \left[\frac{1}{T} \sum_{n=1, N(n)>0}^{T} \left(\sum_{u=1}^{N(n)} Y_u^{n,l} \right) \right]. \tag{9.2}$$

We further note that all these estimators are for long-run performance measures, and so we collect samples only from the stationary region of the Markov chain $\{X^n, n \geq 0\}$ (and of $\{N(n), n \geq 0\}$ in the dynamic population case). To help in this, both in the constant and dynamic population case, we start the simulation in the stationary distribution of the $\{X^n, n \geq 0\}$ and the $\{N(n), n \geq 0\}$ process and then we discard some initial steps (simulation warmup period) in each sample path before starting to collect the data.

9.2.2 Simulation Parameters

We use the following set of available data rates (kbps) [33]: r ={1, 2, 3, 4, 5, 8, 16, 24, 32, 48, 64}. Thus, each Markov chain $\{X_u^n, n \geq 0\}$ has $M = 11$ states. We run every path for 10^6 time slots and discard the contribution of the warmup period of the first $5 * 10^5$ time slots for the static as well as dynamic cell simulation. Thus $T = 5 * 10^5$ in equations 9.1 and 9.2. As in [55], we use the following TPM P of the environment Markov chain $\{X_u^n, n \geq 0\}$ for all users $u \in \mathcal{A}$:

$$P = \gamma I + \frac{1-\gamma}{M-1}(D - I),$$

where I is an $M \times M$ identity matrix and D is an $M \times M$ matrix with all entries equal to 1. This implies that the Markov chain stays in a given state for $\mathit{Geometric}(1-\gamma)$ number of time slots and then moves to one of the remaining $M - 1$ states with equal probability. Note that P is doubly stochastic, and hence $\pi_k = 1/M = 1/11$ yielding $\bar{R}_u = 18.8$ for all $u = 1, 2, \ldots, N$. Further, the length of a time slot is $1.67 * 10^{-3}$ seconds [10]. We choose $\gamma = 0.9999$ implying that on the average the Markov chain stays in one state for 10^4 time slots, i.e., 16.7 seconds, before changing states.

9.2.3 Constant Number of Users

We take samples from $L = 100$ sample paths of the Markov chain $\{X_u^n, n \geq 0\}$. Since all users are assumed stochastically identical, we use the same $\lambda_u = \lambda^p$ for all $u = 1, 2, \ldots, N$. In the notation λ^p we use the subscript p to indicate the mean arrival rate of packets. We use λ in sections 8.6 and 9.2.4 to denote the arrival rate of users themselves. We vary λ^p from 0.1 to 1.7 and report the results in table 9.1.

λ^p	ξ for UIP	ξ for MWA	abs Imp	% Imp
0.1	1.77	1.98	0.21	10.61
0.25	177.79	249.81	72.02	28.83
0.5	6320.02	6427.75	107.73	1.68
0.75	26245.1	26385.78	140.68	0.53
1.0	59211.02	59384.51	173.49	0.29
1.1	78763.46	78945.77	182.31	0.23
1.25	105715.01	105906.88	191.87	0.18
1.5	158305.29	158503.72	198.43	0.13
1.6	186283.35	186486.07	202.72	0.11
1.7	209638.7	209847.32	208.62	0.10

Table 9.1: Performance of the index policies and the MWA in the static cell: ξ for UIP and MWA, absolute improvement (abs Imp) and percentage improvement (% Imp) of UIP with respect to the MWA.

We report the mean queue length (ξ) for both UIP and MWA, the absolute improvement (abs Imp = ξ for MWA - ξ for UIP) and percentage improvement (% Imp = 100*(ξ for MWA - ξ for the UIP)/ξ for MWA) of UIP with respect to the MWA. From the Table 9.1 we see that the UIP performs better (lower ξ) than the MWA across the entire range of λ^p from 0.1 (low traffic) to 1.7 (high traffic). The % Imp is significantly better in the low traffic regime with a percentage reduction in ξ as high as 28.83 % for $\lambda^p = 0.25$. The similar performance for heavy traffic regime is expected because in heavy traffic $\min(r_{i_u}, y_u + a_u) = y_u + a_u$ for most time slots. Thus the UI given by equation 8.46 becomes quite similar to the MWA index given by equation 6.1. It is also worth noting, however, that although the % Imp decreases with increase in λ^p (except from 0.1 to 0.25), the absolute improvement increases monotonically with λ^p.

9.2.4 Poisson Arrival of Users

In this section, we assume users arrive according to a Poisson process with rate λ. Once in the cell, sojourn time of a user is exponentially distributed with mean a. Thus, in steady state the number of users is a Poisson random variable with mean $N_{\text{avg}} = \lambda a$. We choose $a = 1$ minute and $N_{\text{avg}} = 10$. The index for our recommended index policy is given by equation 8.69. We serve the user with the maximum value of the chosen index. The parameters of the user arrival process remain the same for both the MWA and the index policies. Again, we consider $L = 100$ sample paths of the process $\{(X^n, Y^n, A^n), n \geq 0\}$ with $T = 5 * 10^5$ (10^6 total time slots and $5 * 10^5$ warmup slots) in each sample path. We report the results in Table 9.2 using the same format as that for Table 9.1.

λ^p	ξ for UIP	ξ for MWA	abs Imp	% Imp
0.1	14.26	19.06	4.8	25.18
0.25	5188.08	5282.37	94.29	1.78
0.5	5805.52	5897.56	92.04	1.56
0.75	17905.93	18010.79	104.86	0.58
1	337271.5	37393.96	122.46	0.33
1.1	46920.62	47048.08	127.46	0.27
1.25	62684.45	62812.31	127.86	0.20
1.5	84970.78	85090.1	119.32	0.14
1.6	103527.23	103654.22	126.99	0.12
1.7	109686.71	109812.72	126.01	0.11

Table 9.2: Performance of the index policies and the MWA in the dynamic cell: ξ for UIP and MWA, absolute improvement (abs Imp) and percentage improvement (% Imp) of UIP with respect to the MWA.

It is clear from the Table 9.2 that the best improvement for UIP, close to 25 %, corresponds to $\lambda^p = 0.1$. The results follow a similar trend as the static cell. As in the static cell, the index of the MWA and the UI are similar in the high traffic regime yielding similar results.

Chapter 10

Conclusions and Future Remarks

The literature on scheduling in wireless networks is vast and the algorithms used have been proven to be stable. However, none of the current algorithms considered explicitly attempt to optimize any system wide objective function such as throughput, queue lengths or even the penalty cost of starving users. Our contribution is to attempt to fill this void created by the lack of any systematic approach to "derive" implementable policies from a sound optimization based procedure.

We create a framework to derive well performing and easy to implement policies by modeling the system as an MDP. We consider two settings: the infinitely backlogged setting in part I where every user always has ample data to be served, and the externally arriving data setting in part II where the base stations maintains a data queue for each user. In the infinitely backlogged case, the optimal policy resulting from the MDP maximizes the long term net reward per time slot. The net reward in each time slot has two components, a positive component that is the reward earned from serving the user chosen for service in that slot, and a negative component that is the penalty accrued for all the customers not served in that slot. In the externally arriving data case, the optimal policy minimizes the total weighted cost of holding the data in queues. As is typical in most MDP's based on real life situations, the optimal policy is analytically intractable, and hence we derive heuristic policies. These policies are termed "index policies" because they involve computing an index for every user based only on the current state of that user. The index policy itself is to serve the user for which this index is maximized.

We compare the performance of our suggested index policies with existing policies using simulation and some analytical results when possible. In the infinitely backlogged setting we demonstrate that for any given level of quality of service (mean age, for instance) the throughput of our suggested policies is greater than that of the exisiting PFA. In fact, in the examples we consider, a throughput approaching the maximal throughput can also be realized using suitable parameters of the index policy while still maintaining reasonable quality of service levels. In the externally arriving data setting considered in part II the througput of every stable policiy is easily seen to be the same. In this part we demonstrate that our suggested policy is stable and performs better than the most widely used existing policy.

We derive all the index policies for a "static cell" (see section 1.5). However, we conduct simulation studies for a "dynamic cell" as well using straightforward extension of policies derived for the static cell. As expected, our suggested policies perform better than existing policies even for a dynamic cell.

We also consider some characteristics (monotonicity) of optimal policies in both part I and II. While we prove the monotonicity in age and queue length in parts I and II respectively, we could not prove monotonicity in rate (and the arrivals in part II). Proving monotonicity in rate, under some conditions on the structure of the underlying Markov chains if needed, is one direction in which this work can be extended.

We consider only the base station - user interaction in this thesis. However, cellular networks have many more components as described in section 1. Modeling and performance analysis of a network with more than one base stations is another area that can be considered for future work.

Finally, other wireless networks that involve mobility of information sources or data access points can also be modeled along similar lines, i.e., using Markov chains to model their movements. We have not considered this area so far, but given the fast proliferation of wireless technologies and users demanding increasing mobility and flexibility, this could be an interesting and worthwhile direction of future research.

Bibliography

[1] Informa telecoms & medias world cellular data metrics report. [Online]. Available: http://www.intomobile.com/2009/05/20/informa-mobile-data-revenues-rise-24-in-2008.html

[2] L. O. Walters and P. S. Kritzinger, “Cellular networks: past, present and future,” Crossroads, vol. 7, pp. 4 – ff35, Dec. 2000.

[3] A. S. Tanenbaum, Computer Networks. Prentice Hall, 1996.

[4] Gsm data knowledge site. [Online]. Available: http://www.mobiledata.com

[5] Mobile world celebrates four billion connections. [Online]. Available: http://www.gsmworld.com/newsroom/press-releases/2009/2521.htm

[6] A. A. M. Zeng and V. K. Bhargava, “Recent advances in cellular wireless communications,” IEEE Communications Magazine, vol. 37, pp. 128–138, Sep. 1999.

[7] T. S. Rappaport, Wireless Communications. Prentice Hall, 1996.

[8] 3g - cdma2000 1xev-do technologies. [Online]. Available: http://www.cdg.org/technology/3g_1xEV-DO.asp

[9] D. Tse, “Multiuser diversity in wireless networks.” [Online]. Available: http://www.eecs.berkeley.edu/~dtse/stanford416.ps

[10] A. Jalali, R. Padovani, and R. Pankaj, “Data throughput of cdma-hdr: a high efficiency-high data rate personal communication wireless system,” in Proc. (IEEE) Vehicular Technology Conference, Tokyo, Japan, May 2000, pp. 1854 – 1858.

[11] M. Andrews, “Instability of the proportional fair scheduling algorithm for hdr,” IEEE Transactions on Wireless Communications, vol. 3, p. 2004, 2002.

[12] B. Awerbuch and T. Leighton, “A simple local-control approximation algorithm for multicommodity flow,” in Proceedings of the 34th Annual Symposium on Foundations of Computer Science, 1993, pp. 459 – 468.

[13] ——, “Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks,” in Proceedings of the 26th Annual ACM Symposium on Theory of Computing, 1994, pp. 487 – 496.

[14] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, vol. 37, pp. 1936–1948, Dec. 1992.

[15] ——, “Dynamic server allocation to parallel queues with randomly varying connectivity,” IEEE Transactions on Information Theory, vol. 30, pp. 466–478, 1993.

[16] M. J. Neely, E. Modiano, and C. E. Rohrs, “Power and server allocation in a multi-beam satellite with time varying channels,” in Proceedings of IEEE INFOCOM ’02, 2002, pp. 1451–1460.

[17] M. Andrews, K. Jung, and A. Stoylar, "Stability of the max-weight routing and scheduling protocol in dynamic networks and at critical loads abstract," pp. 145–154, 2007.

[18] N. W. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in Proceedings of IEEE INFOCOM '02, 2002, pp. 1451–1460.

[19] S. Muthukrishnan and R. Rajaraman, "An adversarial model for distributed dynamic load balancing," in Proceedings of the 10th ACM Symposium on Parallel Algorithms and Architectures, 1998, pp. 47–54.

[20] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control," in Proceedings of IEEE INFOCOM '05, 2005, pp. 1794–1803.

[21] M. Andrews and L. Zhang, "Scheduling over nonstationary wireless channels with finite rate sets," IEEE/ACM Transactions on Networking, vol. 14, pp. 1067–1077, 2006.

[22] ——, "Scheduling over a time-varying user-dependent channel with applications to high speed wireless data," in Proceedings of the 43rd Annual Symposium on Foundations of Computer Science, 2002, pp. 293–302.

[23] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," Probability in the Engineering and Informational Sciences, vol. 18, pp. 191–217, 2004.

[24] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," Analytic Methods in Applied Probability. In Memory of Fridrih Karpelevich. Yu. M. Suhov, Editor, vol. 207, pp. 185–202, 2002.

[25] S. Shakkottai, A. L. Stolyar, and R. Srikant, "Pathwise optimality of the exponential scheduling rule for wireless channels," Advances in Applied Probability, vol. 36, pp. 1021–1045, 2004.

[26] A. Parekh, "A generalized processor sharing approach to flow control," Ph.D. dissertation, MIT, 1992. [Online]. Available: http://www.tecknowbasic.com/thesis.pdf

[27] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," in Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, Palo Alto, CA, USA, Aug. 1996, pp. 143–156.

[28] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in Symposium proceedings on Communications architectures and protocols, Austin, TX, USA, Sep. 1989, pp. 1–12.

[29] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," IEEE/ACM Transactions on Networking, vol. 6, p. 611624, 1998.

[30] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," IEEE/ACM Transactions on Networking, vol. 5, pp. 690–704, 1997.

[31] P. McKenney, "Stochastic fairness queueing," in Proceedings of IEEE INFOCOM, San Francisco, CA, USA, Jun. 1990, p. 733740.

[32] M. Andrews, “A survey of scheduling theory in wireless data networks,” IMA Volumes in Mathematics and its applications, vol. 143, pp. 1–18, 1999.

[33] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, “A bandwidth efficient high speed data service for nomadic users,” IEEE Communications Magazine, vol. 38, pp. 70–77, Jul. 2000.

[34] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” Foundations and Trends in Networking, vol. 1, no. 1, pp. 1–144, 2006.

[35] X. Liu, E. Chong, and N. Shroff, “Opportunistic transmission scheduling with resource-sharing constraints in wireless networks,” IEEE J. Sel. Areas Commun., vol. 19, pp. 2053–2064, Oct. 2001.

[36] M. Opp, K. Glazebrook, and V. Kulkarni, “Outsourcing warranty repairs: Dynamic allocation,” Naval Research Logistics Quarterly, vol. 52, pp. 381–398, Dec. 2005.

[37] Q. Liu, S. Zhou, and G. B. Giannakis, “Queuing with adaptive modulation and coding over wireless links: Cross-layer analysis and design,” IEEE Trans. Wireless Commun., vol. 4, pp. 1142–1153, May 2005.

[38] M. Puterman, Markov Decision Processes - Discrete Stochastic Dynamic Programming. New York, USA: John Wiley & Sons, Inc, 1994.

[39] I. Kadi, N. Pekergin, and J. M. Vincent, Analytical and Stochastic Modeling Techniques and Applications. Springer US, 2009, ch. 11.

[40] Q. Hu and W. Yue, Markov Decision Processes with Their Applications. Springer US, 2008.

[41] S. M. Ross, Introduction to Stochastic Dynamic Programming. Academic Press, Inc., 1983.

[42] J. C. Gittins, “Bandit processes and dynamic allocation indices (with discussion),” Journal of the Royal Statistical Society, vol. 41, pp. 148–177, 1979.

[43] ——, Multi-Armed Bandit Allocation Indices. John Wiley, 1989.

[44] R. Weber, “On the gittins index for multiarmed bandits,” The Annals of Applied Probability, vol. 2, pp. 1024–1033, Nov. 1992.

[45] K. Glazebrook, J. Nino-Mora, and P. Ansell, “Index policies for a class of discounted restless bandits,” Advances in Applied Probability, vol. 34, pp. 754–774, Dec. 2002.

[46] V. Kulkarni, Modeling and Analysis of Stochastic Systems. New York, USA: Chapman & Hall, Inc, 1995.

[47] R. Burden and D. Faires, Numerical Analysis. Brooks Cole, 2004.

[48] A. Eryilmaz, “Efficient and fair scheduling for wireless networks,” Ph.D. dissertation, Univ. of Illinois at Urbana-Champaign, 2005. [Online]. Available: http://www.ece.osu.edu/~eryilmaz/AtillaEryilmazResearch.html

[49] J. Norris, Markov Chains. Cambridge University Press, 1997.

[50] D. Gross and C. Harris, Fundamentals of Queueing Theory. John Wiley & Sons, Inc, 1985.

[51] F. Chen and V. Kulkarni, Stochastic Processes, Optimization, and Control Theory: Applications in Financial Engineering, Queueing Networks, and Manufacturing Systems. Springer US, 2006, ch. 5.

[52] T. Lindvall, Lectures on the coupling method. John Wiley & Sons, 1992.

[53] C.-H. Wu, M. E. Lewis, and M. Veatch, "Dynamic allocation of reconfigurable resources ina two-stage tandem queueing system with reliability considerations," IEEE Transactions on Automatic Control, vol. 51, pp. 309 – 314, Feb. 2006.

[54] M. F. Neuts, Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach. The Johns Hopkins University Press, 1981.

[55] N. Bolia and V. Kulkarni, "Index policies for resource allocation in wireless networks," IEEE Transactions on Vehicular Technology, vol. 58, pp. 1823–1835, 2009.

[56] S. Asmussen, Applied Probability and Queues. John Wiley & Sons, Inc, 1987.

[57] E. Leonardi, M. Mellia, F. Neri, and M. A. Marson, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in Proceedings of IEEE INFOCOM, vol. 2, 2001.

[58] P.R.Kumar and S.P.Meyn, "stability of queueing networks and scheduling policies," IEEE Transactions on Automatic Control, vol. 40, pp. 251–260, 1995.