MTech Dissertation

Backlog Optimal Downlink Scheduling in Energy Harvesting Base Station in a Cellular Network

Submitted in partial fulfillment of the
requirements for the degree of

Master of Technology
in Communication Engineering

by

Venkhat V
(Roll No. 123079008)

Under the guidance of

Prof. Abhay Karandikar, Prof. Prasanna Chaporkar

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

June, 2015

To my mother

Dissertation Approval

This dissertation entitled **Backlog Optimal Downlink Scheduling in Energy Harvesting Base Station in a Cellular Network** by Venkhat V (Roll No. 123079008) is approved for the degree of **Master of Technology** in **Communication Engineering**.

Examiners

____

____

____

Supervisors

____

____

Chairperson

____

Date: 09/06/15

Place: MUMBAI

Declaration

I declare that this written submission represents my ideas in my own words and wherever others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/ fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by IIT Bombay and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Venkhat V

Date: June 9, 2015

Place: Indian Institute of Technology Bombay, Mumbai

Acknowledgements

I sincerely thank my supervisor Prof. Abhay Karandikar for his support and guidance throughout the last three years. It was his choice to select me as one of his students, that gave me the opportunity to join, learn and evolve as a person here. Without him, this M.Tech would not have been possible and I'm truly grateful for that. I gratefully thank Prof. Prasanna Chaporkar for letting me work with him for the past two years. His thoughts and motivation to solve a problem, gave me the inspiration to try to understand any problem thoroughly from the scratch. I truly thank him for answering my questions, listening to my thoughts and helping me in these two years.

I would like to thank my friend Karunakaran Kumar and Vatsal Shah for the numerous discussions we had which helped me understand, solve the problems and conceptual doubts in my research work. Especially Kumar for listening to my blabbering, even when he was not interested sometimes. I happily thank my friend Arghyadip Roy for his suggestions, discussions and numerous chit chats especially during WiOpt. I would like to thank the Information Networks lab members for attending my talks and presentations.

I would like to thank India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services (IU-ATC) and Department of Science and Technology (DST), Government of India for supporting and funding my M.Tech studies as well as this research work.

I am eternally grateful to my mother for her support, guidance and will to bring us up through the hard times. She has been and continues to be the pillar of support for me. I also thank my brother for pushing me when necessary and sister for her opinions. Finally, I thank all my friends, who made my stay at IITB a wonderful experience in my life.

June 9, 2015 **Venkhat V**

Abstract

Future communication devices are aiming at becoming self sustainable with the use of green energy sources. A transmitter powered by renewable energy source becomes self sustainable, can lead a longer life and reduces cost of maintenance. In this work, we consider the broadcast channel with a transmitter and N receivers. The transmitter is powered by a renewable energy source and has finite battery capacity. The channel gains between user and transmitter are fixed over time slots. The transmitter requires power P_i to transmit a packet to i^{th} user. In this setting, our first objective is to look for a policy that minimizes backlog i.e. the total number of packets in the transmitter in every slot, while accounting for randomness in the packet arrival and energy recharge processes. We then show that there does not exist a policy which minimizes backlog in every slot. Minimizing in every slot is a stricter sense of optimization than expected sense. As there is no policy that minimizes backlog in every slot, we look at minimizing the expected backlog. Hence, our next objective is to minimize the expected backlog at the transmitter. We formulate the problem as an infinite horizon discounted Markov Decision Process (MDP) problem and obtain the structural properties of an optimal policy. One important structural property is that above some energy threshold, it is always optimal to transmit a packet rather than staying idle. These structural properties provide valuable insights for designing close to optimal policies that are computationally efficient for real life implementations. In special cases, we provide complete description of an optimal policy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At present, the world is in power crisis as most conventional energy resources are draining rapidly. Future communication devices are aiming at becoming self sustainable with the use of green energy sources. There are various kinds of green energy sources such as solar, wind, bionic, Radio Frequency (RF) energy, pressure etc. as shown in Figure 1.1.

1.1 Motivation for Energy Harvesting Communication Networks

A transmitter powered by renewable energy source can lead to a longer life and reduces the cost of maintenance. Devices powered by green energy sources can be placed in remote locations, where replacing batteries would be very costly and difficult to maintain. Energy harvesting devices offer other advantages such as energy self sufficiency, self sustainability with lifetimes limited only by the lifetime of their hardware. Circuits and devices side of engineering has been developing energy harvesting devices for decades [4–7]. However, on the communications and networking side, the focus has been on energy-aware communication system design, in the form of optimum average-power constrained communications, and energy-efficient networking [8–10]. Only recently, communications subject to explicit energy harvesting conditions has garnered attention. The main reason is that, the power generated via energy harvesting in these existing devices is in the order of few hundred milliwatts. These are enough for operating low powered sensors, body activity sensing devices etc. However, for communication point of view, we need much more power. Advantages of energy harvesting has been known for a long

Figure 1.1: Green Energy Sources (Source: wonderfulengineering.com)

time and only recently, researchers have started looking into such scenarios. One reason is that, now technologies are growing and there is a belief that better energy harvesting techniques would arise. But the other main reason is the industrial and economic interest in these devices. This is due to the so called, *Internet of Things* (IoT) as shown in Figure 1.2. The vision pertaining to Interent of Things are being enabled by the developing Fifth Generation (5G) wireless technology. The major techniques involved in 5G devices are mm wave, massive MIMO, smart antennas, dense networks and the important one which is of our concern is the energy harvesting communication devices. This is the major reason, why recently research is booming in energy harvesting communication devices.

However, for self sustainability one has to effectively deal with the randomness in energy harvesting process. This introduces new challenges e.g. the existing optimal scheduling algorithms with respect to the conventional sources may not be optimal in energy harvesting networks. So, it is necessary to investigate and obtain optimal scheduling algorithms for energy harvesting scenarios.

Figure 1.2: Internet of Things (Source: Qualcomm)

1.2 Challenges in Energy Harvesting Communication Devices

The major issue in energy harvesting communication is the randomness in both time and value of the energy recharge arrival process. Recharge process is a location and time dependent non-stationary stochastic process. Thus in addition to considering random channel variations, one needs to account for the battery recharge process and energy consumption models in decision process. Another issue is that the recharge process may depend on the current battery level. Specifically, when battery charge is low, the rate of recharge is higher. In this setting, we face the following trade-off, we want to operate battery at low energy level so as to extract more out of the renewable energy source, but at the same time we want to save stored energy in the battery so as to exploit good channel conditions. It is necessary to resolve this trade-off optimally. The next major challenge is the finite capacity of the battery. This leads to following trade-off: we want to conserve the battery to exploit good channel conditions, but if the battery is close to being fully charged then we want to transmit even in bad channel conditions so that the recharge from the renewable energy source is not wasted. Also knowing the recharge process or even statistics or distribution of the process a priori is difficult. This motivates the design of online and learning algorithms which will estimate the information about the recharge process or operate efficiently with no knowledge of energy recharge process. These issues affect the objectives and the constraints involved in designing optimal scheduling or resource allocation techniques in energy harvesting communication devices.

The basics of scheduling, different metrics involved, different objectives and possible constraints are discussed in the next section.

1.3 Basics of Scheduling/Resource Allocation

First, let us look at what cross layer scheduler design means.

1.3.1 Cross Layer Design

Traditional wired communication models follow Open Systems Interconnection (OSI) architecture. OSI model is layered architecture, where the communication networking tasks are divided into hierarchy of services in the respective layers. Non-adjacent layers are not allowed to communicate with each other, while adjacent layers communicate by means of protocols. Cross layer design involves sharing parameters between adjacent layers and design of protocols/algorithms which operate jointly on two or more layers. In wireless networks, cross layer design has resulted in great performance improvements. This improvement is significant in the Media Access Control (MAC) layer by utilizing the channel related information from the physical layer. These algorithms are termed as *Cross layer scheduling algorithms*. In recent research works, it is always assumed that the Channel State Information (CSI) is available at the transmitter. In this thesis as well, it is assumed that the channel state information is known at the transmitter.

1.3.2 Function of a Scheduling Algorithm

A main function in every scheduling algorithm is to choose the power and rate at which transmitter has to transmit data. Actually, scheduling algorithm first looks at the channel information from the physical layer, queue length, packet length etc. Next based on the channel state, it adapts/modifies the power and rate at which it transmits to achieve optimal performance based on some objective, for example minimization of average power consumption. If it is a multiuser setting with a single resource/channel, in addition to adapting powera and rate, it has to choose the user to which it has to transmit. Whereas when there are multiple resources, it has to choose a combination of users among which the resources will be distributed along with power and rate adaptation.

1.3.3 Scheduling Types- Centralized/Distributed

In a multiuser setting, the scheduling algorithms can be classified into two types.

1. Centralized scheduling,
2. Distributed scheduling.

In centralized scheduling, a single entity e.g. base station knows the information about all the users and decides what is to be done. For example, on the uplink, if the base station knows the user's information, decides the action and informs the users about the action to be done, then it is a centralized scheduling algoirthm.

In distributed scheduling, the users will individually take their own decisions. Generally, in ad hoc networks, each user knows only its own information and decides on its own.

In the next section, the general objectives in scheduling algorithms and the constraints which have to be satisfied are explained.

1.4 Objectives of a Scheduling Algorithm

Generally, scheduling algorithm design is framed as an optimization problem with some objectives and constraints. Some of the objective functions which are generally used in literature are mentioned in the following subsection.

1.4.1 Objective Function

Researchers are interested mainly in optimizing three objectives.

1. Throughput,
2. Delay,
3. Power.

There are different forms of throughput that are maximized. Some examples are long term average throughput, expected throughput, short term throughput, system throughput in multiuser setting etc. Different forms of delay which are minimized include average delay, expected delay etc. By Little's law, it is known that minimizing delay is equivalent to minimizing queue length. So the problem of minimizing delay is framed in terms of minimization of queue length. While for optimizing power, generally we try to minimize average power consumption over an infinite time.

Now let us see different Quality of Service (QoS) constraints that have to be satisfied by the scheduling algorithms.

1.4.2 QoS Constraints

There are different notions of Quality of Service (QoS) metrics in a wireless network. Satisfying these QoS metrics ensures that the users are served properly. For example, in a live online game setting, there is a need for delay to be very less, say less than 150 ms. Thus this is the maximum delay allowed in this network. Some of these QoS metrics are

1. Maximum Delay,
2. Minimum Rate ,
3. Fairness
   - Minimum allocation,
   - Fair relative throughput/Time slot allocation,
   - Proportional fair allocation.

Next, the different trade-offs which arise while trying to design scheduling algorithm are discussed.

1.4.3 Trade-Offs

There are three main trade-offs which will affect the scheduling algorithms.

Delay - Power Trade-off: Less the desired power, more the queuing delays. For Example, let us consider a single user point to point scenario. The power required to transmit reliably at a particular rate is a convex function of the rate as shown in Fig 1.3. If R bits/sec have to be transmitted, in error free communication, power required is

$$P(x, R) = \frac{WN_0}{x}(2^{\frac{R}{W}} - 1). \tag{1.1}$$

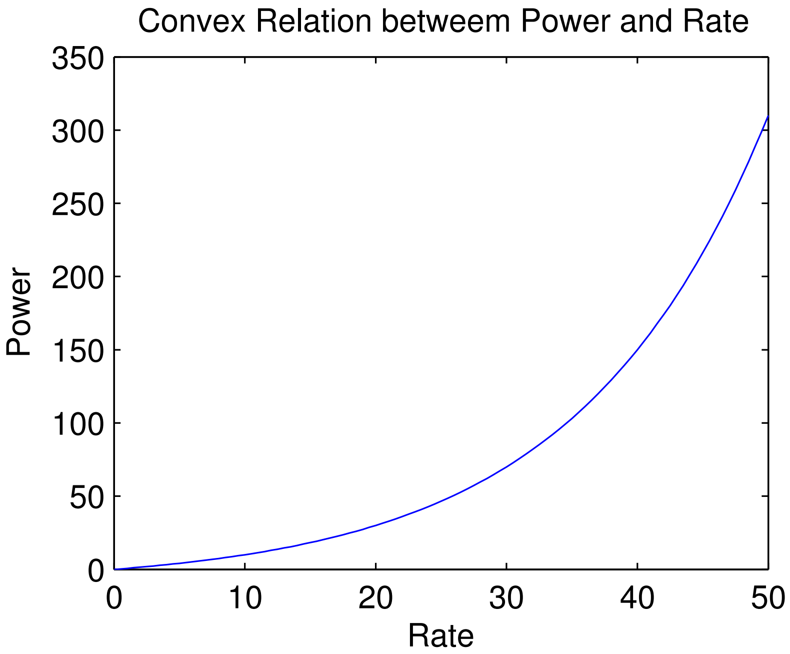


Figure 1.3: Convex Power Rate Relation

new Thus, transmission at lower rates can result in power savings, i.e., the scheduler should transmit data in opportunistic chunks. Moreover, data should be transmitted at an opportunistic time, i.e., when the channel condition is good. Both these power saving considerations result in higher queuing delays at the transmitter.

Throughput - Delay Trade-off: Algorithms that are throughput optimal, do not necessarily ensure small average queue lengths and hence small delays. For example, let us consider queue stability as a metric of QoS while maximizing throughput. In this scenario, not every policy achieving maximum throughput guarantees queue stability.

Throughput - Fairness Trade-off: Exploiting multiuser diversity in an opportunistic manner by scheduling the user with the best channel gain might introduce unfairness at the higher layers. Users who are closer to the base station might experience perennially better channel conditions and thereby obtain a higher share of the system resources at the expense of users who are farther away from the base station. On the other hand, scheduling users with poor channel gains results in a reduction in the overall achievable sum throughput. Thus, there exists a fairness-sum throughput tradeoff.

For further information about the basics of scheduling and channel models, readers are referred look at Nitin Salodkar's thesis [11].

1.5 Outline of the Thesis

In this chapter, we have looked at the motivation behind the research on energy harvesting communication devices. The challenges due to energy harvesting and some of the basics of scheduling have been discussed. The remaining of the thesis is organized into four chapters. Chapter 2 gives a review of the available literature in delay minimization and energy harvesting. Chapter 3 gives the system model, for which we are designing an optimal scheduling algorithm that minimizes backlog. We also state results regarding the optimal policy. This chapter contains the main contribution of this thesis. We conclude the thesis in Chapter 4. We mention the limitations of our work and also express what could be improved or extended in future.

For this thesis to be self contained, the basics of optimization framework, Markov Decision Process have been reviewed in Appendix A.

Chapter 2

An Overview of Research Issues in Delay Optimal and Energy Harvesting Scheduling

In this chapter, first the review of the literature in delay minimization is discussed. Then the literature in energy harvesting is classified into two categories. The research works in each category is analyzed separately and some prominent works are analyzed in detail.

2.1 Delay Minimization

In [1], one of the prominent works by Tassiulas and Ephremides, the authors have described the policy which achieves delay optimality everywhere for tandem queuing and parallel queuing systems with adjacency constraints on servers. The system model of this paper is shown in Figure 2.1.

In the parallel queuing system, the policy which schedules such that it serves most number of queues, achieves backlog optimality everywhere. Here backlog represents total number of packets in the system. Whereas in a tandem queuing system with a single destination, the following policy is shown to be delay optimal everywhere: Select a non-empty queue (say i) that is closest to the destination, then choose a non-empty queue that is closest to the queue i and does not interfere with i's transmission and repeat this until no further queue can be selected. The policy schedules a packet from all the chosen queues simultaneously in a slot.

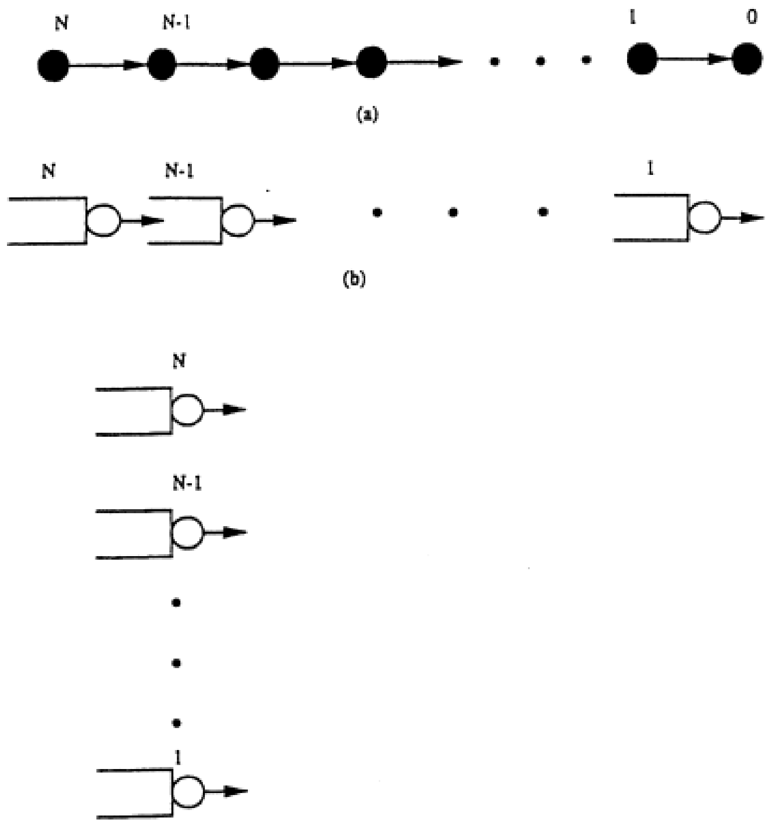


Figure 2.1: a) Packet Radio Network b)Tandem Queue c)Parallel Queue (Source [1])

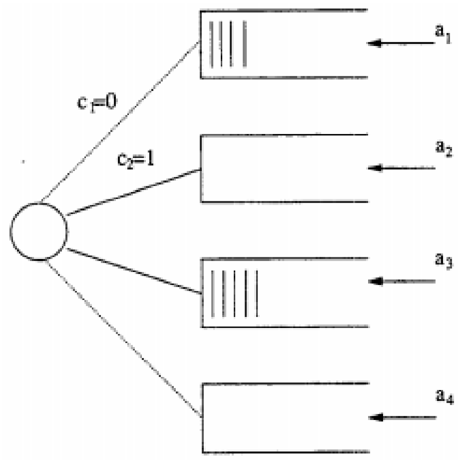


Figure 2.2: N parallel queues with time varying connectivity (Source [2])

Optimal policy for throughput maximization and delay minimization in every slot in a parallel queuing system with time varying on-off connectivity is obtained in [2]. The system model is shown in Figure 2.2.

In this multi user downlink with random connectivity, when all user's arrival and channel connectivity processes are identical, the authors show that the longest connected queue (LCQ) policy minimizes backlog in the system at every time slot. When queues are of infinite size, the authors have obtained the necessary and sufficient conditions in terms of the system parameters for stabilizing the system. Also, if the system can be stabilized, they state that the policy that serves the longest connected queue stabilizes the system. From these papers, it is observed that for some systems, there exists a policy which has optimality at every time slot.

Trade-offs between average power and average delay has been analyzed for a fading wireless channel in [9]. Generally, researchers have been more interested in minimizing average power rather than delay. One reason is that, in most applications, delay within an upper bound, is sufficient as a QoS guarantee. us review some of the prominent works, that minimize power consumption. Under average delay constraint, average power is minimized for a single user fading channel and online implementation using stochastic approximation is obtained in [12]. In [10], the authors consider the problem of minimizing average power and peak transmit power, with a constraint of average delay in system with time varying channel. This problem is solved using dynamic programming formulation coupled with a duality argument and find some interesting features regarding the optimal policies. Even though, vast amount of research work is available in minimizing energy, yet there have been some papers, which minimize different forms of delay. These are discussed below.

Average packet transmission delay is minimized for a single user and multiuser uplink fading channel respectively controlling the power and rate dynamically under conventional energy setup in [13, 14]. Average waiting time of a head of line packet is minimized using dynamic programming in loss tolerant MAC layer multicast in [15].Order optimal delay result in a one hop wireless network with N users and ON-OFF channels is shown in [16]. These works consider some variants of delay optimization in wireless networks. However, these do not consider energy harvesting scenario.

2.2 Energy Harvesting Communications: Classifications

Data communication in energy harvesting systems has been explored in different scenarios, e.g. see [3,17–22]. The literature in energy harvesting communication can be broadly classified into three categories,

1. Offline scheduling strategies - realization of energy recharge process is known, [3, 17, 23, 24],

2. Online scheduling strategies - energy recharge process is random and not known ahead of time, [18, 22, 25–28, 28],

3. Information theory point of view - capacity and theoretical limits, [19–21, 29].

2.2.1 Offline Scheduling Strategies

Optimal offline policy for minimization of transmission completion time in a Gaussian broadcast channel with finite battery setup is computed assuming the knowledge of future energy values in [3]. The system model is shown in Figure 2.3. The optimal power transmit sequence is obtained by directional water filling algorithm. They show that there exist $M - 1$ cut off power levels, so that a user i is allocated the power between $i - 1^{st}$ and i^{th} cut off level.

Optimal online and offline policies for maximizing throughput and minimizing transmission time is obtained for a wireless fading channel in [17]. Transmission completion time under a deterministic setting, i.e. when the arriving energy values are known apriori, is minimized in [23]. The problem of throughput maximization in a point to point link is framed as a Markov Decision Process (MDP) problem and monotone property of an optimal policy is obtained in [24].

2.2.2 Online Scheduling Strategies

In a multiple access communication system,the long term sum throughput is maximized using techniques from calculus of variations in [18]. Here the battery is modeled using the storage dam model and the upper bound on throughput is obtained as a function of

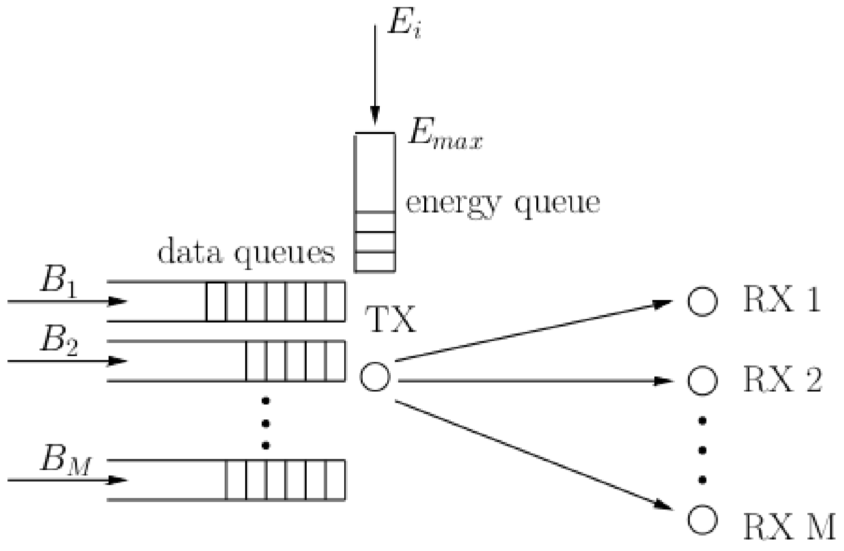


Figure 2.3: M User Broadcast Channel (Source [3])

battery capacity. Optimal policy for maximizing total amount of data transmitted in a given finite duration is obtained in [22]. Also, the related problem, minimization of the transmission completion time is analyzed and optimal policy has been stated. Recent works deal with maximizing throughput in different scenarios with a energy harvesting setup [25–27]. In renewable energy paradigm, an online algorithm for minimizing delay is proposed and its competitive ratio is analyzed for an arbitrary wireless channel in finite time for a Gaussian single user channel and multiuser Gaussian channels in [30] and [28] respectively. Note that most of this work aims at throughput maximization and do not consider delay.

2.2.3 Information Theory View

There have been papers that look into information theoretic capacity of channels in energy harvesting scenarios e.g. see [19–21], but this body of work does not consider delay minimization. In [19], the capacity in Additive White Gaussian Noise (AWGN) channel with random energy arrivals is obtained. They show that the capacity of the AWGN channel in energy harvesting setup is equal to the capacity in a conventional AWGN channel with an average power constraint equal to the mean of energy arrivals. They provide two different capacity achieving schemes. Whereas in [29], the two-user additive Gaussian multiple access channel is considered. They consider the static amplitude constrained Gaussian multiple access channel and prove that the boundary of the capacity region is achieved by discrete input distributions of finite support. When both of the transmitters are equipped with no battery, Shannon strategies applied by users provide an inner bound for the capacity region.

2.3 Motivation for the Thesis

In literature, most problems that are solved address the objective of minimizing energy. Even the papers which have addressed delay, do not consider energy harvesting scenario. In energy harvesting system, point to point link with different channel conditions and in both online, offline energy setting, optimal scheduling strategies have been computed already. There are open research problems in a multiuser setting, i.e downlink with N users, and when energy arrivals are random and are not known ahead of time. In this work, we investigate the problem of minimizing the backlog at the renewable energy empowered transmitter in a broadcast channel. The transmitter is assumed to have a finite battery that is recharged as per some stationary stochastic process. We formulate the problem of obtaining optimal policy for two different objectives. In one case, we look for minimizing backlog in every slot and in second, we minimize expected backlog. We investigate the structure of these policies, with regard to how it tackles the finite size battery and the randomness in energy arrival.

Chapter 3

Minimizing Backlog for Downlink in Energy Harvesting Base Station

We investigate the broadcast channel with multiple receivers and a single transmitter having a finite capacity battery that is powered by a green energy source. Our aim is to minimize the backlog at the transmitter. Note that minimizing backlog is equivalent to minimizing queuing delay.

3.1 System Model

We consider a simple system model with one base station and N users and a battery which is recharged by a renewable energy source as shown in Figure 3.1

Thus, there is one server and N parallel queues. The time is divided into intervals of fixed length τ called time slots. Server can serve one packet in a slot. This is similar to downlink network with N separate queues for N users and a single controller in the base station which decides which user is to be scheduled. We only assume slow fading, i.e., the channel gains do not vary over time. Let the channel gains be $h_1, \ldots, h_N$. Packets are of the constant length l. Without loss of generality, $|h_i| > |h_{i+1}|, \forall i \in \{1, \ldots, N-1\}$.The transmission rate is given by the Shannon's capacity formula $B \log_2 \left(1 + \frac{P|h|^2}{BN_0}\right)$, where N_0 is Noise power spectral density. Without loss of generality, assume bandwidth B to be 1. The power required to transmit a packet from the base station to a user within a slot is given by $P_i = \frac{N_0}{|h_i|^2} \left(2^{\frac{l}{\tau}} - 1\right)$ for $i = 1, \ldots, N$. So, $P_i < P_{i+1}$ $\forall i$. This is shown in Figure 3.2, where as the channel degrades from user 1 to user N, power required increases

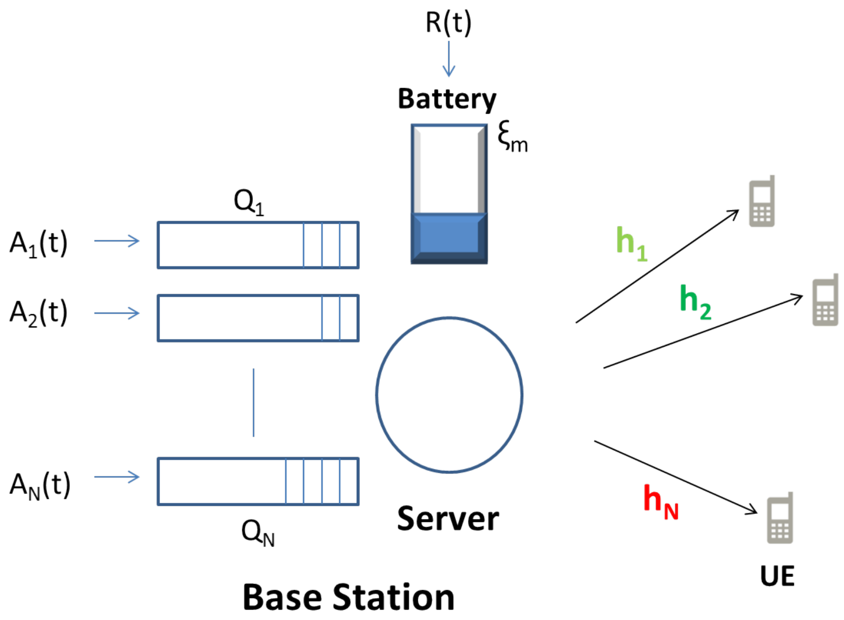


Figure 3.1: System Model

and is shown via colour change from green to red respectively.

Let $A_i(t)$ be the number of packets arriving in queue i at the beginning of slot t, for $t \geq 1$. Let $R(t)$ be the recharge energy arrivals which are added to the energy buffer/battery at the beginning of slot t, for $t \geq 1$. Energy arrival values are not known ahead of time. The packet arrival processes and the energy recharge process are random processes. The system model with the arrivals is shown in Fig. 3.1. Action or decision is taken after the arrivals. Queues are considered to be of infinite capacity and thus packet loss never happens. Battery is of finite capacity with ξ_m being the maximum value. Let $Q_i(t)$ indicate the number of packets in the queue i at the beginning of slot t. Let $E(t)$ indicate the amount of energy in the battery at the beginning of slot t. A queue i is said to be *connected*, if $Q_i(t) > 0$ and $E(t) \geq P_i$. Thus in any slot t, a packet can be transmitted only from the set of connected queues, not otherwise. Figure 3.3 gives an example when queues 1,2,4. are connected. This can be considered as a time varying connectivity, but depending on the power level in the battery. We define a few terms which will be used in this thesis hereafter.

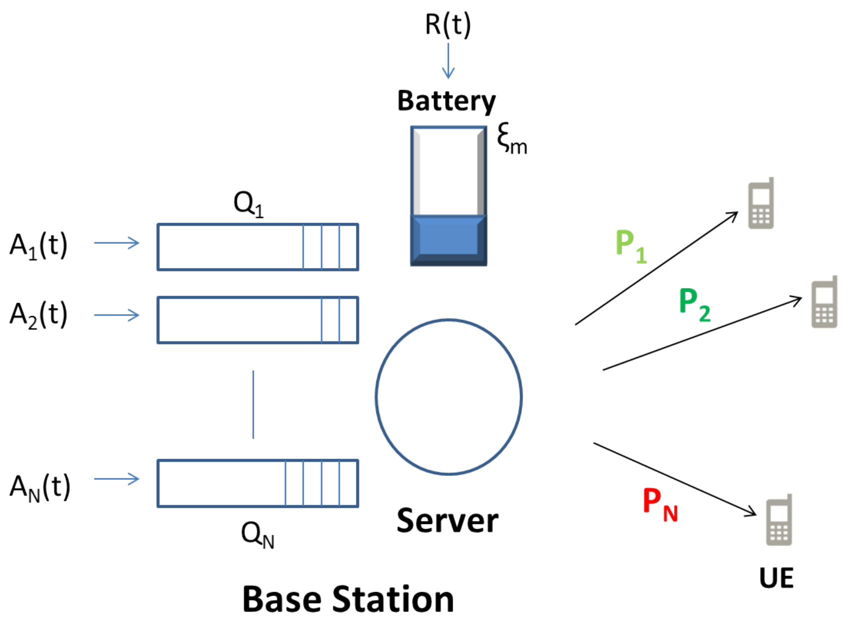


Figure 3.2: Power Requirement

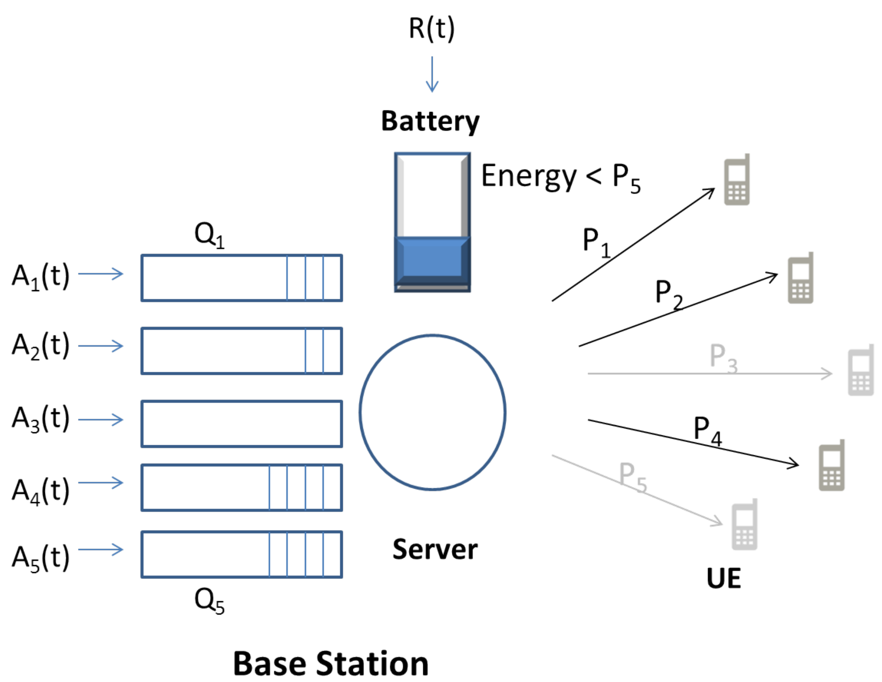


Figure 3.3: Connected Queue

3.1.1 Definitions

Definition 1 (Scheduling Policy) *Scheduling policy is a sequence of decision rules at each slot t, that chooses a connected queue from which a packet will be transmitted in the slot or decides to stay idle.*

We assume scheduling policy to be causal i.e., the action taken is a function of the past actions, energy arrivals and packet arrivals.

Definition 2 (Stationary Policy) *Stationary policy is a map $\pi : \mathcal{S} \rightarrow \{0, 1, \ldots, N\}$, i.e. the policy maps the system state s to an action $\{0\} \cup \mathcal{C}_s$, where $\mathcal{C}_s$ is the set of connected queues in state s.*

The stationary policy does not depend on time. Moreover, given the current state, the decision solely depends on the state and not on the past.

Definition 3 (Non-idling Policy) *A policy π is non-idling,*

1. *if it is stationary,*
2. *if $\mathcal{C}_s \neq \emptyset$, then the policy schedules a queue from $\mathcal{C}_s$ for every s.*

Essentially what this definition states is that, a non-idling policy cannot stay idle, when there is a connected queue. Any stationary policy which is not non-idling is referred to as idling policy.

The queue length and the battery energy level depend on the scheduling policy. This dependence is made clear by mentioning the policy in the superscript. Let us define an indicator variable, $I_i^{\pi}(t)$ which is 1, if a packet is scheduled from queue i by policy π in slot t and 0 otherwise. Let $P^{\pi}(t)$ denote power spent in slot t and ξ_m denote battery capacity. $(\boldsymbol{Q(0)}, E(0))$ denotes the initial values in slot 0. Queue length under policy π evolves as follows. For every $t \geq 1$ and $i \in \{1, \ldots, N\}$,

$$Q_i^{\pi}(t) = Q_i^{\pi}(t-1) + A_i(t) - I_i^{\pi}(t-1). \tag{3.1}$$

The queue state at t is $\boldsymbol{Q^{\pi}(t)} = [Q_1^{\pi}(t), \ldots, Q_N^{\pi}(t)]^T$, where $\boldsymbol{x}^T$ denotes the transpose of vector $\boldsymbol{x}$. Thus in vector notation queue state evolves as follows:

$$\boldsymbol{Q^{\pi}(t) = Q^{\pi}(t-1) + A(t) - I^{\pi}(t-1)}.$$

The battery energy level under policy π in slot t is given as follows,

$$E^{\pi}(t) = \min\left\{E^{\pi}(t-1) + R(t) - P^{\pi}(t-1), \xi_m\right\}. \quad (3.2)$$

Recall that policy π can schedule only from a connected queue, which implies $E^{\pi}(t-1)+R(t) \geq P^{\pi}(t-1)$ always.

In the next section, we discuss the challenges involved in designing delay optimal policy for our system model.

3.2 Challenges in Designing Optimal Policy

Let us first define the notion of delay optimality akin to the one considered in [2].

Definition 4 (Backlog optimality everywhere) *Scheduling policy π is backlog optimal everywhere if it satisfies $\sum_{i=1}^{N} Q_i^{\pi}(t) \leq \sum_{i=1}^{N} Q_i^{\pi'}(t) \quad \forall \pi', t = \{0, 1, 2, \ldots\}$, under any packet and recharge energy arrivals and any initial state $(\boldsymbol{Q(0)}, E(0))$.*

Note that the backlog optimality everywhere is the sample path wise optimality. So we call the policy π that achieves backlog optimality everywhere as a pathwise optimal policy. The reason why pathwise optimality is important is that, it is a more stricter sense of optimality than expected optimality. If pathwise optimality exists, then there is no need to look for expected optimization. For simpler systems, pathwise optimality exist as in [1,2]. When the systems become more complex, achieving pathwise optimality may not be possible. In our system model, we ask the question whether a policy exists that achieves pathwise optimality and if exists, then what is that optimal policy.

In our model, if all queues remain connected in all slots, then any non-idling policy is backlog optimal everywhere. The queues remain connected if for example, energy arrival $R(t) > P_N, \ \forall\ t$. Next we address the existence of backlog optimality everywhere when *queues do not remain connected in all the slots* . Specifically, we show that backlog optimality everywhere does not exist in this scenario. We state this formally in the following theorem.

Theorem 1 *There does not exist a policy π that achieves backlog optimality everywhere.*

This theorem is proved in the following four subparts. In the first part, we show that if there exists an optimal policy, then there exists a stationary policy which is optimal. In the second part, it is shown that if an optimal stationary policy exists, then it belongs to the class of non-idling policies. Next we show that the optimal non-idling policy must schedule the lowest index connected queue (LICQ). In the last part, it is shown that LICQ policy is not an optimal policy with the help of an example. These four parts are proved in the following four lemmas.

3.2.1 Lemmas

Lemma 1 *If there exists a policy that is backlog optimal everywhere, then there exists a stationary policy that is optimal.*

Proof: Let π be an optimal policy. Let the initial system state be s. Let the sample path be $\{\boldsymbol{A}(t), R(t)\},\ t \geq 1$. Let us denote $u^{\pi}(t)$ as the action chosen by policy π in slot t. Let us shift the optimal policy to the left by one slot and denote it as π'. So, π' is such that $u^{\pi'}(t) = u^{\pi}(t+1),\ \forall t \geq 0$. Next, let us shift the packet and energy arrivals to the left by one slot and let them be $\{\boldsymbol{A}'(t), R'(t)\},\ t \geq 1$. So $\boldsymbol{A}'(t) = \boldsymbol{A}'(t+1)$ & $R'(t) = R(t+1), \forall t \geq 1$. At slot 1, let the system state under policy π be $s' = (\boldsymbol{Q}^{\pi}(1), E^{\pi}(1))$. If the system starts at state s', with arrivals $\{\boldsymbol{A}'(t), R'(t)\},\ t \geq 1$ and under policy π', then

$$\sum_{i=1}^{N} Q_i^{\pi'}(t) = \sum_{i=1}^{N} Q_i^{\pi}(t+1), \quad \forall t \geq 0. \tag{3.3}$$

Since π is optimal for every sample path, π is also optimal for the the shifted packet and energy arrivals $\{\boldsymbol{A}'(t), R'(t)\},\ t \geq 1$. Hence from Eq. 3.3, it follows that π' is also optimal. Thus if optimal policy exists, then there exists a stationary policy which is optimal. ■

Lemma 2 *Optimal stationary policy belongs to the class of non-idling policies.*

Proof: Suppose a policy π_1 which does not belong to non-idling policies is optimal. π_1 has atleast one system state s such that $\mathcal{C}_s \neq \emptyset$, where it idles without choosing any of the connected queues. Let π_2 be a non-idling policy same as policy π_1 except at state s, where it chooses any one connected queue. Then if the system starts at state s, then $\sum_{i=1}^{N} Q_i^{\pi_2}(1) = \sum_{i=1}^{N} Q_i^{\pi_1}(1) - 1$. Thus $\sum_{i=1}^{N} Q_i^{\pi_1}(1) > \sum_{i=1}^{N} Q_i^{\pi_2}(1)$ which is a contradiction. Hence any policy which idles cannot be an optimal policy. ■

Now let us the define the notion of Lowest index connected queue policy, which is used often hereafter.

Definition 5 (Lowest Index Connected Queue (LICQ) Policy) *The non-idling policy which chooses the connected queue with the lowest power requirement is referred as Lowest Index Connected Queue policy.* $u^*(s) = \min \mathcal{C}_s,\ \forall s$ *such that* $|\mathcal{C}_s| > 0$.

Now we state and prove the third of the four lemmas.

Lemma 3 *Among the class of non-idling policies any policy other than the Lowest Index Connected Queue (LICQ) policy is not optimal*

Proof: Let π_1 be an optimal non-idling policy that is different from LICQ policy. Then there exists a state s, such that $\mathcal{C}_s \neq \emptyset$ and $|\mathcal{C}_s| > 1$, in which π_1 does not choose the lowest index connected queue. Lets assume that the system starts in state s. Let $\min \mathcal{C}_s = i$. Let π_{LICQ} be the LICQ policy. Then π_{LICQ} chooses i whereas the other policy π_1 chooses another connected queue, say j, such that $j > i$. Then, energy remaining in policy π_2 is $E(0) - P_i$, which can be written as $kP_1 \leq E(0) - P_i < (k+1)P_1$ for some integer k. From slot 1 till slot k, assume packet arrivals to be $[1\ 0\ \ldots\ 0]^T$ and zero energy arrivals. In slots 1 to k, π_{LICQ} transmit a packet from queue 1 whereas π_1 may transmit from any of the connected queues. At the end of k^{th} slot, energy remaining in the battery under π_1 is strictly smaller than π_{LICQ}. Moreover, the energy remaining in π_{LICQ} is smaller than P_1. At slot k+1, assume packer arrival be $[1\ 0\ \ldots\ 0]^T$ and energy arrival be $R(k+1) = P_1 - (E(0) - P_i - kP_1)$. So, $E^{\pi_{LICQ}}(k+1) = P_1$ and $E^{\pi_1}(k+1) < E^{\pi_{LICQ}}(k+1)$. So, LICQ policy π_{LICQ} chooses queue 1, whereas policy π_1 stays idle in slot $k+1$. Thus $\sum_{i=1}^{N} Q_i^{\pi_{LICQ}}(k+1) \leq \sum_{i=1}^{N} Q_i^{\pi_1}(k+1) - 1$. Hence $\sum_{i=1}^{N} Q_i^{\pi_1}(k+1) > \sum_{i=1}^{N} Q_i^{\pi_{LICQ}}(k+1)$. which is a contradiction. Thus we have shown an example of packet and energy arrivals where every non-idling policy other than LICQ policy fails to attain backlog optimality everywhere. ■

Intuitively, choosing LICQ i.e., queue with the lowest power requirement seems optimal as it retains the most energy in the battery for future transmissions.

Lemma 4 *Lowest Index Connected Queue (LICQ) policy is not an optimal policy.*

Proof: Let us consider a system with $N = 2$. Let the policy π_{LICQ} be optimal. Let us consider π_2 as an idling policy, which transmits packets only from queue 1 and stays idle if queue 1 is not connected. We show an example where policy π_2 achieves lesser backlog than the LICQ policy. Initial state is $(\boldsymbol{Q(0)}, E(0)) = ([0,1]^T, P_2)$. From slot 1 till slot $k-1$, energy arrival, $R(t) = P_2$ and packet arrivals, $A(t) = [0,1]^T$. For all slots greater than k-1, the energy arrival, $R(t) = 0$ and packet arrivals, $A[t] = [1,0]^T$.

Table 3.1: Action in each slot (Queue from which packet is scheduled)

t	0	1	–	k-1	k	k+1	–	$k + \frac{kP_2}{P_1} - 1$
$\pi_{LICQ}(t)$	2	2	–	2	0	0	–	0
$\pi_2(t)$	0	0	–	0	1	1	–	1

$$\sum_{i=1}^{N} Q_i^{\pi_{LICQ}}\left(k\left(1+\frac{P_2}{P_1}\right)\right) - \sum_{i=1}^{N} Q_i^{\pi_2}\left(k\left(1+\frac{P_2}{P_1}\right)\right) = k\left(\frac{P_2}{P_1} - 1\right)$$

We have shown that there exists packet arrivals and energy arrivals under which LICQ policy does not minimize backlog everywhere and hence it is non-optimal in this sense.

■

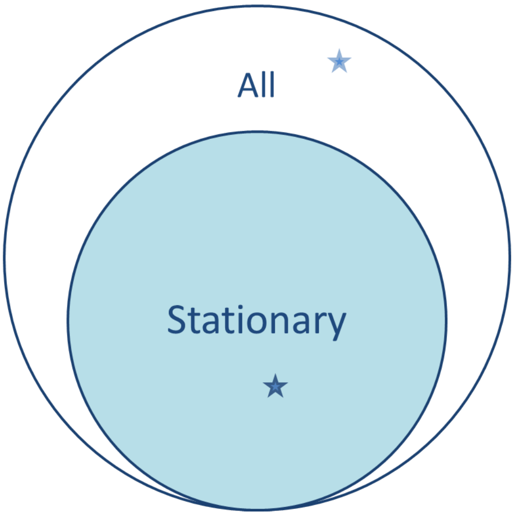
Remark 1 *The example can be generalized to any number of queues.*

Remark 2 *Difference in the backlog under LICQ policy and that under policy π_2 is $k\left(\frac{P_2}{P_1} - 1\right)$ and can become unbounded as k increases. However, battery capacity needs to be at least kP_2. Hence under the given example, the difference between the backlog increases if the battery is scaled appropriately.*

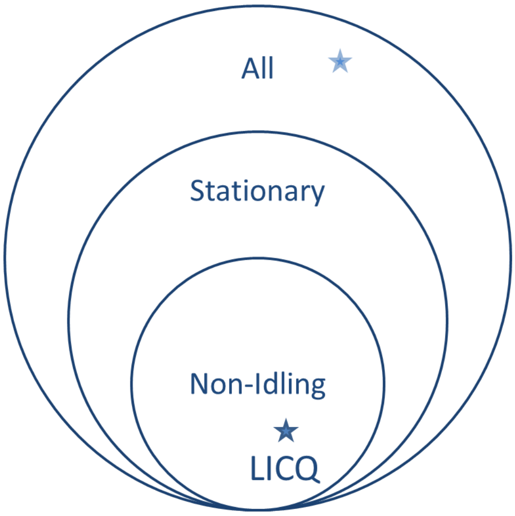
Remark 3 *Even within the class of non-idling policies, it can be shown that LICQ policy is not backlog optimal everywhere. In fact the backlog under π_{LICQ} can grow arbitrarily larger than a non-idling policy along some sample path. Readers are refered to Appendix B for the example, where backlog under π_{LICQ} grows arbitrarily large. Thus, π_{LICQ} policy is not even bounded distance away from optimal.*

The four lemmas are graphically represented in Table 3.2. Here star, represents an optimal policy. The blue coloured area indicates the set in which an optimal policy can exist.

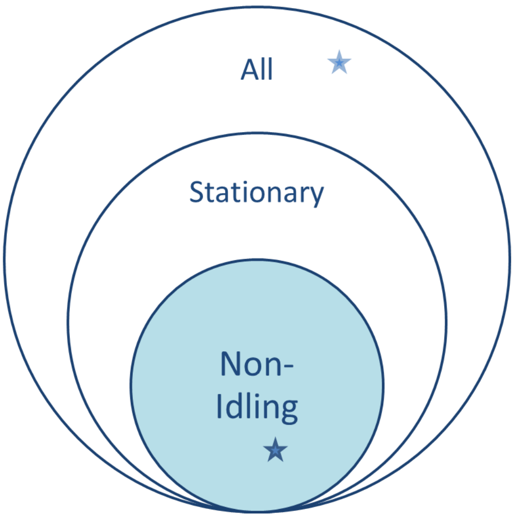
Table 3.2: Lemmas



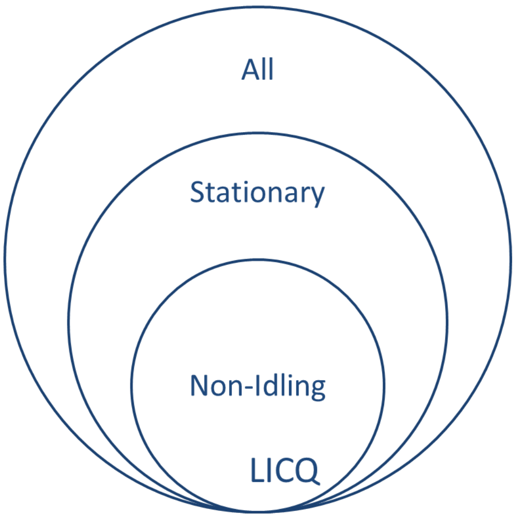
Lemma 1



Lemma 3



Lemma 2



Lemma 4

Proof of Theorem 1: From Lemma 1, we know that if an optimal policy exists, then there exists a stationary policy that is backlog optimal everywhere. In Lemma 2 and Lemma 3, we have shown that if an optimal stationary policy exists, then the optimal policy must be the LICQ policy. Finally, we show that LICQ policy is not backlog optimal everywhere. Hence as a consequence of the four lemmas, it is proved that there does not exist a policy that achieves backlog optimality everywhere. ■

This motivates us to construct policies that are backlog optimal in the expected sense. In the following section, we present our approach in detail. We formulate the problem as a Markov Decision Process (MDP) and determine the optimal policy.

3.3 MDP Formulation and Structural Properties

We first formulate the problem of minimizing backlog as an infinite horizon discounted Markov Decision Process problem. Then we obtain some significant structural properties of an optimal policy. If we look at the long term average backlog, when the system is not stabilizable, the objective function will be infinity for any policy. Hence it does not make sense to minimize average backlog, when system is non stabilizable. Whereas when the system is stabilizable, for a finite battery with large enough capacity, the battery will be almost full in all slots. Hence, all non-empty queues will be connected in every slot and hence any non-idling policy will achieve average backlog optimality. So we take the objective of minimizing expected discounted backlog, with battery capacity to be finite but not so large. In this scenario, it makes sense to take discounted cost, as this gives more weightage to transmitting a packet now than waiting for future slots. By this way, we are spending power as early as possible. This reduces instances of battery overflow and wastage.

3.3.1 MDP Formulation

First let us define the system model assumptions, which are being considered for framing the expected backlog minimization problem as a discounted Markov Decision Process problem. For a user i, the arrival process $\{A_i(t)\}_{t\geq 1}$ is assumed to be independent and identically distributed (i.i.d). The arrival processes for different users are assumed to be independent. Also for simplicity we assume that $A_i(t) \in \{0, 1\}$ for every i and t. The recharge process is $\{R(t)\}_{t\geq 0}$ is i.i.d. Let e_{max} be the maximum value of recharge arrival. Assume $e_{max} < P_1$. We define the system state to be $s = (\boldsymbol{q}, \xi)$, where $\boldsymbol{q}$ denotes the number of packets present in the queues and ξ denotes the energy present in the battery. Note that the state space $\mathcal{S}$ is $N+1$ dimensional. An action chosen by a policy in any state could be either to remain idle or to schedule from a connected queue. Thus, in a state $(\boldsymbol{q}, \xi)$, possible actions are $U(\boldsymbol{q}, \xi) = \{0\} \cup \big\{i : q_i > 0 \text{ and } \xi \geq P_i;\ i \in \{1, \ldots, N\}\big\}$. Action $u = 0$ implies that no queue is scheduled and it is possible in every state. Union of all action spaces are $U = \{0, \ldots, N\}$. We assume the the queue buffer capacity to be large, but finite. The reward function $r : \mathcal{S} \times U \to \Re_+$ is,

$$r(\boldsymbol{q}, \xi, u) = \sum_{i=1}^{N} q_i.$$

Let us consider $\lambda \in (0,1)$ as a discount factor, $u^\pi(t)$ is the queue scheduled by policy π in slot t. We refer to $u^\pi(t)$ as the action taken by policy π in slot t. Let us define the cost function of policy π, $J^\pi : \mathcal{S} \to R_0^+$ for the state $(\boldsymbol{q}, \xi)$ that we start with.

$$\begin{aligned} J^\pi(\boldsymbol{q}, \xi) &= \lim_{T\to\infty} E\left[\sum_{t=0}^{T} \lambda^t r(\boldsymbol{Q^\pi(t)}, E^\pi(t), u^\pi(t)) \right. \\ &\qquad\qquad \left. | S_0 = (\boldsymbol{q}, \xi)\right], \\ &= E\left[\sum_{t=0}^{\infty} \lambda^t \sum_{i=1}^{N} Q_i^\pi(t)\right]. \end{aligned}$$

Note that since queue is finite, reward is finite and hence limit and expectation can be interchanged. Now, let us define the notion of expected backlog optimality.

Definition 6 (Expected backlog optimality) *A scheduling policy π is expected backlog optimal if it satisfies the following relation,*

$$J^\pi(\boldsymbol{q}, \xi) \leq J^{\pi'}(\boldsymbol{q}, \xi) \quad \forall \boldsymbol{q}, \xi. \tag{3.4}$$

Our objective is to minimize the queue backlog at the transmitter. Let us define p_e as the probability of energy arrival value being e, with e assumed to be discrete valued. Let $\alpha_{(\boldsymbol{q},\boldsymbol{q}')}$ be the transition probability from queue state $\boldsymbol{q}$ to $\boldsymbol{q}'$. $p_{\boldsymbol{a}}$ is the probability of packet arrival being $\boldsymbol{a}$ with $\boldsymbol{a} = [a_1, \ldots, a_N]^T$; $a_i \in \{0,1\}$. $\mathbb{I}_u$- $[0, \ldots, 0, 1, 0 \ldots, 0]^T$ is a $N \times 1$ vector with 1 in the u^{th} position, zeros elsewhere; $\mathbb{I}_0$ =zero vector. The optimal reward function satisfies the Bellman's equation of dynamic programming, given by

$$\begin{aligned}
J^*(\boldsymbol{q},\xi) = &\min_{u\in U(\boldsymbol{q},\xi)} \Big\{ r(\boldsymbol{q},\xi,u) \\
&+\lambda \sum_{e=0}^{\xi_m-\xi+P_u-1} p_e \sum_{\boldsymbol{q'}} \alpha_{(\boldsymbol{q},\boldsymbol{q'})} J^*(\boldsymbol{q'},\xi-P_u+e) \\
&+\lambda P(e\geq \xi_m-\xi+P_u)\sum_{\boldsymbol{q'}} \alpha_{(\boldsymbol{q},\boldsymbol{q'})} J^*(\boldsymbol{q'},\xi_m)\Big\}, \\
= &\min_{u\in U(\boldsymbol{q},\xi)} \Big\{ r(\boldsymbol{q},\xi,u) \\
&+\lambda \sum_{e=0}^{\xi_m-\xi+P_u-1} p_e \sum_{\boldsymbol{a}} p_{\boldsymbol{a}} J^*(\boldsymbol{q}-\mathbb{I}_u+\boldsymbol{a},\xi-P_u+e) \\
&+\lambda P(e\geq \xi_m-\xi+P_u)\sum_{\boldsymbol{a}} p_{\boldsymbol{a}} J^*(\boldsymbol{q}-\mathbb{I}_u+\boldsymbol{a},\xi_m)\Big\}.
\end{aligned}$$

At each epoch, the policy maps the state to its optimal action. Since, it is an infinite horizon problem with discounted rewards and state space is finite, we know from [31] that, there exists a stationary deterministic policy which attains optimality. Let $\pi^* = \{u^*, u^*, \ldots\}$ represent the optimal stationary deterministic policy. The optimal action at each state is given by,

$$\begin{aligned}
u^*(\boldsymbol{q},\xi) = &\arg\min_{u\in U(\boldsymbol{q},\xi)} \Big\{ r(\boldsymbol{q},\xi,u) \\
&+\lambda \sum_{e=0}^{\xi_m-\xi+P_u-1} p_e \sum_{\boldsymbol{q'}} \alpha_{(\boldsymbol{q},\boldsymbol{q'})} J^*(\boldsymbol{q'},\xi-P_u+e) \\
&+\lambda P(e\geq \xi_m-\xi+P_u)\sum_{\boldsymbol{a}} p_{\boldsymbol{q'}} J^*(\boldsymbol{q'},\xi_m)\Big\}.
\end{aligned}$$

The optimal policy π^* can be obtained by solving the dynamic programming equation numerically using value iteration or policy iteration approach etc. But for every change in the system parameter, intensive computations have to be carried out to calculate the optimal policy. This does not give us any insight or inference into the optimal policy. In order to understand the optimal policy and have insights about why it chooses such actions, we analytically find some structural properties about an optimal policy in the next subsection.

3.3.2 Structural Properties of an Optimal Policy

In our first result, we show that when available energy level is high, the optimal policy is non-idling.

Theorem 2 (Work Conservation of Optimal Policy) *There exists an energy threshold ξ_{th} such that for every state $s = (q, \xi)$ such that $\xi > \xi_{th}$ and $\mathcal{C}_s > 0$, then the optimal action $u^*(s) \neq 0$.*

Proof: Let us assume a state s such that $\xi > \xi_{th}$ and suppose $u^*(s) = 0$. Let the system start with state s. Let π_1 be non stationary policy which chooses action 0 in slot 0. Let π_2 be an optimal policy. Let us compare between actions 0 and N. There are two

Table 3.3: Action at each slot

slot	0	1	2	-	m	m+1	m+2	-	$m+\frac{P_N}{P_1}$
π_1	0	N	N	$-$	N	1	1	$-$	1
π_2	N	N	N	$-$	N	0	0	$-$	0

possible explanations, according to the nature of recharge values. These are as follows:
a)Let us consider recharge values to be 0 from slot 0 till slot $m + \frac{P_N}{P_1}$. Here m would be $\frac{\xi}{P_N} - 1$ as shown in Table 3.3. At slot $m + 1$, energy in policy π_2 becomes zero. At the end of slot $m + \frac{P_N}{P_1}$, energy in policy π_1 is also zero. Policy π_2 transmits m packets from queue N until battery gets drained. Since, policy π_1 has transmitted only $m-1$ packets in the same number of slots, it has P_N energy more than that of π_1. Now if π_1 transmits from the first queue, it can transmit many packets and reduce the backlog. This is the only way policy π_1 can minimize backlog better than policy π_2. When this happens, we show that if energy is greater than some threshold, policy π_1 can never be better.

$$J^{\pi_1}(\boldsymbol{q}, \xi) - J^{\pi_2}(\boldsymbol{q}, \xi) \geq \overbrace{\lambda + \lambda^2 + \ldots + \lambda^{\frac{\xi}{P_N}}}^{\text{term1}} \\ \underbrace{- \frac{P_N - P_1}{P_1} \lambda^{\frac{\xi}{P_N}+2} (\sum_{k=0}^{\infty} \lambda^k)}_{\text{term2}},$$

$$\text{For (term1} \; - \; \text{term2)} > 0, \xi > P_N \left(\frac{\rho(N)}{\log \lambda} - 1 \right) \geq \xi_{th}, \tag{3.5}$$

where,

$$\rho(j) \triangleq \log \left(\frac{\lambda}{1 + \lambda \left(\frac{P_j}{P_1} - 1 \right)} \right).$$

b) When recharge values from slot 0 are non zero, term 1 in the above equation may increase, term 2 may decrease, so eventually value difference increases. Note that, when we compare with action $j < N$, then the threshold value obtained will be less than that of action N.

$$\text{For } \xi > P_N \left(\frac{\rho(N)}{\log \lambda} - 1 \right) \geq \xi_{th}, \quad u^*(s) \neq 0. \tag{3.6}$$

If energy is greater than this ξ_{th}, then optimal action at this state s, $u^*(s) \neq 0$. ■

Above some energy threshold, it is never optimal to stay idle. Only reason for which a policy may want to stay idle is to wait for packets to arrive in a lower indexed queue rather than transmitting a packet now from a higher indexed queue which may require a lot more energy. However, when enough energy is available, it becomes more prudent to transmit a packet to reduce cost now rather than conserving energy for future potential cost reduction. The value of ξ_{th} depends on the discount factor λ.

Let us assume that the battery capacity $\xi_m > \xi_{th} + P_N$. In the next result, we show that the optimal action is to either remain idle or follow π_{LICQ} i.e. transmit from the lowest index connected queue. Formally, we show the following.

Theorem 3 *At a state $s = (\boldsymbol{q}, \xi)$ such that $|\mathcal{C}_s| > 1$ and if $u^*(s) \neq 0$, then the optimal action is to choose the LICQ. $u^*(s) = \min \mathcal{C}_s = i$.*

Proof: Let π_1 be an optimal policy. Suppose there exists a state s such that $u^*(s) = j$, even when $\min \mathcal{C}_s = i$. Let the system start with state s. Let π_2 be a non-stationary policy, which chooses action i at slot 0. As a consequence of Theorem 2, whenever battery level in policy π_1 crosses ξ_{th}, it transmits and since $e_{max} < P_1$, battery level in π_1 never reaches ξ_m in any sample path. Since we know that $\xi_m > \xi_{th} + P_N \geq \xi_{th} + (P_j - P_i) + e_{max}$, energy level under policy π_2 as well does not reach ξ_m. If optimal policy π_1 chooses action i in some slot, say t' as shown in Table 3.4, then in slots 1 to $t' - 1$, policy π_2 chooses same actions as optimal policy π_1. In slot t', policy π_1 chooses action i and $E^{\pi_2}(t') = E^{\pi_1}(t') + P_j - P - i$, π_2 chooses action j. Hence from slot $t' + 1$, the queue state and energy state are same for both policies π_1 and π_2 and their rewards become equal. It is possible that, sample path under optimal policy π_1 may never reach the state, where it chooses action i. In that case, from slot 1 as energy under policy π_2 is higher, it can do better or as good as policy π_1. By choosing action i in slot 0, there exists a policy which is better or atleast as good as policy π_1. Hence, the optimal action at state s, $u^*(s) = \min \mathcal{C}_s = i$.

■

Slot	0	1	-	-	-	t'
π_1	j	u_2	u_3	$-$	$-$	i
π_2	i	u_2	u_3	$-$	$-$	j

Table 3.4: Illustration for Proof of Theorem 3

Spending lower power saves more energy in battery, so more packets can be transmitted in future and hence backlog is lesser when compared to transmitting from any other connected queue.

In the next result we show that if queue 1 is connected then the optimal action is to choose 1.

Theorem 4 *At a state $s = (\boldsymbol{q}, \xi)$ such that $1 \in \mathcal{C}_s$, then the optimal action at this state, $u^*(s) = 1$.*

Proof: Let π_1 be an optimal policy. Suppose there exists a state s such that $u^*(s) = 0$, even when $\min \mathcal{C}_s = 1$. Let the system start with state s. Let π_2 be a non-stationary policy, which chooses action 1 at slot 0. As a consequence of Theorem 2, battery level in policies π_1 and π_2 never reaches ξ_m in any sample path. Let $\tilde{t} \geq 1$ be the first instance when policy π_1 chooses to transmit a packet from a connected queue, say x. Note that if optimal policy decides never to transmit a packet in any slot, in that case, policy π_2 has lesser reward than π_1 and hence π_2 is better than π_1. So, when $\tilde{t}$ exists, policy π_2 stays idle in slots 1 to $\tilde{t}$. If optimal policy π_1 chooses action 1 in some slot, say t' as shown in Table 3.5, then in slots $\tilde{t}+1$ to $t'-1$, policy π_2 chooses the same actions as optimal policy π_1. In slot t', policy π_1 chooses action 1 and π_2 chooses action x. Hence from slot $t'+1$, the queue state and energy state are same for both policies π_1 and π_2 and their rewards become equal. It is possible that, optimal policy π_1 may never choose action 1. In that case, from slot 1 as energy under policy π_2 is higher, it can do better or atleast as good as policy π_1. At slot $\tilde{t}$, $J^{\pi_1} = J^{\pi_2} + (1 + \lambda + \lambda^2 + \cdots + \lambda^{\tilde{t}-1})$. Between

slot	0	1	-	-	t	-	-	-	t'
π_1	0	0	$-$	0	x	u_1	u_2	$-$	1
π_2	1	0	$-$	0	0	u_1	u_2	$-$	x

Table 3.5: Illustration for Proof of Theorem 4

slots $\tilde{t}$ and t', rewards are same. After t', the state is the same in both π_1 and π_2 and the rewards will be equal. So, $J^{\pi_2} < J^{\pi_1}$ which is a contradiction. Optimal action $u^*(s)$ is not 0. So, when $u^*(s) \neq 0$, we know from Theorem 3, the optimal action $u^*(s) = \min \mathcal{C}_s = 1$.

■

In Theorem 3, there is a necessity for knowing the optimal action to be non zero, whereas here in this theorem we characterize it completely, without any apriori knowledge about the optimal action, that the optimal action is 1, whenever queue 1 is connected. As a consequence of Theorem 3, we show that if only non idling policies are allowed, then π_{LICQ} is optimal.

Corollary 1 *Among the class of non-idling policies, the policy that chooses the connected queue with the lowest index i.e., LICQ policy is expected backlog optimal.*

Proof: Under a non idling policy, whenever there is a connected queue, the action is not 0. According to Theorem 3, it is observed that, transmitting a packet from a connected queue with the lowest power requirement i.e., Lowest Index Connected Queue (LICQ) is better than transmitting from any other connected queue. Thus, LICQ policy is an expected backlog optimal policy among this class of non-idling policies. ■

In this special class of policies, we have completely characterized an optimal policy that minimizes the expected backlog.

In the next section, the simulation results for our system model are presented. We compare our numerical results with our analytical results obtained so far.

3.4 Performance of LICQ Policy

The simulation parameters are as follows. The number of users is $N = 3$. The power required to transmit a packet from the queues are $[P_1\ P_2\ P_3] = [4\ 6\ 9]$ respectively. Number of slots is 100,000 over which the simulations are carried out. The packet arrivals follow Bernoulli process with values 0 and 1 with mean arrival rate $\alpha = [0.1\ 0.1\ 0.1]$. The recharge energy arrivals are of Poisson distribution with mean $\bar{E}$. The battery capacity ξ_m is assumed to be 50 units. Note that the simulations are carried out without the assumptions that $e_{max} < P_1$ and finite queue buffer, which are required for analytical guarantees. Also note that on account of infinite state space, the computation of optimal policy through methods like policy iteration and value iteration is not possible. Hence, we simulate the performance of LICQ policy, which has been shown to be optimal in the class of non idling policies.

3.4.1 Simulation Results

The scenario is simulated and performance of LICQ policy is shown with respect to different metrics. Energy Load of the system= $\alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$. Let us define energy ratio to be $\frac{\alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3}{E}$. Note that the energy ratio is equivalent to Erlang load on the energy queue. When energy ratio is greater than 1, it implies the energy arriving is less than what is required, hence the system is not stabilizable. In Figure 3.4, we plot average delay in the network as a function of energy ratio. As expected, the average delay increases with energy ratio.

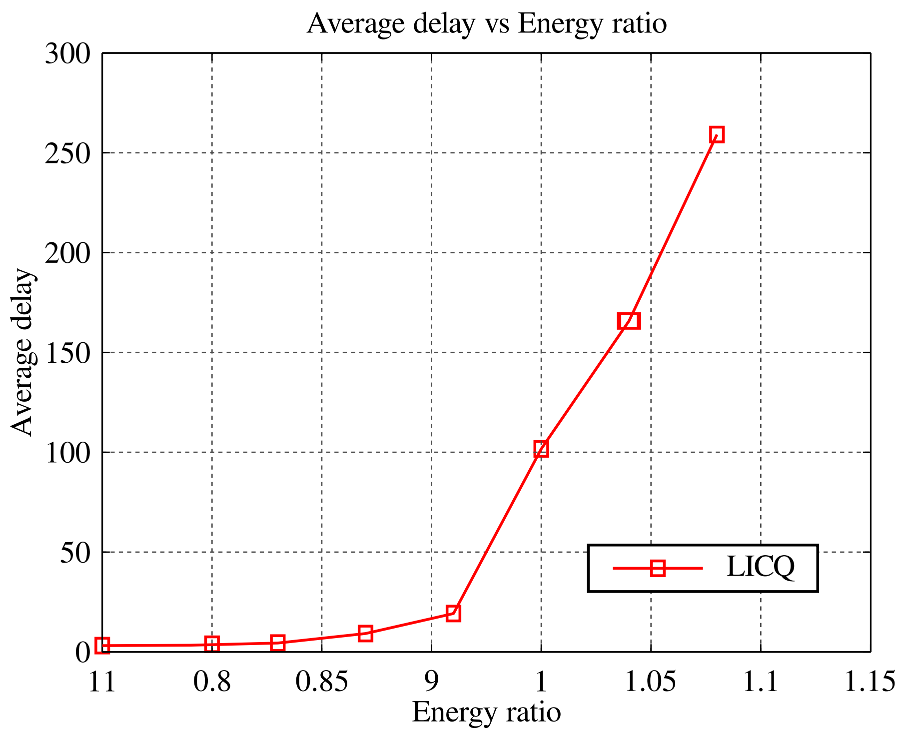


Figure 3.4: Average Delay vs Energy Ratio

In Figure 3.5, using Jain's index, we investigate fairness of the LICQ policy, in terms of delay for various users. Jain Index=$\frac{1}{n}\frac{(x_1+x_2+x_3)^2}{x_1^2+x_2^2+x_3^2}$. where x_i = Number of packets transmitted from queue i/ No. of packets arrived in queue i. It can be shown that, as energy ratio increases, the fairness is affected as most of the times LICQ policy transmits from the lowest index queue.

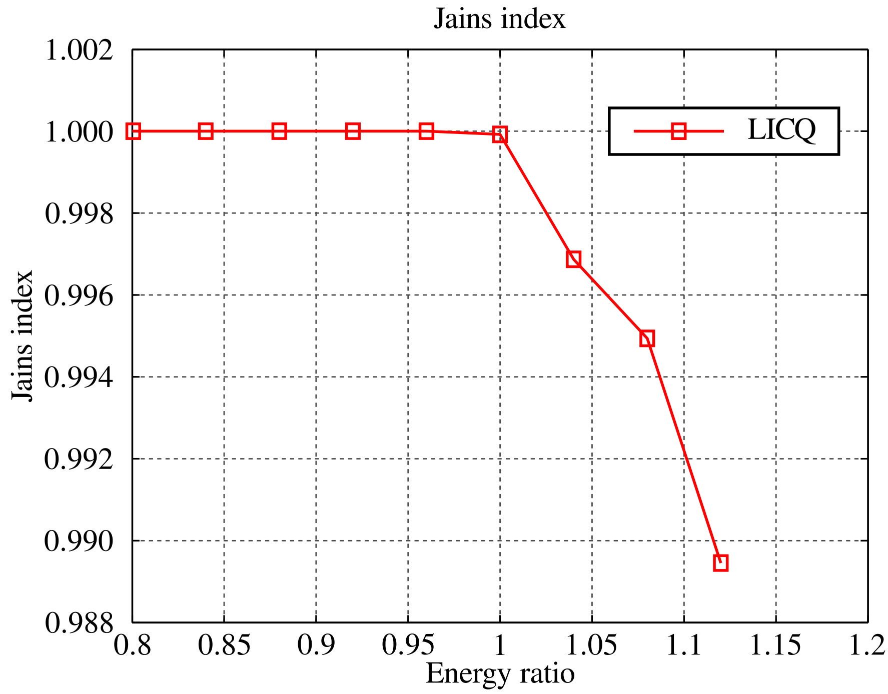


Figure 3.5: Jain Index vs Energy ratio

3.4.2 Comparison of LICQ Policy with Other Policies

We compare the LICQ policy with other policies among which some belong to the class of non-idling policies and some belong to idling policies. Figure 3.6 shows the logarithm of average delay over different values of energy ratio in the four policies. Here battery capacity $\xi_m = 50$. The four policies in Figure 3.6, are

- Lowest Index Connected Queue (LICQ) policy,
- Longest Connected Queue (LCQ) policy,
- Longest Waiting Time Connected Queue (LWCQ) policy. Among the connected, it chooses the queue which has not been served for maximum time.
- Longest Queue (LQ) policy. This is an idling policy. When the longest connected queue is connected, it serves or else stays idle in that slot.

The first three policies belong to the class of non-idling policies.

In Figure 3.7, the average delay of the policies is plotted versus the energy ratio. As expected, when energy ratio becomes than 1, the average delay k.pdf on increasing, as the system is not stabilizable. However, all non-idling policies i.e LICQ, LWCQ, LCQ attain almost the same average delay. When the system is stabilizable, the battery is

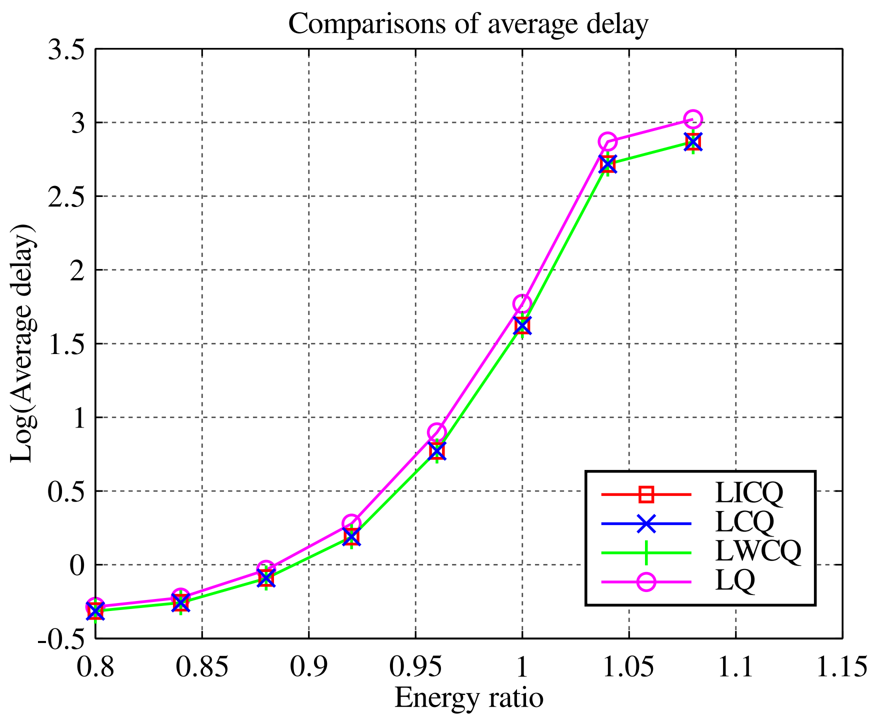


Figure 3.6: Logarithm of Average Delay vs Energy Ratio

almost full in all the slots and hence any non-idling policy will attain the same average delay. Whereas, when system is not stabilizable, then in most of the slots, only lowest index queue is connected and hence any non-idling policy will transmit from the lowest index queue. So, non-idling policies achieve same average delay.

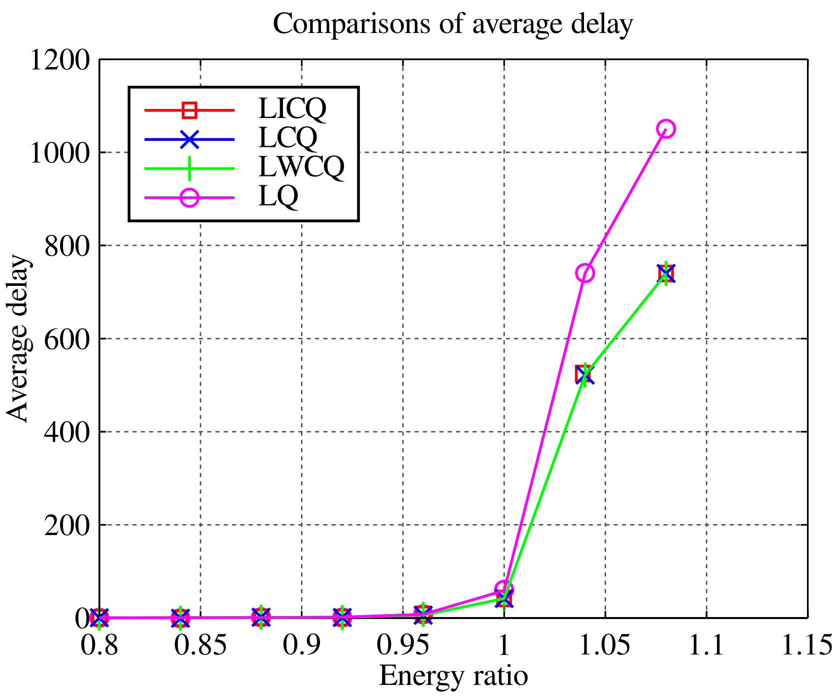


Figure 3.7: Average Delay vs Energy Ratio

In Figure 3.8, the average delay is plotted against different values of battery capacity ξ_m. Here we consider the energy ratio to be 0.96. Under this energy ratio, the system is stabilizable. It is not much clear, as to why the policies perform this way as in Figure 3.8, when the battery capacity is varied.

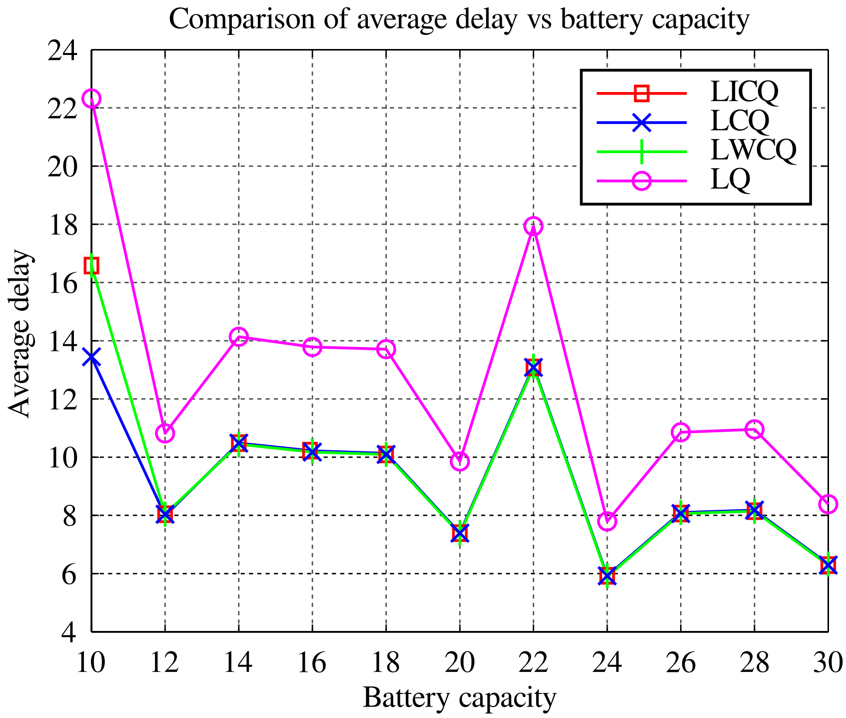


Figure 3.8: Average Delay vs Battery Capacity

Figure 3.9 compares the Jain's index of fairness among the four policies. Here $\xi_m = 50$. Here, longest queue (LQ) policy remains fair even when energy ratio increases beyond 1, because it chooses a packet from the longest queue if it is connected or else waits. It maintains all queue lengths to be same and hence achieves Jain's index of 1 over any energy ratio.

Figure 3.10, Figure 3.11, Figure 3.12 illustrate each user's individual average delay performance in the four different policies. When energy ratio is > 1, the non-idling policies transmits from either queue 1 or queue 2, as they are the queue which will be connected in most of the slots. So non-idling achieve lesser delay for users 1 and 2. Whereas for user 3, it is connected only in a small percentage of the total slots and hence it has large delay under the non-idling policies. But, LQ policy has equal delay in all the users as expected.

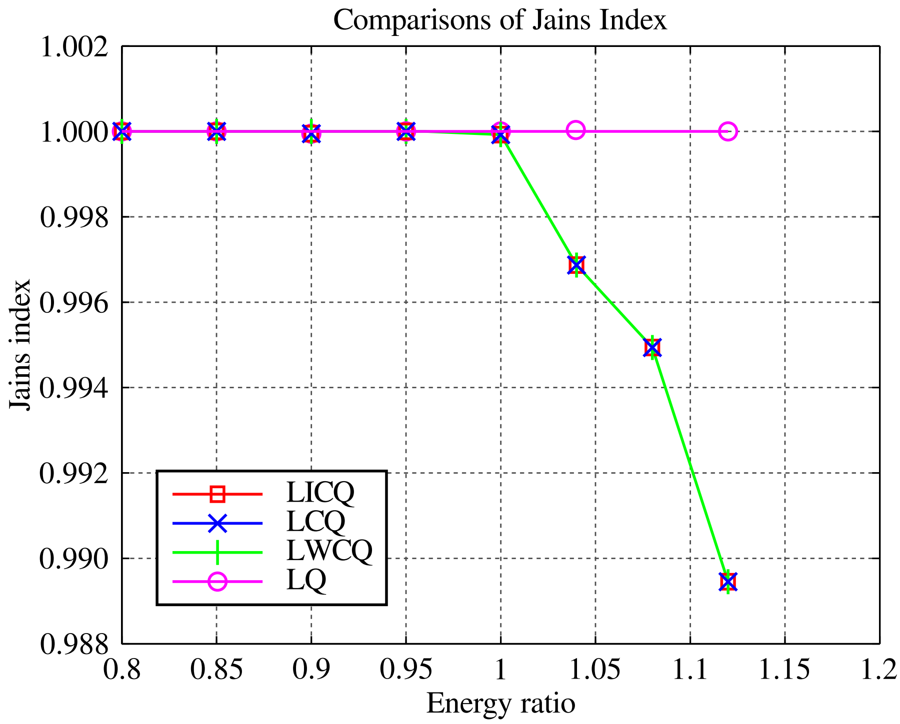


Figure 3.9: Jain Index vs Energy ratio

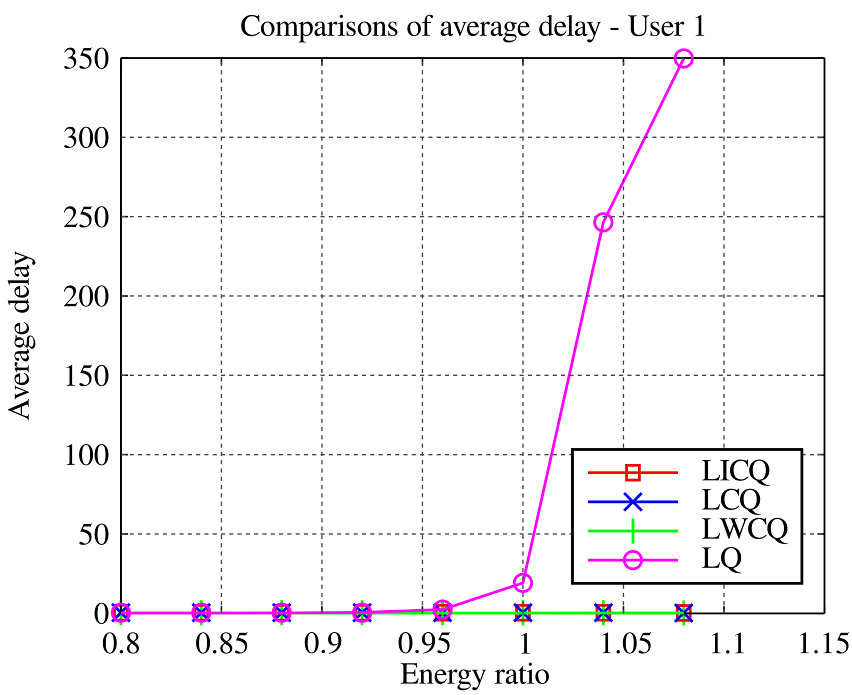


Figure 3.10: User-1 Average Delay vs Energy Ratio

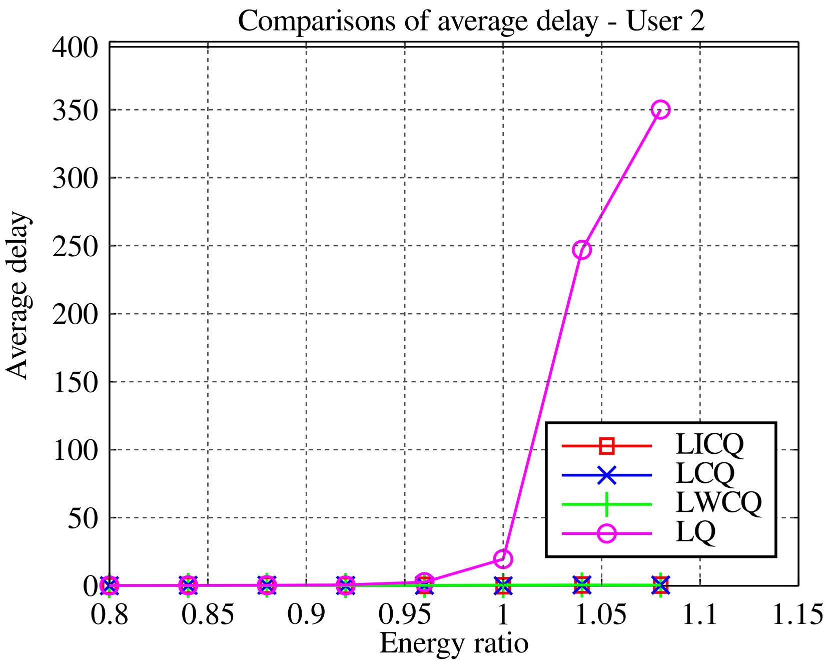


Figure 3.11: User-2 Average Delay vs Energy Ratio

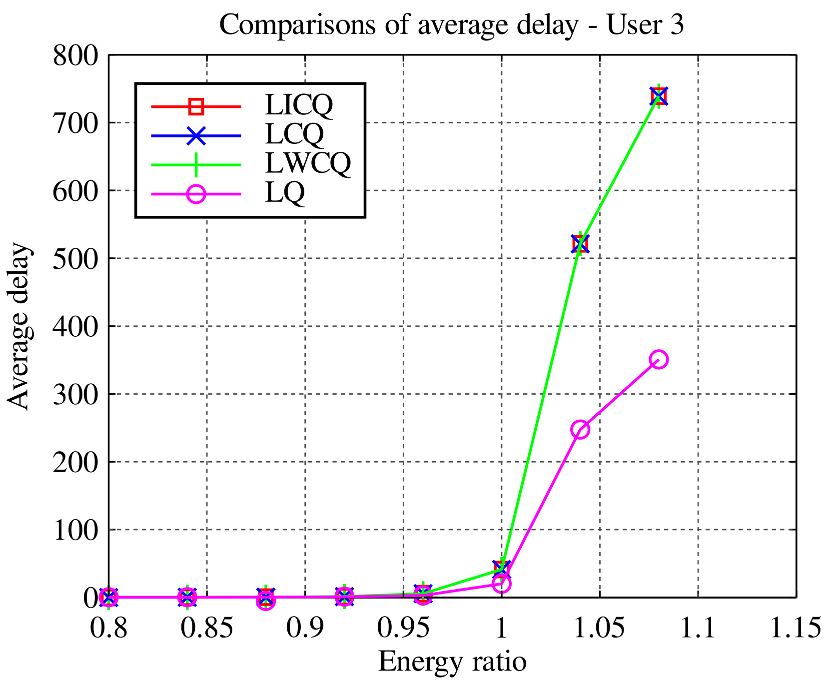


Figure 3.12: User-3 Average Delay vs Energy Ratio

Chapter 4

Conclusions

In N users and 1 base station model, under finite battery setup in minimizing expected backlog, structural properties of an optimal policy have been proved. Importantly, we have shown that above some threshold in battery energy, it is optimal to transmit, rather than staying idle. Among the class of non-idling policies, the policy that schedules the connected queue with the lowest index (LICQ), i.e lowest power requirement is optimal. Hence under this special class of policies, optimal policy is completely characterized. But, the same LICQ policy is not backlog optimal everywhere and is justified via a counter example. From the analysis of backlog optimality at every slot, it can be inferred that with energy being a random value, an optimal policy does not exist. Simulation results justify the analytical results and our intuitions.

There are some limitations in this work. The recharge energy arrival process that we chose does not characterize a real life scenario. Recharge energy processes are non-stationary, for example solar energy is more in daytime and less in nights. The channel we considered is a fixed channel, not a random fading channel. The energy threshold ξ_{th} depended on the discount factor λ.

Random on-off connectivity in the channel can be added to the present model. By this way, channel gains will vary randomly among $\{0, h_i\}$ for the i^{th} user. For this system, we can formulate optimal policies that minimize expected backlog. Another problem is to minimize energy wastage in an uplink system model with fixed channel gains. Each user's battery is powered by green energy. The objective function is to determine the policy that minimizes the maximum energy level in battery in any user. These problems can be investigated as future scope for research.

Appendix A

Markov Decision Process

In this appendix, we give the fundamentals of optimization framework, Markov decision process (MDP). The basics given in this appendix are restricted to dicrete time MDP, as our concern is in discrete time optimization.

A.1 MDP Formulation Parameters

In order to frame a optimization problem into a MDP problem, there are five parameters that one has to figure out.

1. **Decision epochs**

   These are times at which decisions are made. The set T of decision epochs can be either discrete set or a continuum. The set T can be finite, or infinite. Accordingly, they are called finite horizon problem and infinte horizon problem repsectively.

2. **State Space**

   At each decision epoch, the system occupies a state. The set of all possible states is called the state space S.

3. **Action Space**

   At each state s, there are a set of possible actions A_s. The action space A is set of all possible actions. $A = \bigcup_{s \in S} A_s$

4. **Cost Function**

   In decision epoch t,for choosing an action a at state s, the decision maker receives a cost $C_t(s, a)$. For infinite horizon problem, cost is independent of time t. i.e $C(s, a)$

5. **Transition Probability**

   After taking an action, the system state at the next epoch is determined by the probability distribution $p_t(.|s,a)$. For infinite horizon problem, transition probability is independent of time t, i.e $p(.|s,a)$

A Markov decision process is characterized by $\{T, S, A, p_t(.|s,a), C_t(s,a)\}$.

A.1.1 Decision Rules and Policy

Decision Rule: A decision rule gives a procedure/algorithm to decide an action in each state at a decision epoch.

A decision rule can be either memoryless or history dependent. Memoryless means that action in epoch t, a_t depends only on s_t. To choose an action in epoch t, history dependent decision rule depends on the entire history sequence of the states and actions A decision rule can also be classified into deterministic or random. A decision rule is deterministic, if it selects one action with certainty. Whereas, it is randomized, if it only specifies a probability distribution on the set of actions. So as a result, the decision rules can be classified into four categories as combination of deterministic or random and history dependent or memoryless.

Policy: A policy specifies the decision rule to be used in all decision epochs.

A policy π is a sequence of decision rules. $\pi = \{d_1, d_2, ldots\}$. A policy is stationary, if the decision rules are independent of time. i.e if $d_t = d$ for all t. Stationary policies are important in infinite horizon problems.

A.2 Finite Horizon MDP

Assumptions:

1. The decision epochs $T = \{1, 2, \ldots, N\}$

2. State space S is finite or countable

3. Action set A_s is finite for each s

The objective would be to minimize or maximize the sum of costs till epoch N as in Equation A.1.

$$\inf_{\pi \in \Pi} E\left[\sum_{t=1}^{N-1} C_t(X_t, a_t) + C_N(X_N)|X_1 = s\right] \tag{A.1}$$

where Π is set of all policies.

Theorem: Assume state space S is finite or countable. Action set A_s is finite for each $s \in S$. Then there exists a deterministic memoryless policy which is optimal.

The optimal value function $V_n(s)$ satisfies the optimality equation A.3

$$V_n(s) = \min_{\pi \in \Pi} E\left[\sum_{t=n}^{N-1} C_t(X_t, a_t) + C_N(X_N)|X_n = s\right] \tag{A.2}$$

$$V_n(s) = \min_{a \in A_s} \left\{C_t(s, a) + \sum_{j \in S} p_t(j|s, a)V_{t+1}(j)\right\} \tag{A.3}$$

where $V_N(s) = r_N(s)$, some fixed reward on ending state, action a that minimizes the above term defines the optimal policy. Dynamic programming algorithm can be used to solve the equation numerically and obtain the optimal policy. Please refer [31] for the dynamic programming algorithm.

A.3 Infinite Horizon MDP

Depending on the objective function, there are many kinds of infinite horizon Markov decision processes. Among them, two are of interest to us.

A.3.1 Infinite Horizon Discounted MDP

Assumptions:

1. The decision epochs $T = \{1, 2, \ldots\}$
2. State space S is finite or countable
3. Action set A_s is finite for each s
4. Stationary costs and transition probabilities $C(s, a) and p(j|s, a)$, do not vary from time.

5. Bounded costs: $|C(s, a)| \leq M$ for all $s \in S$ and $a \in A_s$

The objective function in a discounted MDP is given in Equation A.4

$$\inf_{\pi \in \Pi} \lim_{N \to \infty} E\left[\sum_{t=1}^{N} \lambda^t C(X_t, a_t) | X_1 = s\right] \tag{A.4}$$

where $0 < \lambda < 1$ is the discount factor. Under the assumptions, the following value function $V^*(s)$ exists:

$$V^*(s) = \inf_{\pi \in \Pi} \lim_{N \to \infty} E\left[\sum_{t=1}^{N} \lambda^t C(X_t, a_t) | X_1 = s\right] \tag{A.5}$$

and satisfies the following optimality equation:

$$V^*(s) = \min_{a \in A_s} \left\{ C(s, a) + \lambda \sum_{j \in S} p(j|s, a) V^*(j) \right\} \tag{A.6}$$

Theorem: With assumptions 1-5, there exists a stationary deterministic policy that is optimal.

The optimality equation A.6 can be solved using different methods which include value iteration, policy iteration and linear programming. Also, it is proven that they will converge to the optimal value and an optimal policy [31].

A.3.2 Infinite Horizon Average Cost MDP

Assumptions:

1. The decision epochs $T = \{1, 2, \ldots\}$

2. State space S is finite or countable

3. Action set A_s is finite for each s

4. Stationary costs and transition probabilities $C(s, a) and p(j|s, a)$, do not vary from time.

5. Bounded costs: $|C(s, a)| \leq M$ for all $s \in S$ and $a \in A_s$

6. The Markov chain corresponding to any stationary deterministic policy contains a single recurrent class.

The objective function in an average cost MDP is given in Equation A.7

$$\inf_{\pi \in \Pi} \lim_{N \to \infty} E\left[\frac{1}{N}\sum_{t=1}^{N} C(X_t, a_t)|X_1 = s\right] \tag{A.7}$$

Theorem: Under assumptions 1-6, there exists a stationary deterministic policy that is optimal.

For average cost MDP as well, iterative algorithms such as value iteration, policy iteration etc. can be used to obtain the optimal policy numerically.

Kindly refer to the book by Puterman [31] for more detailed information about Markov Decision Processes.

Appendix B

Proof of Remark 3

In this appendix, we give the proof of Remark 3. Remark 3 stated that, even within the class of non-idling policies, LICQ policy is not backlog optimal everywhere.

Proof of Remark 3: Take number of users, $N = 3$. Power required to transmit a packet in a slot from queues 1, 2 and 3 are 4, 8 and 12 respectively, i.e. $P_1 = 4$, $P_2 = 8$ and $P_3 = 12$. Let π_{LICQ} represent the Lowest Index Connected Queue policy. Let π_2 be an non-idling policy, which is defined as follows:

1. choose queue i, if $\mathcal{C}_s = \{i\}, \forall i = \{1, 2, 3\}$
2. choose queue 1, if $\mathcal{C}_s = \{1, 2\}$
3. choose queue 3, if $\mathcal{C}_s = \{1, 2, 3\}$
4. choose queue 3, if $\mathcal{C}_s = \{1, 3\} or \{2, 3\}$
5. stay idle, if $\mathcal{C}_s = \emptyset$

For backlog optimality everywhere, intuitively it seems π_{LICQ} may be optimal. A counter example is shown to prove that π_{LICQ} is not optimal. Initial state of the queue is $[1\ 1\ 1]^T$ and energy is 12 which can be seen in Table B.3 and Table B.1 respectively. Action taken by the two policies at every slot can referred from Table B.2. The packet arrivals are

1. $[1\ 1\ 1]^T$ in slots from 1 till $\alpha - 1$.
2. $[0\ 0\ 0]^T$ in slots from α till $3\alpha - 1$
3. $[1\ 0\ 0]^T$ in slots from 3α till $5\alpha - 1$

Table B.1: Energy available at the beginning of each slot

t	0	1	–	α-1	α	α+1	–	2α-1	2α	2α+1	–	3α-1	3α	3α+1	–	5α-1
$E^{\pi_{LICQ}}(t)$	12	20	–	8α+4	8α	8(α-1)	–	8	12	12	–	12	0	0	–	0
$E^{\pi_2}(t)$	12	12	–	12	0	0	–	0	12	20	–	8α+4	8α	8α-4	–	4

Table B.2: Action at each slot

t	0	1	–	α-1	α	α+1	–	2α-1	2α	2α+1	–	3α-1	3α	3α+1	–	5α-1
$\pi_{LICQ}(t)$	1	1	–	1	2	2	–	2	3	3	–	3	0	0	–	0
$\pi_2(t)$	3	3	–	3	0	0	–	0	1	1	–	1	1	1	–	1

The energy arrivals are

1. 12 in slots from 1 till $\alpha - 1$.
2. 0 in slots from α till $2\alpha - 1$
3. 12 in slots from 2α till $3\alpha - 1$
4. 0 in slots from 3α till $5\alpha - 1$

. Here, in this illustrative example, at slot $5\alpha - 1$ slot, $\sum_{i=1}^{N} Q_i^{\pi_{LICQ}}(5\alpha - 1) = (\alpha - 1) + \sum_{i=1}^{N} Q_i^{\pi_2}(5\alpha - 1)$. where α can be any arbitrarily high value.

Remark 4 *The example shown can be generalized to any N queue system.*

Remark 5 *Difference in the backlog under LICQ policy and that under policy π_2 is $\alpha - 1$ and can become unbounded as α increases. However, battery capacity needs to be at least*

Table B.3: Queue state at the beginning of each slot

t	0	1	–	α-1	α	α+1	–	2α-1	2α	2α+1	–	3α-1	3α	3α+1	–	5α-1
$Q_1^{\pi_{LICQ}}(t)$	1	1	–	1	0	0	–	0	0	0	–	0	1	2	–	2α
$Q_1^{\pi_2}(t)$	1	2	–	α	α	α	–	α	α	α-1	–	1	1	1	–	1
$Q_2^{\pi_{LICQ}}(t)$	1	2	–	α	α	α-1	–	1	0	0	–	0	0	0	–	0
$Q_2^{\pi_2}(t)$	1	2	–	α	α	α	–	α	α	α	–	α	α	α	–	α
$Q_3^{\pi_{LICQ}}(t)$	1	2	–	α	α	α	–	α	α	α-1	–	1	0	0	–	0
$Q_3^{\pi_2}(t)$	1	1	–	1	0	0	–	0	0	0	–	0	0	0	–	0

$8\alpha + 4$. *Hence under the given example, the difference between the backlog increases if the battery is scaled appropriately.*

This example clearly shows that π_{LICQ} is not optimal everywhere, in fact the backlog under π_{LICQ} can grow arbitrarily larger than the policy π_2 along some sample path. Thus, π_{LICQ} policy is not even bounded distance away from optimal. ■

Publication

1. Venkhat V, Prasanna Chaporkar and Abhay Karandikar, "Minimizing Backlog for Downlink of Energy Harvesting Networks", Proceedings of 13th International Symposium on Modeling & Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt 2015.

Bibliography

[1] L. Tassiulas and A. Ephremides, "Dynamic scheduling for minimum delay in tandem and parallel constrained queueing models," *Annals of Operations Research*, vol. 48, no. 4, pp. 333–355, 1994.

[2] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *Information Theory, IEEE Transactions on*, vol. 39, pp. 466–478, Mar 1993.

[3] O. Ozel, J. Yang, and S. Ulukus, "Optimal broadcast scheduling for an energy harvesting rechargeable transmitter with a finite capacity battery," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 2193–2203, June 2012.

[4] P. Mitcheson, "Energy harvesting for human wearable and implantable bio-sensors," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 3432–3436, Aug 2010.

[5] Y.-Y. Chen, D. Vasic, F. Costa, W.-J. Wu, and C. Lee, "Self-powered piezoelectric energy harvesting device using velocity control synchronized switching technique," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 1785–1790, Nov 2010.

[6] T. Starner, "Human-powered wearable computing," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 618–629, 1996.

[7] J. Kymissis, C. Kendall, J. Paradiso, and N. Gershenfeld, "Parasitic power harvesting in shoes," in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pp. 132–139, Oct 1998.

[8] N. Salodkar, A. Karandikar, and V. Borkar, "A stable online algorithm for energy-efficient multiuser scheduling," *Mobile Computing, IEEE Transactions on*, vol. 9, pp. 1391–1406, Oct 2010.

[9] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *Information Theory, IEEE Transactions on*, vol. 48, pp. 1135–1149, May 2002.

[10] B. E. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," pp. 709–717, 1999.

[11] N. Salodkar, *Online Algorithms for Delay Constrained Scheduling over a Fading Channel*. PhD thesis, Indian Institute of Technology Bombay, 2015.

[12] N. Salodkar, A. Bhorkar, A. Karandikar, and V. Borkar, "An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel," *Selected Areas in Communications, IEEE Journal on*, vol. 26, pp. 732–742, May 2008.

[13] M. Goyal, A. Kumar, and V. Sharma, "Power constrained and delay optimal policies for scheduling transmission over a fading channel," in *INFOCOM 2003, IEEE*, vol. 1, pp. 311–320, March 2003.

[14] M. Goyal, A. Kumar, and V. Sharma, "Optimal cross-layer scheduling of transmissions over a fading multiaccess channel," *Information Theory, IEEE Transactions on*, vol. 54, pp. 3518–3537, Aug 2008.

[15] P. Chaporkar and S. Sarkar, "Minimizing delay in loss-tolerant mac layer multicast," *Information Theory, IEEE Transactions on*, vol. 52, pp. 4701–4713, Oct 2006.

[16] M. Neely, "Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks," *Networking, IEEE/ACM Transactions on*, vol. 16, pp. 1188–1199, Oct 2008.

[17] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 1732–1743, Sept 2011.

[18] M. Khuzani and P. Mitran, "On online energy harvesting in multiple access communication systems," *Information Theory, IEEE Transactions on*, vol. 60, pp. 1883–1898, March 2014.

[19] O. Ozel and S. Ulukus, "Information-theoretic analysis of an energy harvesting communication system," in *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), IEEE 21st International Symposium on*, Sept 2010.

[20] O. Ozel and S. Ulukus, "Achieving awgn capacity under stochastic energy harvesting," *Information Theory, IEEE Transactions on*, vol. 58, pp. 6471–6483, Oct 2012.

[21] W. Mao and B. Hassibi, "On the capacity of a communication system with energy harvesting and a limited battery," in *Information Theory Proceedings (ISIT), IEEE International Symposium on*, July 2013.

[22] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 1180–1189, March 2012.

[23] J. Yang and S. Ulukus, "Transmission completion time minimization in an energy harvesting system," in *Information Sciences and Systems (CISS), 44th Annual Conference on*, March 2010.

[24] R. Serfozo, "Monotone optimal policies for markov decision processes," in *Stochastic Systems: Modeling, Identification and Optimization, II* (R.-B. Wets, ed.), vol. 6 of *Mathematical Programming Studies*, pp. 202–215, Springer Berlin Heidelberg, 1976.

[25] N. Roseveare and B. Natarajan, "An alternative perspective on utility maximization in energy-harvesting wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 63, pp. 344–356, Jan 2014.

[26] K. Huang, "Spatial throughput of mobile ad hoc networks powered by energy harvesting," *Information Theory, IEEE Transactions on*, vol. 59, pp. 7597–7612, Nov 2013.

[27] O. Orhan and E. Erkip, "Throughput maximization for energy harvesting two-hop networks," in *Information Theory Proceedings (ISIT), IEEE International Symposium on*, July 2013.

[28] R. Vaze, "Competitive ratio analysis of online algorithms to minimize packet transmission time in energy harvesting communication system," in *INFOCOM, Proceedings IEEE*, April 2013.

[29] O. Ozel and S. Ulukus, "On the capacity region of the gaussian mac with batteryless energy harvesting transmitters," in *Global Communications Conference (GLOBECOM), IEEE*, Dec 2012.

[30] R. Vaze, R. Garg, and N. Pathak, "Dynamic power allocation for maximizing throughput in energy-harvesting communication system," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[31] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., 1st ed., 1994.