ADAPTIVE TRANSMISSION FOR ENERGY EFFICIENCY IN WIRELESS DATA NETWORKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Elif Uysal Biyikoglu

June 2003

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Professor Balaji Prabhakar
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Professor Abbas El Gamal

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Professor Benjamin Van Roy

Approved for the University Committee on Graduate Studies:

To the eternal memory of my grandmother,

Suheyla Sevim Alpaslan

Acknowledgements

I wish to express my gratitude to Professor Balaji Prabhakar, for his insightful guidance throughout my time at Stanford. I do not know how he has found so much time to give me detailed and genuine advice on research, academic life and many other topics. His high standards about the novelty and significance of problems, rigorous approach to solving them, and enthusiastic sharing of ideas has made this work possible, and taught me a lot about research.

Working with Professor Abbas El Gamal has been an outstanding experience. I learned a lot, from not only his extensive knowledge of Information Theory, but also from his energetic approach to both applied and theoretical research. I think my future work will carry the influence of his perfectionism.

Looking back at the last four years, I realize that it was a very good choice to come to Stanford, for reasons that I was not completely aware of when I decided to come. The interaction of the university with the Silicon Valley makes it a dynamic and extremely exciting environment. I happened to experience this environment at an extraordinary point in time: from the incredible economic boom of 1999 to the bursting of the "dot-com bubble" and the following recession. Witnessing this whirlwind of change contributed immensely to my understanding of real and lasting technology and its relation to research.

The four-year Stanford Graduate Fellowship I was awarded enabled me to do research free from any funding concerns. I thank Texas Instruments for providing my fellowship.

I thank Professor Ben Van Roy for being on my thesis committee. His comments have much improved the presentation of this work. Professor Andrea Goldsmith has also provided useful critique.

Interactions with colleagues at Stanford have been very beneficial in many ways. Dr. Haitham Hindi, Abtin Keshavarzian, Chandra Nair, Dr. Rong Pan, Dr. Dana Porrat, Dr. Kostas Psounis, Devavrat Shah, Mayank Sharma, Stavros Toumpis are among these colleagues. It has been enriching to know them.

I thank Arati Maneshwar and Dr. Falk Herrmann of Bosch Research, Palo Alto, for their genuine interest in this work and sharing their implementation experience in the area of ad-hoc sensor networks, which added an extra dimension to this dissertation. Weekly meetings with Professor Nick McKeown and his research group in the area of high-performance networking have also been useful in giving perspective.

I thank my parents, sister and my grandmother for the support and encouragement they offered, from half a world away. I thank my husband Bulent Biyikoglu for unintentionally causing me to come to Stanford! Words cannot express my gratitude for his generosity, kindness and patience throughout these years. Finally, I thank my baby son Yunus for being the biggest motivator for me to finish writing up in due time!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, the concept of "anytime, anywhere" communication has captured popular interest. It seems that the world awaits with excitement a lifestyle where people as well as devices can communicate with each other and interact with the Internet seamlessly, even as their locations change. Such portable/mobile *pervasive* networking requires data networks that are integrated and easier to use, and are partially or completely wireless. Pervasive wireless networks, as well as being envisioned to provide mobile connectivity for various aspects of modern life, *e.g.*, information, automation, security, and medical and educational applications, also have a dual role in less developed parts of the world. In regions where telecommunication is just burgeoning, wireless networks can make it possible to bypass the setup of costly wireline infrastructure, enabling these regions to quickly catch up with the information age.

A big challenge in making pervasive wireless networking a reality is energy-efficiency. Wireless nodes, especially those that are portable or mobile, typically live on battery energy. The rate at which energy is drained in relation to how often the battery can be recharged is key to determining battery size, which influences the weight, size, and cost of nodes. Finite battery capacity implies either a finite lifetime or a limited duration between rechargings of each node, which certainly limits the amount of mobility and in some cases, such as in multi-hop networks, constrains the network topology.

Multi-hop wireless networks have attracted renewed attention recently [45, 58, 39, 10, 70]. As opposed to centralized networks (such as cellular networks) where nodes transmit to and receive from a fixed base station only, multi-hop wireless networks include peer-to-peer communication. The increasing importance of such networks is largely due to their potential for *ad-hoc networking*: establishing and maintaining communication in the absence of a pre-established infrastructure. Clearly, ad-hoc networking is a key element of pervasive wireless networks. At present, there are various classes of ad-hoc networks, with widely differing characteristics. Sensor networks, for example, typically include hundreds or thousands of nodes, with low data rates, on the order of a few Kbps [51, 79]. Some wireless local area networks (LANs) and personal area networks (PANs), on the other hand, are also ad-hoc, but they usually have less than a hundred nodes while data rates are on the order of Mbps. Despite the variety, these networks have in common the property that they are energy-constrained [70, 37, 81].

Energy-efficiency in wireless communication and networking is the central theme of this thesis, and central to our results is the observation that transmission energy can be controlled by varying transmission rate: with information-theoretically optimal, as well as practical suboptimal coding schemes, the energy needed to transmit a certain amount of data[1] can be reduced by transmitting the data at a *lower rate*. Meanwhile, lowering transmission rate corresponds to lowering the rate at which data packets are served, which causes increased *delay*. Hence, there is an inherent tradeoff between energy and delay.

The energy-delay tradeoff is not captured by traditional information-theoretic treatments of wireless communication, where the transmitter is modeled to have an infinite supply of bits and the objective is to maximize the rate of transmission under a power constraint. Such a model is suitable when data is generated continuously at a fixed rate, such as in voice communication. However, in data applications it is increasingly the case that the

[1] Information-theoretically, a finite amount of data cannot be transmitted reliably with finite energy [74], however this technical detail can be surmounted by allowing a small but non-zero error probability.

rate at which data is generated is unknown, and the data rate also varies in time. Moreover, data is usually delay constrained. The conventional information-theoretic treatment, which ignores the burstiness or the randomness in data arrivals, does not address delay [34, 27].

On the other hand, multiaccess network theory does analyze network issues such as delay, buffer overflow etc., and suggests algorithms such as ALOHA and CSMA that can control these quantities. However, physical layer aspects such as coding, channel modeling and decoding are usually left out of the models. Addressing the problem of minimizing energy while having a handle on delay requires a model that embodies queuing-theoretic as well as information-theoretic aspects [34, 27, 12, 82]. Constructing such a model that captures the energy-delay tradeoff, that yields itself to analysis and suggests practical rate/power allocation algorithms is the main goal of this dissertation. These rate/power allocation algorithms will be referred to as packet *scheduling* algorithms. Operationally, the scheduling algorithms will work on several layers of the network protocol stack at once; specifically, the physical, link and network layers. We believe that a cross-layer approach like the one here will be useful for achieving the full potential of wireless networks.

1.1 Outline of Thesis

A review of the power control literature, which places this work in perspective, is done in Chapter 2. A significant portion of the literature on power control is comprised of information-theoretic studies where the objective is to maximize achievable rate for given average power constraints by adapting to the time variation in the channel. Power control algorithms suggested by these studies operate on the physical and link layers of the protocol stack. In another major portion of power control research, the adaptation is to interference. Algorithms that target the medium access (MAC) and network layers are developed. It has been understood that a truly satisfactory treatment of wireless networking needs to address multi-access issues as well as channel fading, and in addition, queuing issues due to random arrival of messages [12, 82]. In this regard, we survey recent works which have taken a *cross-layer* approach to the problem as we do in this dissertation.

Chapter 3 sets up the minimum-energy scheduling problem formulation. The problem is to minimize the total energy to transmit an arbitrary sequence of packets arriving in a finite time window, within a finite deadline. No specific channel model or coding/modulation scheme is assumed: the model relies on the sole assumption that transmission energy per bit is a convex, decreasing function of the number of channel uses per bit. In fact, this assumption holds in most theoretical or practical cases of interest, as argued in Chapter 3 and also by [12]. Having a finite deadline constraint on a sequence of packets not only models scenarios of practical interest, but also renders an exact *offline* analysis of the problem possible. In offline analysis, it is assumed that events in the future are known ahead of time, and the optimal solution under this extremely optimistic assumption is found. The solution then presents a bound on any solution with more realistic assumptions, such as, for example, an *online* solution which knows only the past and present. In this thesis, offline analysis not only provides us with bounds as such, but with insight about how to obtain good online solutions.

Chapter 4 extends the problem formulation to multi-user channels. The formulation is generalized to cases such as *uplink* (multiple transmitters, single receiver) and *downlink* (single transmitter, multiple receivers), with or without channel fading. The uplink and the downlink are modeled using the multiaccess and broadcast channel models in information theory, respectively.

In the uplink case, one possibility is to time-share between different users' packet transmissions. Although it is well-known that time-sharing between users is strictly sub-optimal except in the special case of users having completely symmetric channels (see [69]), from an application viewpoint it might be practical to do. The *best time-sharing* offline schedule is found by an iterative algorithm called *MoveRight*. The MoveRight algorithm uses the property that energy functions are convex and decreasing, to progressively converge to the optimal solution by making a series of simple local optimizations. In order to find the *optimal* offline schedule, one must consider multiuser coding, where codewords of different users overlap in time. In this case, we exhibit an iterative algorithm, called *FlowRight*, that yields the optimal offline schedule. FlowRight is in the spirit of the MoveRight algorithm, the principle behind its convergence being the convexity and monotonicity of the energy functions. It solves not only the uplink offline optimal scheduling problem, but also many other varieties including the downlink problem, as well as offline scheduling in single or multi-user fading channels. FlowRight also produces the optimal offline solution under various model variations such as individual deadlines for packets, finite buffer constraints, etc.

The offline solutions can be carried on to the practically interesting case of online scheduling by using the idea of a look-ahead buffer. By observing users' packet arrival processes for a certain amount of time and buffering packets in the mean time, one can schedule them with offline optimal algorithms, at the expense of a predictable, bounded delay. In Chapter 5, we exhibit this online scheduling heuristic and apply it to the online versions of all scenarios mentioned above. In the particular case of a slowly fading channel, it is observed that adapting to packet arrival processes in this manner can provide great savings in energy compared to a popular benchmark, which is water-filling adaptation to the fading channel.

Finally, Chapter 6 presents conclusions and directions for further work.

Chapter 2

Power Control

2.1 Introduction

This chapter will briefly review power control research. Power control, the problem of assigning code rates and/or power levels to transmitters in a wireless network, has commonalities with other network resource allocation problems such as routing, flow control, admission control. However, the fact that the channel characteristics are usually variable in time due to interference and/or fading distinguishes the wireless power control problem and makes it challenging.

In wireless networks, users potentially co-exist in a medium where they interfere with each other if they collide in time and frequency. Limitations on battery capacity at terminals, and regulatory/economic limitations on bandwidth make up the constraint region in which the power control problem is set. Whether nodes can communicate reliably, how far they can transmit messages, how long they can continue communicating, and how much interference they cause on each other depends on their stored energy and how they regulate this energy. Time-variability in the channel presents another reason why power needs to be controlled over time.

A large number of previous studies were motivated by spread-spectrum cellular networks. The number of users that can be accommodated in a cell, *i.e.*, network capacity, is heavily influenced by the amount of interference that users impose on each other. In this setting, power control policies manage the interference to maximize network capacity. In contrast, another large body of research starts by looking at a single interference-free link, in a time-varying channel, with an information-theoretic approach. Determination of the capacity of the fading channel leads to optimal power control algorithms. These results have been extended to multi-user settings.

After briefly reviewing those two groups of studies, we describe a third group, comprised of relatively recent research where network-theoretic and information-theoretic methods are combined. Studies in the third group also differ from those in the first two groups in that rather than limiting their scope to the networking layer or the physical layer, they have an inter-layer approach.

2.2 Network Capacity-centric Power Control Research

Most of the earlier work on power control in the wireless setting has considered multiple users in a cellular network, accessing a common base station [63, 46, 38, 30, 49]. The main motivation for those works has been systems like IS-95 CDMA, where interferers are treated as noise.

A simple form of power control that has been used in practice is scaling transmitted power such that users are received at equal power levels at the base station. When interference is treated as noise, this scaling is necessary for all users to achieve the same rate. To see this, consider the AWGN multiaccess channel model:

$$Y = X_1 + X_2 + Z$$

where X_1 and X_2 are the signals from users 1 and 2, of received powers P_1 and P_2, respectively, Z is Gaussian noise, of power N. When each user considers the other as noise, their achievable rates are bounded as $R_1 \leq (1/2)\log_2(1 + P_1/(P_2 + N))$ and $R_2 \leq (1/2)\log_2(1 + P_2/(P_1 + N))$. Hence, to maximize the rate R achieavable by all users at the same time, they must be received at an equal power level. This usually means that far away transmitters need to boost their power levels by a larger amount to overcome attenuation (also called "combatting the near-far problem".)

Such adaptation relies on the knowledge at the transmitter of the channel gain from the transmitter to the receiver. In practice, this knowledge may be obtained by feedback, with a certain delay.

Algorithms to control co-channel interference (interference that users using the same channel (code or time slot) cause on each other) in order to increase network capacity have been the focus of [46, 60, 80, 85, 32, 86, 8, 9]. In [85], the downlink of a cellular system is modeled, with channel gain matrix $Z = \{Z_{ij}\}$, Z_{ij} being the gain from transmitter i to receiver j. An interesting problem is that of finding the maximum C/I ratio (signal power to interference power ratio) simultaneously achievable at all receivers. It is shown that the maximum achievable C/I is given by $1/(\lambda^* - 1)$, where λ^* is the largest eigenvalue of the matrix Z, and the transmit power vector that achieves this is the corresponding eigenvector. Algorithms to approximate this optimal power control are presented in [85].

In [32], Foschini and Miljanic propose a distributed autonomous power control algorithm and explore its convergence properties. In [86] these power control schemes are shown to be robust in the presence of implementation details. Asynchronous, iterative power control algorithms are explored in [80].

"Active link protection" [8] refers to maintaining the quality of service of operational (active) links above given thresholds at all times, and admitting new connections (users) into the network as resources permit. A suite of distributed and autonomous admission/power control algorithms is studied in [8]. Power-controlled shared channel access for packetized data traffic, incorporating various costs for packet transmission and queuing, is studied in

[9]. In most of the above papers, the communication part of the problem is summarized into a desired SINR (or C/I) criterion. Communication-theoretic concepts are usually abstracted out. In the next section, we describe another large body of research where the approach is communication-theoretic.

2.3 Information Capacity-centric Power Control Research

Consider the discrete-time AWGN channel where signal is affected by multiplicative gain:

$$Y[n] = S[n]X[n] + Z[n],$$

where $X[n]$ is the transmitted symbol at time n, $S[n]$ is a channel gain, and $Z[n]$ is Gaussian noise. In general the signal is complex, but assuming the receiver can detect the phase perfectly and invert it, the channel reduces to two parallel real-valued channels. In this section, and in the rest of the thesis it will be understood that we consider one of these channels. The above will be used as a model for frequency-nonselective fading channels, and though it is in discrete-time, the results below are also valid for the continuous-time channel that it represents.

While a physically motivated model of a fading channel usually has an uncountably infinite state-space, finite state models models have been used often [54, 41], and have been shown to closely approximate capacity and power control results for continuous counterparts (see, for example, [42]). Many researchers studied capacity and power control problems under the assumption of perfect CSI (Channel State Information) at the receiver and transmitter. This refers to instantaneous knowledge of $S[n]$, and is quite different from considering delayed feedback from the receiver to the transmitter, which is a technically much more involved problem [54].

Under perfect CSI at the transmitter and receiver, the capacity of a finite-state ergodic channel with infinite input and output alphabets is ([14])

$$C = \sum_{i=1}^{|\mathcal{S}|} p_i C_i$$

where $|\mathcal{S}|$ denotes the cardinality of the state space, and p_i is the steady-state probability of state i. As shown in [44], the above result can be obtained as the limit of the maximum average mutual information between the input and the output as the blocklength grows:

$$C = \lim_{N\to\infty} \frac{1}{N} \max I(X^N; Y^N | S^N)$$

For the AWGN channel,

$$C = \sum_{i=1}^{|\mathcal{S}|} p_i \frac{1}{2} \log(1 + \frac{s_i^2 P_i}{\sigma^2}) \tag{2.1}$$

where σ^2 is the noise variance, s_i^2 is the i^{th} realization of the channel gain, and P_i is the power allocated to state s_i according to some power control policy σ. We now describe two power control policies that we shall refer to later.

Definition: Channel Inversion Let P_r be some desired received power level. Then, $P_i = P_r/s_i$, for all $i \in |\mathcal{S}|$.

Definition: Water-filling over Channel States Let P be the average transmit power constraint. $P_i = \{\nu - \sigma^2/s_i^2\}^+$ for all $i \in |\mathcal{S}|$, where ν is chosen such that $\sum_{i=1}^{|\mathcal{S}|} p_i P_i = \bar{P}$.

A problem that has attracted much attention is finding the power allocation that maximizes the expression in Equation 3.1, under an average power constraint:

$$\max_\sigma \sum_{i=1}^{|\mathcal{S}|} p_i \frac{1}{2} \log(1 + \frac{s_i^2 P_i}{\sigma^2})$$

$$\text{subject to: } \sum_{i=1}^{|\mathcal{S}|} p_i P_i \leq \bar{P}$$

The policy that achieves the above maximum is water-filling. Note from the definition of water-filling that it calls for a variable-rate, variable power coding scheme. Such a scheme can be formed by multiplexing Gaussian codebooks with power P_i and rate $\frac{1}{2}\log(1+P_i/\bar{P})$. Surprizingly, it has been shown in [15] that a single Gaussian codebook is sufficient provided that the code symbols are amplified by $\sqrt{P_i/\bar{P}}$ before being sent on the channel.

The absence of CSI at the transmitter results in only a small loss in capacity [14]. The capacity in this case is $\frac{1}{2}\sum_{i=1}^{|\mathcal{S}|} p_i log(1+\bar{P}s_i/\sigma^2)$, achieved by a single Gaussian codebook of power $\bar{P}$. Note that achieving this capacity (and the previous one with transmitter CSI) requires averaging over the statistics of the channel gain. Achieving this averaging in time may take arbitrarily long, depending on the dynamics of the fading. When codeword length is limited by practical delay constraints, these power control policies may achieve a rate much less than the promised ergodic capacity. In general, for a given rate and reliability, there is a tradeoff between delay and power: the more power-efficient one wishes to be, the more delay one needs to tolerate.

The impracticality in real applications of large delay motivates "outage capacity", and "delay-limited capacity". Outage capacity is the rate that is achievable with probability $1-p_o$, where p_o is the probability that a communication outage is declared. The zero-outage condition leads to delay-limited capacity, the reliable communication rate that can be achieved under all possible realizations of the given channel state distribution. It is achieved by channel inversion [17]. Klein shows in [54] that for a given fading distribution, the power control policy that maximizes the minimum rate across channel states is channel inversion. Note that for some distributions the event $\{s_i = 0\}$ has positive probability, in which case channel inversion would result in infinite power, those cases are usually handled by "truncated" channel inversion. If, rather than absolute delay constraints, there is a minimum rate constraint, the optimal power allocation is a combination of channel inversion and waterfilling: the channel is inverted so as to always achieve a certain SNR and guarantee the minimum rate, and any power left over is distributed over states according to water-filling [54].

Caire et al. address the interesting case of the transmitter not knowing the statistics of the fading, while having causal CSI [18]. They propose a heuristic, which consists of finding what would be the optimal power allocation over all the states in the past and using this water-filling solution for the present time.

In the infinite-horizon power control problem, perfect causal CSI at the transmitter and receiver is sufficient: if the transmitter knew all future channel states, it could not have obtained a higher average rate. This is not true, however, in the finite-horizon case: information about future channel states can be used to optimally allocate power across time. Note, however, that in the finite-horizon case one cannot achieve the ergodic capacity.

For the multiaccess channel, Knopp and Humblet show [55] that when users have symmetric channel statistics and equal power constraints, letting the user with the stronger channel to transmit (ties broken arbitrarily), maximizes the sum of long-term average rates. For the user that transmits, the power allocation is according to water-filling. This result is generalized to the case of maximize an arbitrary linear combination (as opposed to the sum) of user's average rates in [69], where a "greedy algorithm" finds the optimal power and rate allocation.

The multiacces studies cited thus far make the simplifying assumption that users' fading realizations do not change during a codeword. The same assumption will be used throughout the thesis.

2.4 Studies with an Inter-layer Focus

As we pointed out in Chapter 1, a fundamental question in a wireless network is how much energy is needed to transmit a certain amount data within a certain duration. The energy-delay tradeoff has seen increased attention from the research community partly due to emerging low-power pervasive computing applications such as sensor networks and ad-hoc networks, and the move from voice to various data applications on cellular systems. In such networks, energy-efficiency is influenced by network topology, routing, buffering, as well as the traditional lower-layer components such as the multiaccess technique, modulation and coding, system and circuit design. To answer the energy-optimality question, most of these factors need to be considered together.

Minimum energy communication is a fundamental scientific question. Recently, Berger pointed out in his Shannon Lecture [11] that intra-organism communication in biological systems is exremely efficient[1] due to the organism's ability to adapt the channel properties to the source rate. Additionally, intra-organism commununication is usually of multicast nature, with networking capability much more efficient than in networks that can be built currently. These observations indicate that energy-efficiency of networks may be greatly increased by designing source coding, communication and networking together.

Queuing theory was used together with information theory in the modeling of a multiaccess system by Telatar and Gallager in [68]. Prior to that work, queuing theory was used to analyze network capacity and stability in collision-resolution type systems (i.e. ALOHA, CSMA), but in such analyses noise was ignored and interference was dealt with rather naively. On the other hand, precise information-theoretic models for multiuser communication were developed [25], but these models ignored the random arrival of messages into the buffer, treated them as a continuous stream, ignoring delay. To combine the strengths of both treatments, [68] proposed the model of a multiaccess system where each message arrives to a new transmitter, and the total rate at which all the active transmitters serve packets is determined using error exponents: the service rate as a function of the number of active users u is given by

$$\phi(u) = W\rho u \ln\left(1 + \frac{P}{(1+\rho)(N_o W + (u-1)P)}\right)$$

where $2W$ is the two-sided bandwidth in Hz, P is the power used by each transmitter, $N_o/2$ is the noise density, for any $0 < \rho < 1$. Each transmitter codes its packet into an infinitely long codeword and starts transmitting it to the receiver. Once the receiver gets

[1] Another study that gives evidence to the efficiency of biological systems is [1].

enough symbols to decode with sufficient reliability, it sends feedback to the transmitter to stop. For a desired error probability P_e, the service demand of a packet is $-(\ln P_e) + \rho \ln M$, where $\ln M$ is message length in nats. This is observed to be a processor-sharing queue where service rate depends on the number of jobs in the system, and using known results about processor-sharing (see [53]) the stability region of the multiaccess system is determined. Tradeoffs such as delay versus error probability are explored.

A recent paper by Yeh [84] extends this formulation to a multiaccess scenario where messages get queued at separate transmitters. It is shown, in the symmetric case, that the average system delay for packets is minimized by a *Longer-Queue Higher-Rate* allocation strategy.

To our knowledge, the earliest appearance of joint queue state/channel state adaptive power control is in a paper by Collins and Cruz [22]. A more comprehensive treatment appears in [12], where the objective is to obtain the optimal power-delay tradeoff curve and develop algorithms that minimize power while keeping the buffer size below a certain level.

In [83], the minimum-delay power control problem (with a given power constraint) is posed. It is shown that under the two extremes of very fast fading and very slow fading, the optimal power control policy goes to the two extremes of water-filling in time, and channel inversion, respectively. In [21], a time-slotted multiple user system with bursty arrivals is considered. Time-slots are assumed to be long enough to achieve capacity over them. The difference in average energy consumption when delay is minimized with and without the knowledge of other user's queues is explored. It is found, using the results of [59], that a sufficient condition for stability is to transmit using multiple access codes after the queue states cross a finite threshold. Medard et al. [59] showed that the capacity region of the time-slotted ALOHA system with power-constrained users is the same as the capacity region of the multiple access channel.

Delay-optimal scheduling of data packets from different queues is studied in the context of satellite transmission systems in [62, 33]. In [62], it is shown, using Lyapunov methods, that the througput-optimal schedule for a system of several queues is a maximum weight matching where the weight of each queue is the backlog multiplied by the rate at which it can transmit at that time (transmitters have fixed power constraint and rate varies with channel state). In a related study, [33], optimal energy allocation and admission control for communication satellites in earth orbit is considered. The goal is to choose which data requests to serve at a given time, in order to maximize expected total reward when the energy that can be stored is finite and is replenished at certain regular intervals. An optimal policy is derived, using dynamic programming. Energy-efficient caching and on-demand transmission of data is considered in [40]. Optimal online schedules are developed using dynamic programming.

The studies we have cited thus far dealt solely with transmission power. Note that transmission power is just one of the components of energy drain in a wireless terminal. In fact, energy consumption due to source and channel coding/decoding, keeping the system in standby mode, etc. can be significant, and may even exceed the transmission component in some systems. A number of recent works have addressed total system energy minimization (see [28, 3], and references therein). For example, [28] addresses the problem of optimizing the power consumption due to compression, channel coding and transmission subject to a fixed end-to-end source distortion. It is shown that the best coding and transmission strategy depends on the channel state.

Simunic studies energy-efficient hardware and software design [67] on a WLAN card used on a Linux laptop. A semi-markov decision model is used to model the traversing of a device between “sleep”, “idle”, and “transmit” states.

Another study that evaluates the actual total energy consumption of a wireless local area network interface experimentally is [29]. A set of measurements of power taken during several operational modes of and IEEE 802.11 node operating in an ad-hoc networking environment. Some implications on protocol design are discussed. In fact, energy-aware design of network protocols has drawn much attention recently (*e.g.*, [77, 16, 61]). Although there is a large amount of research, strong results are rare, and seem hard to obtain unless basic components of the problem are modeled and understood well. In the following chapters, we present an attempt at doing that.

2.5 Summary

We have reviewed power control research and have specifically isolated three groups of studies in this area. Studies in the first group are characterized by the use of power control to optimize the number of users that can be accomodated in a wireless network, while satisfying a service quality criterion such as a certain signal to noise ratio, or a certain information rate. Interferers are treated as noise, and the main limitation is interference, rather than fading. Studies in the second group have a predominantly information-theoretic approach. A generic problem is to find the power control policy that achieves the ergodic capacity in the fading single or multi-user channel, under an average power constraint. While the first group's operational target is the medium-access or network layers, the second group works with physical or link-layer techniques.

Thirdly, we looked at the more recent efforts at combined network-theoretic and information-theoretic treatments of wireless communication. Power control mechanisms that result from such approaches are inter-layer in nature. We believe that such inter-layer algorithms are needed to achieve the potential of wireless networks. The work in the following chapters is an effort of this nature.

Chapter 3

Minimum Energy Packet Scheduling

3.1 Introduction

As pointed out in Chapters 1 and 2, previous treatments of power control have usually ignored the tradeoff between energy and delay. Working with this tradeoff, though, is key to energy-efficient transmission of packetized data which can be bursty, or generated at a rate that is variable or unknown. There is then a need for a novel problem formulation that captures the energy/delay tradeoff in a way that promotes analysis as well as suggesting practical algorithms. The goal of this chapter is to develop such a formulation: specifically, one of minimizing the energy used by a node to transmit packetized information on a point-to-point link within a given amount of time. The formulation and results in this chapter appeared previously in [65, 71].

The outline of the point-to-point (Fig. 3.1) problem formulation is the following: data

Figure 3.1: Single link model

packets arrive at the transmitter's buffer at arbitrary times, during a certain time window starting from $t = 0$. The goal is to transmit all these packets to the receiver by time $t = T$, while minimizing total transmission energy. Before making the model more precise (in Section 3.2) it is worthwhile to discuss the function of the deadline T. Suppose T is finite. In this case, the setup models a number of realistic wireless networking scenarios: (i) A node with finite lifetime and finite energy supply such as in a sensor network [64]. (ii) A battery operated node with finite-lifetime information; that is, information that must be transmitted before a deadline. (iii) A battery operated node that is periodically recharged. In this case minimizing transmission energy ensures that the node does not run out of energy before it is recharged.

While the deadline constraint models a number of realistic scenarios that can occur in a wireless network, it also has theoretical significance. For one thing, the constraint provides an immediate bound on delay that is particularly suited for offline analysis of the problem. When the deadline constraint is replaced with a constraint on average delay, or an objective to minimize average delay, the analysis necessiates knowledge of packet arrival statistics. Even with knowledge of arrival statistics, such analysis is rather complicated and does not lead to closed-form answers. Dynamic programming methods have been applied ([22, 12]).

We vary packet transmission times and power levels to find the optimal, *i.e.*, minimum-energy, schedule for transmitting the packets within the given amount of time. The observation that leads to this approach is that transmission energy can be lowered by reducing transmission power and transmitting a packet over a longer period of time. It has been known (see [2], and more recently, [35]) that with many coding schemes, the energy needed to transmit a given amount of information is strictly decreasing and convex in the transmission duration. In the next section, several examples in support of this observation will be provided.

The above discussion implies that it makes sense to transmit a packet over a longer period of time to conserve energy. However, since all packets must be transmitted within the given amount of time, the transmission time of any one packet cannot be arbitrarily long as this would leave too little a time for the transmission of future packets and increase the overall energy spent. In the following sections this trade-off is precisely explored and used to devise energy-efficient schedules.

The outline of the rest of this chapter is as follows: Section 3.2 sets up the minimum energy packet transmission scheduling problem. In Section 3.3, the offline optimal-energy transmission schedule for fixed length packets is found, and extended to variable length packets in Section 3.4.

The form of the offline optimal-energy schedule (abbreviated to OOE) suggests a natural online schedule. In Section 3.5.4, by letting $T \to \infty$ and assuming Poisson arrivals, we are able to conduct an exact analysis of the optimal offline schedule. This gives us insight into how to design an energy-efficient online schedule that assigns transmission times according to the backlog in the queue. We call this schedule *Lazy*. Under a queue stability constraint, Lazy is compared with a constant schedule, *i.e.*, a schedule that assigns constant transmission times to packets, and it is shown to beat the constant schedule significantly for a range of packet arrival rates. This is an interesting comparison because among schedules that are independent of the packet arrival process (and hence are oblivious of backlogs), the constant schedule achieves the smallest average delay[1], which implies that it has the highest transmission times, and hence the lowest energy. The fact that lazy schedules are more energy-efficient than the constant schedule, therefore, demonstrates the need to take advantage of lulls in packet arrival times.

3.2 Problem Formulation

Consider a wireless node which contains a buffer and a transmitter, as illustrated in Figure 3.1. Assume that M packets arrive at the node in the time interval $[0, T)$ and must be transmitted to a receiver before T (see Figure 3.2). In the figure, the arrival times of

[1]By the well-known theorem "determinism minimizes delay" [76].

packets, t_i, are marked by crosses and inter-arrival epochs are denoted by d_i. Without loss of generality, we assume that the first packet is received at time 0. The node tranmits the packets according to a schedule that determines the beginning and the duration of each packet transmission. We seek an answer to the question: How should the transmission schedule be chosen so that the total *energy* used to transmit the packets is minimized?

Figure 3.2: Packet arrivals in $[0, T)$

Let $\mathcal{E}(q)$ denote the transmission energy per bit for the particular coding scheme that is being used, which has code rate $R = \frac{1}{q}$ bits/transmission [2]. Hence q is the number of transmissions per bit. The following are the only assumptions we make about $\mathcal{E}(q)$ in this chapter:

1. $\mathcal{E}(q) \geq 0$.
2. $\mathcal{E}(q)$ is monotonically decreasing in q.
3. $\mathcal{E}(q)$ is strictly convex in q.

Assumption (1) is obvious. We shall now demonstrate, for two examples on the discrete-time Additive White Gaussian Noise (AWGN) channel, that assumptions (2) and (3) hold.

Example 1. *Optimal channel coding:* Consider an AWGN channel with average signal power constraint P and noise power N. The information-theoretically optimal channel coding scheme, which employs randomly generated codes [23], achieves the channel capacity given by

$$C_1 = \frac{1}{2}\log_2\left(1 + \frac{P}{N}\right) \text{ bits/transmission.} \tag{3.1}$$

More precisely, given any $0 < \alpha < 1$ information can be reliably transmitted at rate $R = \alpha C_1$. To determine the energy per bit $\mathcal{E}(q)$, note that $q = \frac{1}{R}$ can be interpreted as the

[2] The word *transmission* in this thesis frequently refers to the transmission of an entire packet. The term *bits/transmission* will be used to indicate the number of bits per channel use (also known as *bits/symbol*), *i.e.*, the information theoretic rate, and *transmissions/bit* indicates the reciprocal of this rate.

Figure 3.3: Energy per bit vs. transmission time with optimal coding

Figure 3.4: Energy per bit vs. transmission time for the suboptimal coding scheme

number of transmissions per bit. Substituting in equation (3.1), we get

$$\mathcal{E}(q) = qP = qN(2^{\frac{2}{\alpha q}} - 1). \tag{3.2}$$

It is easy to see that $\mathcal{E}(q)$ is *monotonically decreasing and convex* in q, and that as q approaches infinity the energy required to transmit a bit, $\mathcal{E}_\infty = \frac{2}{\alpha}\ln 2 \approx \frac{1}{\alpha}1.3863$. Figure 3.3 plots $\mathcal{E}(q)$ vs. q for $N = 1$ and $\alpha = 0.99$. The range of q in the plot corresponds to SNR values from 20dB down to 0.11dB. This is a fairly typical range of SNR values for a wireless link [56]. In this range $\mathcal{E}(q)$ can be decreased by a factor of 20 by increasing transmission time and correspondingly decreasing power.

Example 2. *A suboptimal channel coding scheme:* Consider a scheme that uses antipodal signaling [66] and binary block error correction coding again over an AWGN wireless link. It can be shown that the minimum error probability per bit using antipodal signaling over an AWGN channel is given by

$$p = Q\left(\sqrt{\frac{P}{N}}\right),$$

where Q is the well known Gaussian Q-function. Using this signaling scheme the channel is converted into a binary symmetric channel (BSC) with cross-over probability p. The optimal binary error correction coding scheme achieves the Shannon capacity for the BSC, given by

$$C_2 = 1 - h(p) \text{ bits/transmission},$$

where $h(p)$ is the binary entropy function.

Thus for any $0 < \alpha < 1$, information can be reliably transmitted at rate $R = \alpha C_2$.

Figure 3.5: Energy per bit vs. transmission time with uncoded MQAM modulation

Again interpreting $q = \frac{1}{R}$ to be the number of transmissions per bit, the energy per bit can be computed as a function of q. This is depicted in Figure 3.4 for $N = 1$ and $\alpha = 0.99$. Note that $\mathcal{E}(q)$ is again *monotonically decreasing and convex* in q, and converges to a limit $\mathcal{E}_\infty = 2.108$, which, as expected, is larger than that using optimal coding. The range of q in the figure corresponds to SNR between 20dB to -3.7dB.

Example 3.*An Uncoded MQAM scheme:* Here, we suppose that every symbol has $M = 2^r$ possible values, hence one symbol carries r bits of information, *i.e.*, the number of transmissions per bit is $\frac{1}{r}$. This modulation scheme is used in some practical wireless systems, *e.g.*, the IEEE 802.11a wireless LAN standard recommends MQAM with $r = 1, 2, 4, 6$ in each OFDM subcarrier.

Figure 3.5 plots the energy per bit as a function of the number of transmissions per bit using MQAM, when the bit error rate is less than 10^{-4}.

The three examples above support the assumptions made earlier about $\mathcal{E}(q)$. Now, let us denote by $w(\tau)$ the transmission energy for a packet that takes τ transmissions (*i.e.*, channel uses). If the packet contains B bits, this corresponds to $q = \frac{\tau}{B}$ transmissions/bit, and $w(\tau) = B\,\mathcal{E}(\frac{\tau}{B})$. From our assumptions about $\mathcal{E}(q)$, it follows that $w(\tau)$ is a nonnegative, monotonically decreasing and convex function of τ.

3.3 Optimal Offline Scheduling

In this section we determine the energy-optimal offline schedule for the above model of a finite number of packets to be transmitted in a given finite time horizon. After briefly introducing the basic setup, a necessary condition for optimality is stated (Lemma 2). This motivates the definition of the specific schedule OOE (Definition 1). OOE is shown to be feasible (Lemma 3), and energy-optimal (Theorem 1).

Suppose that the arrival times t_i, $i = 1, \ldots, M$ of the M packets that arrive in the interval $[0, T)$ are known in advance, *i.e.*, before $t = 0$. Assuming equal length packets each with B bits, the offline scheduling problem is to determine the transmision duration vector $\vec{\tau}$ so as to minimize $w(\vec{\tau}) = \sum_{i=1}^{M} w(\tau_i)$.

The assumption that $w(\tau)$ decreases with τ trivially implies that it is sub-optimal to have $\sum_i \tau_i < T$. For, we could simply increase the transmission times of one or more packets and reduce $w(\vec{\tau})$. Hence we only consider "non-idling" transmission schedules where $\sum_i \tau_i = T$. It is also sufficient to consider FIFO schedules where packets are transmited in the order they arrive. The FIFO and non-idling conditions combined with the causality constraint, *i.e.*, that packet transmission cannot begin before the packet arrives, yield the following feasibility conditions.

Lemma 1 *A non-idling FIFO schedule $\vec{\tau}$ is feasible iff*

$$\sum_{i=1}^{k} \tau_i \geq \sum_{i=1}^{k} d_i$$

for $k = 1, 2 \ldots, M-1$, and $\sum_{i=1}^{M} \tau_i = \sum_{i=1}^{M} d_i$.

We now state a key observation of this section:

Lemma 2 *A necessary condition for optimality is*

$$\tau_i \geq \tau_{i+1} \quad \textit{for} \quad i \in \{1, \ldots, M-1\}. \tag{3.3}$$

Proof. Let $\vec{\tau}$ be a feasible vector such that $\tau_i < \tau_{i+1}$ for some $i \in \{1, \ldots, M-1\}$. Further suppose that it is optimal. Consider the schedule $\vec{\sigma}$ such that $\sigma_i = \sigma_{i+1} = \frac{\tau_i + \tau_{i+1}}{2}$ and $\sigma_j = \tau_j$ for $j \neq i, i+1$. It is easy to verify that $\vec{\sigma}$ is feasible. Comparing the energies used by $\vec{\tau}$ and $\vec{\sigma}$ we obtain

$$\begin{aligned} w(\vec{\tau}) - w(\vec{\sigma}) &= w(\tau_i) + w(\tau_{i+1}) \\ &\quad -w(\sigma_i) - w(\sigma_{i+1}) \\ &= w(\tau_i) + w(\tau_{i+1}) \\ &\quad -2w(\frac{\tau_i + \tau_{i+1}}{2}) \\ &\stackrel{(a)}{>} 0, \end{aligned}$$

where inequality (a) follows from the strict convexity of $w(\cdot)$. This contradicts the optimality of $\vec{\tau}$ and proves the lemma. ■

The proof of the above lemma suggests the form of the optimal offline schedule: Equate the transmission times of each packet, subject to feasibility constraints. We proceed to do just this and define the optimal schedule next.

Given packet inter-arrival times $d_i, i \in \{1, \ldots, M\}$, let $k_0 = 0$, and define

$$\begin{aligned} m_1 &= \max_{k \in \{1,\ldots,M\}} \{\frac{1}{k}\sum_{i=1}^{k} d_i\} \text{ and} \\ k_1 &= \max\{k : \frac{1}{k}\sum_{i=1}^{k} d_i = m_1\}. \end{aligned}$$

For $j \geq 1$, let

$$\begin{aligned} m_{j+1} &= \max_{k \in \{1,\ldots,M-k_j\}} \{\frac{1}{k}\sum_{i=1}^{k} d_{k_j+i}\} \text{ and} \\ k_{j+1} &= k_j + \max\left\{k : \frac{\sum_{i=1}^{k} d_{k_j+i}}{k} = m_{j+1}\right\}, \end{aligned}$$

where k varies between 1 and $M - k_j$. We proceed as above to obtain pairs (m_j, k_j) until $k_j = M$ for the first time.[3] Let $J = \min\{j : k_j = M\}$. The pairs $(m_j, k_j), j = 1, \ldots, J$

[3] Note that, by definition, $k_j < k_{j+1}$. Therefore the k_j are increasing with j and will equal M for some j.

are used to define a schedule whose transmission times are denoted by $\vec{\tau}^*$, and Theorem 1 shows that $\vec{\tau}^*$ is the optimal offline schedule.

Definition 1 *The vector of transmission times $\vec{\tau}^*$ given by:*

$$\tau_i^* = m_j \quad if \quad k_{j-1} < i \leq k_j \tag{3.4}$$

is called OOE (for Offline Optimal-Energy).

Figure 3.6 shows an example run of OOE. The arrivals in the figure have been randomly generated (with exponentially distributed inter-arrival intervals of mean 1) using a time window of $T = 20$. The heights of the bars are proportional to the magnitudes of the d's and τ^*'s.

Figure 3.6: An example run of d's (top) and τ^*s (bottom)

Lemma 3 *The following hold for $\vec{\tau^*}$ of OOE:*
(i) It is feasible and $\sum_{i=1}^{M} \tau_i^* = T$.
(ii) It satisfies the condition stated in Lemma 2.

Proof. We first establish (i). For $1 \leq k \leq k_1$,

$$\sum_{i=1}^{k} \tau_i^* = k\, m_1 \geq k \sum_{i=1}^{k} \frac{d_i}{k} = \sum_{i=1}^{k} d_i,$$

where the inequality follows from the definition of m_1.

Similarly for $k_1 < k \leq k_2$,

$$\begin{aligned}
\sum_{i=1}^{k} \tau_i^* &= k_1\, m_1 + (k - k_1) m_2 \\
&\geq \sum_{i=1}^{k_1} d_i + (k - k_1) \sum_{i=k_1+1}^{k} \frac{d_i}{k - k_1} \\
&= \sum_{i=1}^{k} d_i.
\end{aligned}$$

Proceeding thus, we obtain that $\sum_{i=1}^{k} \tau_i^* \geq \sum_{i=1}^{k} d_i$ for all $k, 1 \leq k \leq M$.

To finish the proof of (i) it only remains to show that $\sum_{i=1}^{M} \tau_i^* = T$. Now

$$\sum_{i=1}^{M} \tau_i^* = \sum_{j=1}^{J} (k_j - k_{j-1}) m_j, \tag{3.5}$$

where $k_0 = 0$ and $k_J = M$. By definition of m_j and k_j, it follows that for each j

$$(k_j - k_{j-1}) m_j = \sum_{k=k_{j-1}+1}^{k_j} d_k.$$

Using this at equation (3.5), we get $\sum_{i=1}^{M} \tau_i^* = \sum_{i=1}^{M} d_i = T$. This establishes (i).

As for (ii), it suffices to show that $m_j > m_{j+1}$ since this implies $\tau_i^* \geq \tau_{i+1}^*$ for each i. We first show that $m_1 > m_2$. For any $k \in [k_1 + 1, k_2]$,

$$\begin{aligned}
m_1 &= \frac{d_1 + \ldots + d_{k_1}}{k_1} \\
&\overset{(a)}{>} \frac{d_1 + \ldots + d_{k_1}}{k} + \frac{d_{k_1+1} + \ldots + d_k}{k} \\
&= \frac{k_1}{k} m_1 + \frac{(k - k_1)}{k} \frac{(d_{k_1+1} + \ldots + d_k)}{k - k_1},
\end{aligned}$$

where (a) follows from the definition of m_1. Choosing $k = k_2$, we get

$$m_1 > \frac{k_1}{k_2} m_1 + \frac{k_2 - k_1}{k_2} m_2,$$

from which it follows that $m_1 > m_2$.

In a similar fashion it can be shown that $m_2 > m_3$ and, more generally, that $m_j > m_{j+1}$ for any $j, 1 \leq j \leq J - 1$. This establishes (ii) and completes the proof of the lemma. ∎

Theorem 1 *The schedule OOE of Definition 1 is the optimum offline schedule.*

Proof. Consider any other feasible schedule $\vec{\tau}$. Let i be the first index where $\tau_i \neq \tau_i^*$. We show that $w(\vec{\tau}) > w(\vec{\tau^*})$. There are two possibilities to consider.

Case 1: $\tau_i > \tau_i^*$. Since $\sum_j \tau_j = T$ (else, $\vec{\tau}$ would idle for some time, making it sub-optimal), there must be at least one $j > i$ for which $\tau_j < \tau_j^*$. Let $r = \min\{j : i < j \leq M, \tau_j < \tau_j^*\}$. Consider the schedule $\vec{\sigma}$ defined as follows:

$$\sigma_i = \tau_i - \Delta \quad (3.6)$$

$$\sigma_r = \tau_r + \Delta \quad (3.7)$$

$$\sigma_j = \tau_j \text{ for all } j \neq i, r \quad (3.8)$$

where $\Delta = \min\{(\tau_i - \tau_i^*), (\tau_r^* - \tau_r)\}$.

Claim 1: The schedule $\vec{\sigma}$ does not idle and is feasible.

Proof of Claim 1: Since $\sum_j \sigma_j = \sum_j \tau_j = T$ it does not idle. By the definition of the indices

i and r, and the feasibility of $\vec{\tau}$ and $\vec{\tau^*}$, it follows that

$$\begin{aligned}
\sum_{j=1}^{k} \sigma_j &= \sum_{j=1}^{k} \tau_j \geq \sum_{j=1}^{k} d_j \quad \text{for } 1 \leq k \leq i-1 && (3.9)\\
\sum_{j=1}^{i} \sigma_j &\geq \sum_{j=1}^{i} \tau_j^* \geq \sum_{j=1}^{i} d_j && (3.10)\\
\sum_{j=1}^{k} \sigma_j &\geq \sum_{j=1}^{k} \tau_j^* \geq \sum_{j=1}^{k} d_j \quad \text{for } i < k \leq r && (3.11)\\
\sum_{j=1}^{k} \sigma_j &= \sum_{j=1}^{k} \tau_j \geq \sum_{j=1}^{k} d_j \quad \text{for } k > r. && (3.12)
\end{aligned}$$

This verifies the conditions for feasibility in 1, and Claim 1 is proved.

Claim 2: $w(\vec{\sigma}) < w(\vec{\tau})$.
Proof of Claim 2:

$$\begin{aligned}
w(\vec{\tau}) - w(\vec{\sigma}) &= w(\tau_i) + w(\tau_r) \\
&\quad - w(\sigma_i) - w(\sigma_r) \\
&= w(\tau_i) - w(\tau_i - \Delta) \\
&\quad + w(\tau_r) - w(\tau_r + \Delta) \\
&\overset{(a)}{>} 0,
\end{aligned}$$

where inequality (a) follows from two facts: (i) $w(\cdot)$ is strictly convex and decreasing, and (ii) $\tau_i > \tau_r$. That is, for any real-valued function f that is strictly convex and decreasing, and for any $a, b \in \mathbb{R}$ such that $a < b$, we have $f(b) - f(b-\delta) + f(a) - f(a+\delta) > 0$, where $0 < \delta < b - a$. This proves Claim 2.

Thus, under Case 1, any feasible schedule $\vec{\tau}$ may be modified to obtain a more energy efficient schedule $\vec{\sigma}$. Therefore schedules which are different from $\vec{\tau^*}$ in the sense of Case 1 are sub-optimal.

Case 2: $\tau_i < \tau_i^*$. We shall argue for a contradiction and show that such a $\vec{\tau}$ is infeasible.

From the definition of $\vec{\tau^*}$ we know that $\tau_i^* = m_j$, assuming $k_{j-1} < i \leq k_j$. In fact $\tau_l^* = m_j$ for all $k_{j-1} < l \leq k_j$.

Since i is the first index where $\vec{\tau}$ and $\vec{\tau^*}$ disagree, $\tau_l = \tau_l^*$ for all $l < i$. Suppose that the schedule $\vec{\tau}$ satisfies the condition of Lemma 2 (else it is sub-optimal and we are done). It follows that $\tau_i \geq \ldots \geq \tau_{k_j}$, and we get

$$\sum_{l=1}^{k_j} \tau_l^* > \sum_{l=1}^{k_j} \tau_l. \tag{3.13}$$

But, by definition of $\vec{\tau^*}$,

$$\sum_{l=1}^{k_j} \tau_l^* = \sum_{l=1}^{j} (k_l - k_{l-1}), m_l = \sum_{l=1}^{k_j} d_l.$$

Equation (3.13) now gives $\sum_{l=1}^{k_j} \tau_l < \sum_{l=1}^{k_j} d_l$, implying that $\vec{\tau}$ is infeasible.

This contradiction concludes Case 2 and the proof of Theorem 1 is complete. ■

Lazy scheduling trades-off delay for energy. To do this it must necessarily buffer packets. The energy savings that come from simply keeping a small buffer is best illustrated by an example: Imagine a scheme that keeps a buffer size of zero (hence transmission times can at most be set equal to inter-arrival times). For the set of packet arrivals shown in Figure 3.6, the optimal offline schedule achieves an energy of 65.445 and the zero-buffer scheme (which, therefore, has no queuing delay) achieves an energy 77.78×10^5; five orders of magnitude larger (using an energy function $\tau(2^{\frac{6}{\tau}} - 1)$).

3.4 Extension to Optimal Offline Scheduling of Variable-Length Packets

This section extends the results of the previous section to variable-length packets. As the optimal schedule and the arguments that establish its optimality are virtually identical to those of the previous section, for brevity, we shall omit a number of details.

Consider a node at which M packets arrive in $[0, T]$, and relax the condition that packets are of equal lengths. The length of packet i equals l_i bits. Without loss of generality we consider schedules that do not idle. Hence, the feasibility condition in Lemma 1 continues to apply; *i.e.*, $\vec{\tau}$ is feasible if and only if for $1 \leq k < M$, $\sum_{i=1}^{k} \tau_i \geq \sum_{i=1}^{k} d_i$.

The arrival times $t_i, i = 1, \ldots, M$ are known at time 0, as are the lengths of the packets, $\vec{l} = [l_1, l_2, \ldots, l_M]$. As before, assume that $t_1 = 0$. Define $w(l, \tau) = l\,\mathcal{E}(\frac{\tau}{l})$. The problem is to determine $\vec{\tau}$, the vector of transmission times, so as to minimize $w(\vec{l}, \vec{\tau}) \triangleq \sum_{i=1}^{M} w(l_i, \tau_i)$.

Since, it is sub-optimal to consider idling policies, we shall only consider schedules $\vec{\tau}$ that satisfy $\sum_i \tau_i = T$.

Lemma 4 *A necessary condition for optimality is*

$$\frac{\tau_i}{l_i} \geq \frac{\tau_{i+1}}{l_{i+1}} \quad \textit{for} \quad i \in \{1, \ldots, M-1\}. \tag{3.14}$$

Proof. Let $\vec{\tau}$ be a feasible vector such that $\frac{\tau_i}{l_i} < \frac{\tau_{i+1}}{l_{i+1}}$ for some $i \in \{1, \ldots, M-1\}$. Further suppose that it is optimal. Consider the schedule $\vec{\sigma}$ such that $\frac{\sigma_i}{l_i} = \frac{\sigma_{i+1}}{l_{i+1}} = \frac{\tau_i + \tau_{i+1}}{l_i + l_{i+1}}$ and $\sigma_j = \tau_j$ for $j \neq i, i+1$. It is easy to verify that $\vec{\sigma}$ is feasible (because $\sigma_i > \tau_i$.)

Comparing the energies used by $\vec{\tau}$ and $\vec{\sigma}$ we obtain

$$\begin{aligned}
& w(\vec{l},\vec{\tau}) - w(\vec{l},\vec{\sigma}) \\
&= w(l_i,\tau_i) + w(l_{i+1},\tau_{i+1}) \\
&= l_i\,\mathcal{E}(\frac{\tau_i}{l_i}) + l_{i+1}\,\mathcal{E}(\frac{\tau_{i+1}}{l_{i+1}}) - l_i\,\mathcal{E}(\frac{\sigma_i}{l_i}) - l_{i+1}\,\mathcal{E}(\frac{\sigma_{i+1}}{l_{i+1}}) \\
&= l_i\,\mathcal{E}(\frac{\tau_i}{l_i}) + l_{i+1}\,\mathcal{E}(\frac{\tau_{i+1}}{l_{i+1}}) - (l_i + l_{i+1})\,\mathcal{E}(\frac{\tau_i + \tau_{i+1}}{l_i + l_{i+1}}) \\
&\stackrel{(a)}{>} 0,
\end{aligned}$$

where inequality (a) follows from the convexity of $\mathcal{E}(\cdot)$. This contradicts the optimality of $\vec{\tau}$ and the lemma is proved. ■

The proof of the above lemma suggests the principle of the optimal offline schedule: Equate the number of transmissions *per bit* for each packet, subject to feasibility constraints. Note that this principle is similar to the one in the previous section; indeed, as will be the optimal schedule and proofs.

Given packet inter-arrival times $d_i, i \in \{1,\ldots,M\}$, let $k_0 = 0$, and define

$$\begin{aligned}
\mu_1 &= \max_{k\in\{1,\ldots,M\}} \left\{ \frac{\sum_{i=1}^{k} d_i}{\sum_{i=1}^{k} l_i} \right\} \quad \text{and} \\
k_1 &= \max \left\{ k : \frac{\sum_{i=1}^{k} d_i}{\sum_{i=1}^{k} l_i} = \mu_1 \right\}.
\end{aligned}$$

For $j \geq 1$, let

$$\begin{aligned}
\mu_{j+1} &= \max_{k\in\{1,\ldots,M-k_j-1\}} \left\{ \frac{\sum_{i=1}^{k} d_{k_j+i}}{\sum_{i=1}^{k} l_{k_j+i}} \right\} \quad \text{and} \\
k_{j+1} &= k_j + \max \left\{ k : \frac{\sum_{i=1}^{k} d_{k_j+i}}{\sum_{i=1}^{k} l_{k_j+i}} = \mu_{j+1} \right\},
\end{aligned}$$

where k varies between 1 and $M - k_j$. We proceed as above to obtain pairs (μ_j, k_j) until $k_j = M$ for the first time. Let $J = \min\{j : k_j = M\}$. The pairs $(\mu_j, k_j), j = 1,\ldots,J$ are used to define the general form of OOE (the OOE of the previous section is simply the special case for which $l_i = B, \forall\, i$). As in the previous section, transmission times of OOE are denoted $\vec{\tau}^*$, and Theorem 2 shows that $\vec{\tau}^*$ is the optimal offline schedule for the variable length case.

Definition 2 OOE. *The schedule with the vector of transmission times $\vec{\tau}^*$ given by:*

$$\tau_i^* = l_i\, \mu_j \quad \textit{if} \quad k_{j-1} < i \leq k_j \tag{3.15}$$

is called OOE.

Lemma 5 *The following hold for the $\vec{\tau^*}$ of OOE:*
(i) It is feasible and $\sum_{i=1}^{M} \tau_i^* = T$.
(ii) It satisfies the condition stated in Lemma 4.

Proof. We first establish (i). For $1 \leq k \leq k_1$,

$$\sum_{i=1}^{k} \tau_i^* = \sum_{i=1}^{k} l_i\, \mu_1 \geq \sum_{i=1}^{k} l_i \frac{\sum_{j=1}^{k} d_j}{\sum_{j=1}^{k} l_j} = \sum_{i=1}^{k} d_i,$$

where the inequality follows from the definition of μ_1.

Similarly for $k_1 < k \leq k_2$,

$$\begin{aligned}\sum_{i=1}^{k} \tau_i^* &= \sum_{i=1}^{k_1} l_i\, \mu_1 + \sum_{i=k_1+1}^{k} l_i\, \mu_2 \\ &\geq \sum_{i=1}^{k} d_i.\end{aligned}$$

Proceeding thus, we obtain that $\sum_{i=1}^{k} \tau_i^* \geq \sum_{i=1}^{k} d_i$ for all $k, 1 \leq k \leq M$. By similar steps, it can be shown that $\sum_{i=1}^{M} \tau_i^* = T$, and (i) is established.

As for (ii), it suffices to show that $\mu_j > \mu_{j+1}$ since this implies $\frac{\tau_i^*}{l_i} \geq \frac{\tau_{i+1}^*}{l_{i+1}}$, for each i.

We first show that $\mu_1 > \mu_2$. For any $k \in [k_1 + 1, k_2]$,

$$\begin{aligned}
\mu_1 &= \frac{d_1 + \ldots + d_{k_1}}{l_1 + \ldots + l_{k_1}} \\
&\overset{(a)}{>} \frac{\sum_{i=1}^{k_1} d_i + \sum_{i=k_1+1}^{k} d_i}{\sum_{i=1}^{k} l_i} \\
&= \frac{\sum_{i=1}^{k_1} d_i}{\sum_{i=1}^{k} l_i} + \frac{\sum_{i=k_1+1}^{k} d_i}{\sum_{i=1}^{k} l_i},
\end{aligned}$$

where (a) follows from the definition of μ_1. Choosing $k = k_2$, we get

$$\mu_1 > \frac{\sum_{i=1}^{k_1} d_i}{\sum_{i=1}^{k} l_i}\mu_1 + \frac{\sum_{i=k_1+1}^{k} d_i}{\sum_{i=1}^{k} l_i}\mu_2,$$

from which it follows that $\mu_1 > \mu_2$.

In a similar way it can be shown that $\mu_2 > \mu_3$ and, more generally, that $\mu_j > \mu_{j+1}$ for any $j, 1 \leq j \leq J - 1$. This establishes (ii) and completes the proof of the lemma. ■

Theorem 2 *The schedule OOE of Definition 2 is the optimum offline schedule.*

Proof. The proof is identical to the proof of Theorem 1. Hence, to avoid repetitions, we only present the highlights and not the details.

As before, consider any other feasible schedule $\vec{\tau}$. Let i be the first index where $\tau_i \neq \tau_i^*$. We show that $w(\vec{l}, \vec{\tau}) > w(\vec{l}, \vec{\tau^*})$. There are the following two possibilities to consider: Case 1. $\tau_i > \tau_i^*$, and Case 2. $\tau_i < \tau_i^*$.

Under Case 1, we use the schedule $\vec{\tau}$ to define another schedule $\vec{\sigma}$ as before and establish the following two claims.

Claim 1: The schedule $\vec{\sigma}$ does not idle and is feasible.
Claim 2: $w(\vec{l}, \vec{\sigma}) < w(\vec{l}, \vec{\tau})$.

Hence we conclude that any feasible schedule $\vec{\tau}$ differing from $\vec{\tau^*}$ in the sense of Case 1 may be modified to obtain a strictly more energy efficient schedule $\vec{\sigma}$. This concludes Case

1.

Under Case 2, we shall argue for a contradiction and show that the schedule $\vec{\tau}$ must be infeasible exactly as in the proof of Theorem 1.

This completes the proof of Theorem 2. ■

3.5 Online scheduling

In this section we develop and evaluate energy efficient online scheduling algorithms based on the optimal offline algorithm discussed in Section 3.3. Henceforth, we shall assume that packets are of the same length.

In order to design online algorithms that are energy efficient on *average*, one needs the statistics of the arrival process. While our approach is general, for concreteness and tractability, we assume Poisson arrivals for the analysis conducted in this section. Note that Poisson arrivals may be unrealistic in a practical setting, such as a wireless LAN environment, where arrivals tend to be more bursty. However, as we shall observe later, when arrivals are bursty, lazy scheduling performs even better than in the Poisson case; for, one can take advantage of a small queuing delay and greatly reduce transmission energy.

We proceed by first formulating the offline algorithm OOE in a manner that is suited for online use (Section 3.5.1). Based on this formulation we propose an online algorithm (Section 3.5.2) and, using simulations, show that on the average it is almost as energy efficient as the optimal offline schedule (Section 3.5.3).

We then investigate the important special case of $T \to \infty$. In this case we are able to analyze the optimal offline schedule exactly, obtain an online lazy schedule as a result of this analysis, and perform comparisons of the energy efficiency of the lazy schedule and a fixed-transmission time online algorithm (Section 3.5.4).

3.5.1 Online formulation of OOE

Consider the time interval $[0, T)$ and as before assume that a packet arrives at time 0. Suppose also that packets arrive as a Poisson process of rate λ. Conditioned on there being $M-1$ arrivals in $(0, T)$, let the inter-arrival times be denoted by D_i. Let the optimal offline schedule, OOE, assign transmission times $\vec{\tau^*}$ to these M packets. The time at which the j^{th} packet starts transmitting is

$$T_j^* = \sum_{i=1}^{j-1} \tau_i^*.$$

The quantity b_j, given by

$$b_j = \max\{k : \sum_{i=1}^{k-1} D_i < T_j^*\} - j,$$

is the *backlog* in the queue when the j^{th} packet starts transmitting. Observe that this backlog does not include the j^{th} packet; that is, if $b_j = 1$, then there is precisely one packet (namely, the $(j+1)^{th}$) in the queue when the j^{th} packet starts transmitting. Finally, let $C_i, i \in \{1, \ldots, M-j-b_j\}$ be the inter-arrival times between packets that arrive *after* T_j^*. Thus, when the j^{th} packet starts transmitting the situation is this: (i) The "time to go" equals $T - T_j^*$, (ii) there are b_j packets currently backlogged, (iii) $M - j - b_j$ packets are yet to arrive and the first of these will arrive in C_1 units of time, the second will arrive in $C_1 + C_2$ units of time, etc.

With this notation and some algebra, it can be shown that τ_j^* is also given by

$$\tau_j^* = \max_{k \in \{1, \ldots, M-(j+b_j)\}} \{\frac{1}{k+b_j} \sum_{i=i}^{k} C_i\}. \tag{3.16}$$

This formula is just an alternative representation of OOE, and gives exactly the same schedule. It schedules packets one at a time, taking into account the current backlog, future arrivals, and the time to go.

Figure 3.7: A comparison of the online algorithm with the optimal offline algorithm

3.5.2 Online algorithm

The alternative form of OOE in Equation 3.16 strongly suggests the following heuristic online algorithm: The transmission time of a packet that starts being transmitted at time $t < T$ when there is a backlog of b packets is set equal to the *expected value* of the random variable

$$\tau^*(b,t) = \max_{k \in \{1,\ldots,M\}} \left\{ \frac{1}{k+b} \sum_{i=1}^{k} D_i \right\}, \tag{3.17}$$

where b is the current backlog, D_i are the inter-arrival times of the M packets that will arrive in (t, T). Note that M is a random number now.

In the following section, we evaluate $\mathrm{E}(\tau^*(b,t))$ numerically when T is finite, and then make comparisons of online schedules based on $\mathrm{E}(\tau^*(b,t))$ with the optimal offline schedule.

3.5.3 Simulations: Finite time horizon

Using simulations we compare the energies expended by the online algorithm defined above and the optimal offline algorithm. The setup is as follows. A finite time horizon $T = 10$ sec is chosen. We assume a packet length of $B = 10$ KBits and a maximum rate of 6 bits/transmission, with a link speed of 10^6 transmissions/sec. (Hence, the minimum transmission duration for a packet is $\frac{10}{6}$ msec, which we shall call a *time unit*). Within the time period T, we assume that packets arrive according to a Poisson process at a loading factor of $\lambda = 0.7$ arrivals per time unit. Since it is possible for packets to arrive arbitrarily close to the finish time T, if we insist that these very late arrivals also be transmitted before the deadline T, then *any* algorithm, including the optimal offline algorithm, incurs a huge energy cost. This makes comparisons of performance difficult and meaningless. We therefore use a "guard band" g around the finish time and disallow packets from arriving after time $T - g$. For the comparison we use $g = .1$ sec.

The energy function we shall use is

$$w(\tau) = 10^6 \tau (2^{\frac{.02}{\tau}} - 1). \tag{3.18}$$

This is the packet transmission energy w as a function of packet transmission time τ in seconds, from Equation (3.2), which corresponds to transmitting at the information theoretic capacity in the AWGN channel with noise power $N = 1$. The packets are of length $B = 10Kbit$ and the symbol rate is 10^6 transmissions/sec.

In Figure 3.7, the energy per packet achieved by the online algorithm is plotted for different values of the loading factor λ. The offline optimal energy per packet is also given on the same plot. The closeness between the two curves indicates that this online heuristic is almost as energy-efficient as the optimal offline algorithm.

3.5.4 Infinite time horizon: Formulation and simulations

The online algorithm we explored above was directly motivated by the optimal offline algorithm, and was seen experimentally to perform close to optimal. It will be quite interesting to take the deadline $T \to \infty$ and convert this algorithm to one that does not depend on the amount of time left until a deadline. Let us define $\mathrm{E}(\tau^*(b)) \triangleq \mathrm{E}(\lim_{t\to\infty} \tau^*(b,t))$. This has an interesting explicit form in the Poisson case.

Theorem 3 *Under Poisson arrivals with rate λ, $E(\tau^*(b)) = \frac{(1+b)}{\lambda}(\frac{\pi^2}{6} - \sum_{k=1}^{b} \frac{1}{k^2})$.*

Proof. Consider a transmitter which, at time 0, has b packets in the queue. Suppose that M packets arrive at this node in $[0, T)$, with the first of these arriving at time 0. This situation can be modeled as $M + b$ packets arriving in $[0, T)$ with $d_1 = \cdots = d_b = 0$ and $\sum_{j=1}^{M+b} d_j = T$. Then, cf. Equation 3.17, the optimal offline schedule will transmit the first packet for an amount of time, say $\tau_M(b)$, which is given by

$$\tau_M(b) = \max_{i \in \{1,\ldots,M+b\}} \{\frac{1}{i} \sum_{j=1}^{i} d_j\} \tag{3.19}$$

Here we analyze the optimal offline schedule by allowing T to approach infinity. Thus suppose that the arrivals in $[0,T)$ occur as a rate λ Poisson process and let T go to infinity to get

$$\tau(b) = \sup_{\{i \geq 1\}} \{\frac{1}{i+b} \sum_{j=1}^{i} D_j\}, \tag{3.20}$$

where the D_j are i.i.d. mean $1/\lambda$ exponential random variables. Define $S_i = \sum_{j=1}^{i} D_j$, and $\tau_n(b) = \max_{\{1 \leq i \leq n\}}\{\frac{S_i}{i+b}\}$.

Lemma 6

$$E(\tau_n(b)) = \frac{1+b}{\lambda} \sum_{k=1}^{n} \frac{1}{(k+b)^2} \tag{3.21}$$

Proof. We start by expressing the distribution function of $\tau_n(b)$:

$$\begin{aligned}
&\Pr(\tau_n(b) < t) \\
&= \Pr(\max_{\{1 \leq i \leq n\}} \{\frac{S_i}{i+b}\} < t) && (3.22)\\
&= \Pr(\frac{S_i}{i+b} < t, \;\; \forall \;\; i: \; 1 \leq i \leq n) && (3.23)\\
&= \int_0^{t(1+b)} \int_{s_1}^{t(2+b)} \ldots && (3.24)\\
&\quad \int_{s_{n-1}}^{t(n+b)} f_{S_1,\ldots S_n}(s_1,\ldots s_n) ds_n \ldots ds_1 && (3.25)
\end{aligned}$$

Note that $f_{D_1,\ldots,D_n}(d_1,\ldots,d_n) = \lambda^n exp(-\lambda \sum_{i=1}^{n} d_i)$ (recall the independence assumption.) The Jacobian of the transformation $S_i = \sum_{j=1}^{i} D_i, \;\; \forall \;\; i$ is 1.

Hence, $f_{S_1,S_2,\ldots,S_n}(s_1, s_2, \ldots, s_n) = \lambda^n exp(-\lambda s_n)$, and equation (3.25) can be written as:

$$
\begin{aligned}
&\Pr(\tau_n(b) < t) \\
&= \int_0^{t(1+b)} \int_{s_1}^{t(2+b)} \cdots \\
&\quad \int_{s_{n-1}}^{t(n+b)} \lambda^n \exp(-\lambda s_n) ds_n \ldots ds_1 \qquad (3.26) \\
&= \lambda^{n-1} \int_0^{t(1+b)} \int_{s_1}^{t(2+b)} \cdots \\
&\quad \int_{s_{n-2}}^{t(n-1+b)} e^{-\lambda s_{n-1}} - e^{-\lambda t(n+b)} ds_{n-1} \ldots ds_1 \\
&= \Pr(\tau_{n-1}(b) < t) - \lambda^{n-1} \int_0^{t(1+b)} \cdots \\
&\quad \int_{s_{n-2}}^{t(n-1+b)} e^{-\lambda t(n+b)} ds_{n-1} \ldots ds_1 \qquad (3.27)
\end{aligned}
$$

Using the identity $\mathrm{E}(Y) = \int_0^\infty \Pr(Y > t)dt$ for any positive random variable Y, we obtain from equation (3.27):

$$
\begin{aligned}
\mathrm{E}(\tau_n(b)) = \mathrm{E}(\tau_{n-1}(b)) + \\
\lambda^{n-1} \int_0^\infty t^{n-1} e^{-\lambda t(n+b)} \int_0^{(1+b)} \cdots \\
\int_{u_{n-2}}^{(n-1+b)} du_{n-1} \ldots du_1
\end{aligned}
$$

by the normalization $u_i = s_i/t$. The $(n-1)$-dimensional volume

$$
\int_0^{(1+b)} \int_{u_1}^{(2+b)} \cdots \int_{u_{n-2}}^{(n-1+b)} du_{n-1} \ldots du_1
$$

can be shown (by an induction argument) to equal

$$
(1+b)\frac{(n+b)^{(n-2)}}{(n-1)!}.
$$

Substituting this into the above equation and integrating with respect to t,

$$\mathrm{E}(\tau_n(b)) = \mathrm{E}(\tau_{n-1}(b)) + \frac{(1+b)}{\lambda(n+b)^2} \tag{3.28}$$

Since $\mathrm{E}(\tau_1(b)) = \frac{1}{\lambda(1+b)}$, Lemma 6 follows. ■

Corollary 1 *Define $\tau(b) = \sup_{k\geq 1} \frac{S_i}{i}$, and recall the definition $E(\tau(b)) \triangleq E(\lim_{n\to\infty} \tau_n(b))$. Then,*

$$E(\tau(b)) = \frac{(1+b)}{\lambda}(\frac{\pi^2}{6} - \sum_{m=1}^{b} \frac{1}{m^2}) \tag{3.29}$$

Proof: Follows from monotone convergence. ■

Side Results about Exponential Random Variables

We shall now digress to note some corollaries of the above analysis. These results about the running averages of i.i.d. exponential random variables were, to our knowledge, first reported in [71].

Corollary 2 *Define $Z_n = \max_{\{1\leq i\leq n\}} \frac{S_i}{i}$, and $I_n = Z_n - Z_{n-1}$. The following hold:*

(1) $E(I_n) = \frac{1}{\lambda n^2}$.

(2) $E(Z_n) = \frac{1}{\lambda}\sum_{i=1}^{n} \frac{1}{i^2}$.

(3) $Pr(\frac{S_n}{n} > Z_{n-1}) = \frac{1}{n}$.

(4) $E(\frac{S_n}{n} - Z_{n-1} | \frac{S_n}{n} > Z_{n-1}) = \frac{1}{\lambda n}$.

(5) $\sup_{\{n\geq 1\}} Z_n = \lim_{\{n\geq 1\}} Z_n = \frac{\pi^2}{6\lambda}$

Proof. Part (1) follows by taking $b = 0$ in equation (3.28). Part (2) follows by setting $b = 0$ in Lemma 6.

We now show parts (3)-(5). For notational convenience, we set $\lambda = 1$ for the time being; the results trivially scale by $\frac{1}{\lambda}$, as will be clear in the calculations below.

To establish part (3), we write

$$\begin{aligned}
&\Pr\left(\frac{S_n}{n} > Z_{n-1}\right) \\
&= \Pr\left(\frac{S_n}{n} > \frac{S_i}{i}, \ \forall \ 1 \le i < n\right) \\
&= \Pr\left(\frac{S_n}{n} > S_1, \ldots, \frac{S_n}{n} > \frac{S_{n-1}}{n-1}\right) \\
&= \int_{z=0}^{\infty} A_{n-1}(z) f_{S_n}(nz) n dz, \qquad (3.30)
\end{aligned}$$

where $A_{n-1}(z) \triangleq \Pr(S_1 < z, S_2 < 2z, \ldots, S_{n-1} < (n-1)z | S_n = nz)$. Recall that S_i are arrival epochs in a Poisson process. The condition $S_n = nz$ is the same as saying that $n-1$ arrivals occurred in $[0, nz)$, and it is well known that under this condition $S_1, \ldots, S_{n-1}$ are distributed as order statistics ([36]), *i.e.*

$$f_{(S_1,\ldots,S_{n-1}|S_n)}(s_1, \ldots, s_{n-1} | s_n = nz) = \frac{(n-1)!}{(nz)^{n-1}}. \qquad (3.31)$$

Therefore,

$$\begin{aligned}
A_{n-1}(z) &= \int_0^z \cdots \int_{s_{n-2}}^{(n-1)z} \frac{(n-1)!}{(nz)^{n-1}} d_{s_{n-1}} \ldots d_{s_1} \\
&= n^{-(n-1)}(n-1)! \int_0^1 \cdots \\
&\quad \int_{u_{n-2}}^{(n-1)} d_{u_{n-1}} \ldots d_{u_1} \\
&= n^{-(n-1)}(n-1)! \frac{n^{(n-2)}}{(n-1)!} \\
&= \frac{1}{n} \qquad (3.32)
\end{aligned}$$

Substituting equation (3.32) into equation (3.30), we obtain $\Pr(\frac{S_n}{n} > \max_{1\le i<n} \frac{S_i}{i}) = \frac{1}{n}$.

Figure 3.8: A plot of $\mathrm{E}(\tau(b))$ vs. b for $\lambda = 1$

For part (4), write $\mathrm{E}(I_n) = \mathrm{E}(I_n | I_n > 0)\Pr(I_n > 0)$. Or, more explicitly, $\mathrm{E}(I_n) = \mathrm{E}(\frac{S_n}{n} - Z_{n-1} | \frac{S_n}{n} > Z_{n-1})\Pr(\frac{S_n}{n} > Z_{n-1})$. From part (3), $\Pr(\frac{S_n}{n} > Z_{n-1}) = \frac{1}{n}$, and from part (1), $\mathrm{E}(I_n) = \frac{1}{\lambda n^2}$, so $\mathrm{E}(\frac{S_n}{n} - Z_{n-1} | \frac{S_n}{n} > Z_{n-1}) = \frac{1}{\lambda n}$. This result is interesting because it says that given the current time average exceeds the previous maximum, the average amount of the excess is exactly $\frac{1}{\lambda n}$. Finally, part (5) follows by setting $b = 0$ in Corollary 1. ■

The Lazy Schedule

In Figure 3.8, $\mathrm{E}(\tau^*(b))$ is plotted as a function of the backlog b when the arrivals are a rate 1 Poisson process. As intuitively expected, the average transmission time of the offline schedule decreases with the backlog, approaching $\frac{1}{\lambda}$ as the backlog, b, approaches infinity. This exact analysis of the offline algorithm not only provides us with insight into the manner in which transmission times should depend on backlog, but also suggests a natural online algorithm: Just before starting a packet transmission, sample the backlog, b, in the queue. Set the transmission duration of the packet to $\mathrm{E}(\tau^*(b))$.

The result is what we call a "lazy schedule". It is of course very interesting to compare this schedule to the offline optimal. Unlike the finite T case where schedules can be compared solely on the basis of their energy expenditure, when $T = \infty$ packet delays (or queue size, stability, etc.) must be taken into consideration. Otherwise, energy comparisons become meaningless since we can simply let transmission times be arbitrarily long and obtain the minimum possible transmission energy per packet whereas the delay can become infinite.

Online Scheduling Under a Stability Guarantee

As before, suppose packets arrive according to a rate λ process at a transmission node with infinite queue capacity. The node transmits a packet p for a duration $\tau(b)$ when the backlog in the queue, excluding packet p, is b. The arrival rate λ is not known at the transmitter, but it is known that $\lambda \leq \lambda_{\max}$.

The transmitter needs to be designed to ensure stability, and since $\lambda_{\max}$ is a worst case estimate of the arrival rate, stability will be ensured if the rate of transmission is higher than $\lambda_{\max}$. Since a lazy schedule varies transmission times depending on the backlog according to the function $\tau(b)$, for stability it suffices that $\tau(b) < \frac{1}{\lambda_{\max}}$ for all b large enough.

Poisson arrivals: Let us define Lazy_1 as the schedule that sets $\tau_{\text{Lazy}_1}(b) = \alpha\frac{(1+b)}{\lambda_{\max}}(\frac{\pi^2}{6} - \sum_{k=1}^{b}\frac{1}{k^2})$. A natural candidate for comparison is the most generic non-adaptive schedule which sets each transmission time equal to a constant value $\tau_{Const}(b) = \frac{\alpha}{\lambda_{\max}}$. We call this schedule "Constant". In our comparison, the arrival process will be a rate λ. Observe that $\tau_{Lazy_1}(b) \to \frac{\alpha}{\lambda_{\max}}$ as $b \to \infty$.

Note that as long as $\alpha < 1$, both scheduling algorithms ensure stability for arrival rates less than $\lambda_{\max}$. We performed simulations using both scheduling algorithms for $\alpha = .95$, $\lambda_{\max} = 1$, varying λ from .3 to .9. To allow energy and delay to come close to equilibrium, each simulation was performed for $100,000$ arrivals. The results are given in Table 3.1.

The energy/packet values in Table 3.1 are dimensionless due to the normalization with noise PSD (see Equation 3.18). The energy values correspond to average SNR per packet of approximately 25 dB to 34 dB for Lazy_1, and 36 dB for Constant.

In order to give a fuller picture, let us also consider lower SNR values. We do this by considering lower rates. In the rest of this section, the maximum rate is set to 2 bits/transmission, while the symbol rate is kept the same as before[4]. Table 3.2 shows

[4]Since the symbol rate is 10^6 transmissions/sec, the minimum transmit time of a 10^4 bit packet (*i.e.*unit time) is now $\frac{10}{2}$ msec as opposed to the previous $\frac{10}{6}$. Note that λ is arrivals/unit time, hence for the same λ, the actual number of packet arrivals/second is lower than before.

λ	Poisson arrivals			
	Lazy_1		Constant	
	E/pkt $\times 10^{-4}$	Dly/pkt	E/pkt $\times 10^{-4}$	Dly/pkt
.3	91.6	3.19	1004.6	1.89
.4	118.8	3.56	1004.6	2.07
.5	159.4	4.01	1004.6	2.30
.6	218.4	4.60	1004.6	2.64
.7	308.6	5.51	1004.6	3.23
.8	435.1	6.92	1004.6	4.23
.9	623.7	9.58	1004.6	6.61

Table 3.1: Average energy/packet and average delay/packet for Lazy_1 and Constant over an infinite time horizon. Delay values are in milliseconds (High SNR).

how the energy per packet ranges for Lazy_1 and Constant. For Lazy, the SNR goes from 7 to 11 dB.

λ	Poisson arrivals			
	Lazy_1		Constant	
	E/pkt $\times 10^{-4}$	Dly/pkt	E/pkt $\times 10^{-4}$	Dly/pkt
.3	4.27	9.56	8.32	5.700
.4	4.48	10.66	8.32	6.219
.5	4.76	12.00	8.32	6.935
.6	5.14	13.76	8.32	7.984
.7	5.62	16.28	8.32	9.622
.8	6.22	20.27	8.32	12.567
.9	6.97	27.81	8.32	19.453

Table 3.2: Average energy/packet and average delay/packet for Lazy_1 and Constant over an infinite time horizon. Delay values are in milliseconds (Low SNR).

Bursty arrivals: We have just seen that the schedule Lazy_1 is more energy-efficient compared to a constant-transmission time schedule when the arrivals are Poisson. The schedule Lazy_1 was developed by conducting an asymptotic analysis of $\tau^*(b,t)$, where $\tau^*(b,t)$ is defined in equation (3.17). The asymptotic analysis for Poisson arrivals assumes that the inter-arrival times D_i in (3.17) are i.i.d. exponentials. Thus, Lazy_1 is "tuned" to Poisson arrivals.

It is therefore interesting to ask just how well Lazy_1 will perform under non-Poisson input processes. To this end we consider the following "bursty" arrival process: The inter-arrival times D_i are i.i.d. with $\Pr(D_i = a_1) = \beta = 1 - \Pr(D_i = a_2)$, where a_1, a_2 and β are parameters. When a_1 is small and β is large arrivals tend to be bursty with a high probability.

First, we run Lazy_1 on the bursty arrival process with $\frac{a_2}{a_1} = 9$, and $\lambda_{\max} = 1$. The results are summarized in Table 3.3. Comparing the energy/pkt values in the last three rows of Tables 3.1 and 3.3, we see that Lazy_1 is indeed better tuned for Poisson arrivals. A second conclusion from the tables is that, at low values of λ, lazy scheduling works better on the bursty arrival process than on the Poisson arrival process.

λ	Bursty arrivals			
	Lazy_1		Constant	
	E/pkt $\times 10^{-4}$	Dly/pkt	E/pkt $\times 10^{-4}$	Dly/pkt
.3	66.972	2.473	1004.6	$\simeq$ 1.583
.4	95.746	3.731	1004.6	$\simeq$1.583
.5	212.769	4.453	1004.6	$\simeq$ 1.583
.6	326.072	5.642	1004.6	2.480
.7	431.995	6.996	1004.6	4.233
.8	552.195	8.598	1004.6	6.263
.9	729.285	11.801	1004.6	10.607

Table 3.3: Average energy/packet and average delay/packet for Lazy_1 and Constant over an infinite time horizon. Delay values are in milliseconds.

Now we develop another algorithm, called Lazy_2, which is derived from the bursty arrival process, and hence potentially better tuned to it. Consider an infinite time horizon as above. Recall that, for a backlog of b, $\tau(b) = \sup_{n \geq 1} \left\{ \frac{1}{n+b} \sum_{i=1}^{n} D_i \right\}$. In order to obtain a bound on $\mathrm{E}(\tau(b))$, we consider:

$$
\begin{aligned}
\Pr(\tau(b) < r) &= \Pr\left(\sup_{n \geq 1} \{ \frac{1}{n+b} \sum_{i=1}^{n} D_i \} < r \right) \\
&= \Pr(\sum_{i=1}^{n} (D_i - r) < br, \ \forall\, n \geq 1)
\end{aligned}
$$

Define $Y_i^{(r)} \triangleq D_i - r$. For a given r, $Y_i^{(r)}$ are i.i.d. random variables of mean $\mathrm{E}(D) - r$. Define $\tilde{S}_n^{(r)} \triangleq \sum_{i=1}^n Y_i^{(r)}$. When $\mathrm{E}(D) - r < 0$, $\tilde{S}_n^{(r)}$ is a random walk with a negative drift. It is known (see Chapter 7 of [36], especially Problem 7.12) that the following bound holds

$$\Pr(\tilde{S}_N^{(r)} \geq br) \leq e^{-s^*\, b\, r}, \tag{3.33}$$

where s^* is the solution of the equation

$$\mathrm{E}(e^{sY_i^{(r)}}) = 1.$$

In our case, the above equation reduces to

$$\beta\, e^{s^*(a_1 - r)} + (1-\beta)\, e^{s^*(a_2 - r)} = 1 \tag{3.34}$$

Using the above definitions and results, $\Pr(\tau(b) \geq r) = \Pr(\tilde{S}_N \geq br) \leq e^{-s^*\, b\, r}$, provided $r > \mathrm{E}(D)$. Now we are ready to bound $\mathrm{E}(\tau(b))$:

$$\begin{aligned}
\mathrm{E}(\tau(b)) &= \int_0^\infty \Pr(\tau(b) \geq r)\, dr \\
&\leq \int_0^{\mathrm{E}(D)} Pr(\tau(b) \geq r) dr \;+\; \int_{\mathrm{E}(D)}^\infty e^{-s^*\, b\, r}\, dr \\
&\leq \mathrm{E}(D) \;+\; \int_{\mathrm{E}(D)}^\infty e^{-s^*\, b\, r}\, dr \\
&\triangleq B(b)
\end{aligned}$$

This suggests an online lazy schedule $\sigma(b) = \alpha * B(b)$, where $\alpha < 1$ is there to ensure stability. We will call this schedule Lazy_2.

The schedule Lazy_2, where $\sigma(b)$ is calculated as described above for $\frac{a_2}{a_1} = 9$, $1 - \beta = \frac{1}{9}$, and $\lambda_{\max} = 1$ is plotted in Figure 3.9 (for $\alpha = 1$). Note that as b grows, $\sigma(b)$ asymptotes to $\frac{1}{\lambda_{\max}}$ in the figure, and in general, it asymptotes to $\frac{\alpha}{\lambda_{\max}}$.

Figure 3.9: A plot of $\sigma(b)$ vs. b for a lazy schedule designed for $\frac{a_2}{a_1} = 9$, $\lambda_{\max} = 1$, and with $\alpha = 1$.

Table 3.4 summarizes results of the comparison of Lazy_2 with Constant on a bursty arrival process. Comparing Tables 3.3 and 3.4 shows that Lazy_2 is indeed a better schedule for the bursty arrivals process than is Lazy_1, as ought to be the case.

λ	Bursty arrivals			
	Lazy_2		Constant	
	E/pkt$\times 10^{-4}$	Dly/pkt	E/pkt$\times 10^{-4}$	Dly/pkt
.3	51.192	7.580	1004.6	$\simeq$1.583
.4	107.495	7.786	1004.6	$\simeq$1.583
.5	209.923	9.110	1004.6	$\simeq$1.583
.6	293.675	10.033	1004.6	2.480
.7	389.735	11.159	1004.6	4.233
.8	513.605	12.959	1004.6	6.263
.9	692.246	16.492	1004.6	10.607

Table 3.4: Bursty arrivals: Average energy/packet and average delay/packet for Lazy_2 and Constant over an infinite time horizon. Delay values are in milliseconds.

The simulation results demonstrate that lazy schedules achieve significantly lower energy than the Constant schedule with a moderate increase in average delay. This comparison with the constant schedule is important since for a given mean service time, the constant schedule achieves the smallest average delay among all schedules that are independent of the arrival process and hence oblivious to backlogs [76]. In turn this implies that the constant schedule has the largest transmission times and hence the lowest energy among backlog-oblivious schedules. The fact that our suboptimal lazy schedule is more energy efficient than the constant schedule demonstrates the advantage of lazy scheduling.

3.6 Inclusion of a Constant Power Component

Much of the analysis is unchanged when energy functions are convex but not decreasing. Of special importance is the case when the energy function $w(\tau)$ is non-increasing for $\tau < t_b$, and non-decreasing for $\tau > t_b$, for some $t_b > 0$. In this case, the optimum transmission durations τ^{**} are obtained from the $\{\tau^*\}$ of Definition 1 by: $\tau_i^{**} = \min(\tau^*, t_b)$.

The importance of this case is in its practical relevance. In practical transceivers, transmission is not the only cause of power consumption. The device itself consumes a certain, usually constant, power to operate, and to receive data. The energy due to these components can be modeled as a linear, increasing function of transmission duration. When this linear function is added to the convex, decreasing transmission energy function, a convex function is obtained, that decreases up to a certain point and increases afterward. Such an energy function is a good model for many practical tranceivers. We state the following lemma for completeness:

Lemma 7 *When $w(\tau) = w_1(\tau) + c\tau$, where $w_1(\tau) > 0$ is convex, decreasing and $c > 0$ is a constant, the optimal transmission durations are $\tau_i^{**} = \min(\tau^*, t_b)$ where $t_b = \arg_{\tau>0} \min(w(\tau))$.*

Proof. Take the schedule $\{\tau^*\}$. Consider the smallest i for which $\tau_i^* > t_b$. Due to the properties of the optimal schedule, either $i = 1$ or there is no such i (in which case the proof is complete). If $i = 1$, by reducing the first transmission time so that $\tau_1^{**} = t_b$, the total energy is reduced. Now, note that $\tau_2^* \leq \tau_1^*$. If $\tau_2^* = \tau_1^*$, again it reduces energy to make $\tau_2^{**} = t_b$. If $\tau_2^* < \tau_1^*$, either $\tau_2^* \geq t_b$, in which case it will again be reduced to t_b, or $\tau_2^* < t_b$, in which case it can be increased to reduce energy. However, since $\tau_2^* < \tau_1^*$, 2 is the start of a band, hence due to causality it cannot be extended to the left. But, since $\tau_2^* \geq \tau_3^*$, extending it to the right (which would decrease some τ_j, $j > 2$) increases energy, hence we stop. ∎

3.7 Conclusions

In this chapter the point-to-point minimum energy packet scheduling problem was formulated, solved in an offline setting, and the idea of conserving energy by lazy scheduling of packet transmissions was put forth. Specifically, an optimal offline schedule for a node operating under a deadline constraint was obtained. An inspection of the form of this schedule naturally led to an online schedule, which was shown, through simulations, to be quite energy-efficient. The deadline constraint was then relaxed and an exact probabilistic analysis of the offline scheduling algorithm was done. A heuristic online algorithm, which varies transmission times according to backlog was devised, and it was shown experimentally that it is more energy efficient than a constant schedule with the same stability region and similar delay.

In closing, we would like to pose the following question which has been left unanswered in the chapter: Suppose packets arrive according to a Poisson process to a node which has a queue of infinite capacity. The transmission time of the packet at the head of the queue is $\tau(b)$ when the backlog in the queue, excluding packet p, is b. Let $\vec{\tau} = [\tau(0), \tau(1), \ldots]$, and let $D(\vec{\tau})$ be the corresponding average delay of a packet. Let

$$\Sigma = \{\vec{\tau} : D(\vec{\tau}) \leq \bar{D}\} \tag{3.35}$$

be the class of all online transmission schedules which meet the average delay constraint $\bar{D}$.

Problem: Which schedule $\vec{\tau} \in \Sigma$ minimizes the average transmission energy per packet?

This important problem remains open and seems quite difficult to address. However, the competitiveness of the heuristic online schedules that will be described in Chapter 5 with respect to the offline optimal does not leave much room for improvement by producing such an optimal online schedule.

Ata et al. [4, 5] address the related question of finding the control policy that minimizes the long term average energy per packet subject to a finite buffer constraint, where packet drops incur a certain penalty. The problem is solved under the assumption of Brownian traffic and the structure of the optimal power control policy is shown.

Chapter 4

Scheduling in a Multi-User Setting

The previous chapter formulated and analyzed the basic single-user energy-efficient scheduling problem. The channel was assumed to be time-invariant so that energy as a function of transmission duration does not depend on when a packet is transmitted. Now, we turn to the more realistic wireless communication scenario where the channels (hence the energy functions) are time-varying due to interference and fading. Specifically, we consider minimum-energy scheduling problems over multi-access channels, broadcast channels, and channels with fading. For concreteness, throughout this chapter we assume rates and powers corresponding to optimal coding over discrete-time AWGN channels. Our results, however, hold for more general channels and coding schemes where the total transmitted power is convex in the transmission rates.

The key results are: (i) showing that for each of these channels, offline scheduling reduces to a convex optimization problem with linear constraints, (ii) devising an algorithm, FlowRight, that iteratively finds the optimal offline schedules (Section 4.2.1). We focus on offline analysis in this chapter. Online algorithms will be the subject of the next chapter, where a heuristic for jointly adapting to both channel fading state and backlog, that achieves energy efficiency close to the optimal offline schedule, will be presented.

The rest of the chapter is organized as follows. First we consider transmission schedules for the multi-access channel. The MoveRight algorithm (proposed in [26]) can find the best *time-sharing* solution to the offline multi-access problem. However, to find the *optimal* solution, we need to consider general multi-access coding schemes, where the users can simultaneously transmit. In the following section we define the multi-access offline scheduling problem and show that it can be cast as a convex optimization problem with linear constraints. In Section 4.2.1, we present FlowRight, which solves this problem. FlowRight is also shown to optimally solve the offline scheduling problem for the broadcast channel. In Section 4.3 we turn to channels with fading, and show that FlowRight can optimally solve the offline scheduling problem in a slow fading channel with perfect CSI at the transmitter and the receiver. The results are shown to generalize to multi-access and broadcast channels. We also show that scheduling in a fast fading channel reduces to the single user problem of [65].

4.1 Preliminaries

It will be useful to start by stating two problems. The first will be a repetition of the main problem of Chapter 3, summarized here for convenience. Recall that the setup was as follows: m packets arrive at the transmitter's buffer in the interval $[0, T)$ at times $0 = t_1, t_2, \ldots, t_m < T$. The node is required to transmit all m packets within the interval $[0, T]$. The question is, how should the packet transmissions be scheduled to minimize the total energy required to transmit the packets. If we let τ_i be the transmission time for packet i, $1 \leq i \leq m$, the offline version of the problem is:

Problem 1 : Single-transmitter single-receiver offline scheduling *Given a vector of packet arrival times* $\{t_i, i = 1, ..., m\}$*, where* $t_1 = 0$*,* $t_i < t_{i+1}$*, and* $t_m < T$*, and an energy function* $w(\tau)$ *that is monotonically decreasing and convex, find a schedule* $\{\tau_i\}_{i=1}^m$ *so as to minimize the total transmission energy:* $\sum_{i=1}^m w(\tau_i)$ *subject to causality*[1] *and deadline*

[1] In our setting, causality corresponds to the obvious constraint that no packet's transmission can start before its arrival time.

constraints.

The following explicit solution was found in Chapter 3:

Define $k_0^* = 0$, and

$$k_j^* = \arg\max_{\mathrm{k}_{\mathrm{j}-1}^*+1\leq \mathrm{k}\leq \mathrm{m}} \left\{ \frac{1}{\mathrm{k}-\mathrm{k}_{\mathrm{j}-1}^*} \sum_{\mathrm{i}=\mathrm{k}_{\mathrm{j}-1}^*+1}^{\mathrm{k}} \xi_{\mathrm{i}} \right\},\ \mathrm{j}=1,2,\ldots,\min\{\mathrm{l}:\mathrm{k}_{\mathrm{l}}^*=\mathrm{m}\}.$$

The sequence $\tau_1^*, \tau_2, \ldots, \tau_m^*$, given by

$$\tau_j^* = \frac{1}{k_{(l+1)}^* - k_l^*} \sum_{i=k_l^*+1}^{k_{(l+1)}^*} \xi_i, \quad k_l^* \leq j \leq k_{(l+1)}^*, \tag{4.1}$$

where ξ_i's are the packet inter-arrival times, is the optimal schedule.

In [26] the minimum energy scheduling problem for a multiple user channel, *e.g.*, uplink and downlink, involving several transmitters and receivers where time-sharing is used is investigated. The setup is identical to that of the previous problem except that packets can have different energy functions. The offline time-sharing scheduling problem is formulated as follows.

Problem 2 : Multiple-user offline time-sharing scheduling [26] *Given a vector of packet arrival times* $\{t_i, i = 1, ..., m\}$*, where* $t_1 = 0$*,* $t_i < t_{i+1}$*, and* $t_m < T$*, and energy functions* $w_i(\tau)$ *that are strictly monotonically decreasing and convex, find a schedule that minimizes the total transmission energy:* $\sum_{i=1}^{m} w_i(\tau_i)$ *subject to causality and deadline constraints.*

This is also a convex optimization problem but does not in general admit a simple closed-form solution. By exploiting the special features of the problem, an algorithm, MoveRight, which finds the global optimal schedule efficiently, is developed. MoveRight iteratively moves the start times of packet transmissions, one packet at a time, so that each move locally optimizes the energy function. The algorithm was shown to solve other scheduling problems, such as when packets have individual deadlines, and when the transmit buffer is finite.

4.2 Scheduling for the Multi-Access Channel

Consider the discrete-time AWGN multi-access channel with K transmitters and a single receiver. Data packets are generated at each transmitter's buffer at arbitrary times t_i, in the interval $[0, T)$. Figure 4.1 shows an example sequence of packet arrival times for two users, where packet arrival times of users 1 and 2 are marked by crosses and circles, respectively. These packets must be transmitted reliably to the receiver in the time interval $(0, T]$. The received signal at time k is

$$Y[k] = \sum_{i=1}^{K} \sqrt{s_i[k]} X_i[k] + Z[k], \tag{4.2}$$

where $X_i[k]$ is user i's signal, and $Z[1], Z[2], \ldots$ are i.i.d. zero-mean Gaussian noise with variance σ^2. For now, we assume the s_i's to be constant in time. Later we consider $s_i[k]$'s that vary with k to model frequency-flat fading.

Figure 4.1: Packet arrivals in $[0, T)$

It is well known that for the AWGN multi-access channel the set of feasible average *received* powers, $\{P_i\}_{i=1}^{K}$, for a given set of rates $\{r_i\}_{i=1}^{K}$ is given by (see [69]):

$$\sum_{i \in S} P_i \geq \sigma^2 (2^{2(\sum_{i \in S} r_i)} - 1),$$

for all $S \subset \{1, 2, \ldots, K\}$. To achieve points on the boundary of the region, one needs to use optimal codes with block lengths approaching infinity. However, for long enough packets one can come arbitrarily close to the boundary using codes with finite block lengths and achieving a reasonable level of reliability. To simplify expressions here, and throughout the paper, we assume codewords are long enough so that points on the boundary are basically achievable, but much shorter than the time window in which they must be transmitted. We restrict our discussion to two users, set $\sigma^2 = 1$ and define $f(r) \triangleq 2^{2r} - 1$. The results can be readily extended to more than two users.

For $K = 2$, the feasible region of received powers is simply:

$$
\begin{aligned}
P_1 &\geq f(r_1), \\
P_2 &\geq f(r_2), \\
P_1 + P_2 &\geq f(r_1 + r_2),
\end{aligned}
$$

This is plotted in Figure 4.2. Note that when the received power is P_i, the *transmitted* power is P_i/s_i. Time-sharing, *i.e.*, one user transmitting at rate $\frac{r_1}{\alpha}$ for a fraction α of the time and the second transmitting at rate $\frac{r_2}{1-\alpha}$ for a fraction $1-\alpha$ of the time, yields the region with the dashed boundary specified by: $P_1 = \alpha f(\frac{r_1}{\alpha})$ and $P_2 = (1-\alpha) f(\frac{r_2}{1-\alpha})$. Note that the time-sharing boundary always touches the boundary of the multi-access region at the point $(\alpha f(r_1 + r_2), (1-\alpha) f(r_1 + r_2))$, where $\alpha = \frac{r_1}{r_1 + r_2}$.

Figure 4.2: Feasible Region of (P_1, P_2) for a given (r_1, r_2). The multi-access region is bounded below by the solid boundary and the time-sharing region is bounded below by the dashed boundary.

We refer to the sequence of arrivals of the i^{th} user as *stream* i and merge the two streams into one sequence $\{t_i\}$, as shown in Figure 4.1, where stream 1 arrivals are marked by crosses and stream 2 arrivals are marked by circles. We denote the inter-arrivals of this new sequence by *data epochs*, or in short, epochs, and mark them ξ_i, $i = 1, \ldots, m$. Without loss of generality, we assume that a packet (from either one of the two users) is received at time 0, so $t_1 = 0$.

Before we present the multi-access offline scheduling problem, we make the following two key observations (Lemmas 8,9), the first of which can be obtained from Lemma 3.3 in [69], but included here for completeness.

Lemma 8 I*n the symmetric case ($s_1 = s_2$), B_1 and B_2 bits can be transmitted in τ time units with minimum energy by time sharing between the users. In the asymmetric case ($s_1 < s_2$), time-sharing is strictly sub-optimal, and the unique optimal scheme is the corner point of the multi-access energy region where P_1 is at its minimum possible value. In the AWGN case, this point corresponds to successive cancellation where users are decoded in decreasing order of s_i's.*

Proof: Since the duration of the interval and the number of bits to be transmitted in that interval by each user are fixed, the average rates will be fixed at $r_{1i} = \frac{B_1}{\tau}$, and $r_{2i} = \frac{B_2}{\tau}$. Given this average rate pair, we wish to minimize the total transmitted energy, $\tau(P_1/s_1 + P_2/s_2)$. The solution can be readily seen from the multi-access achievable powers region in Figure 4.2. When $s_1 = s_2$, any power pair on the line $P_1 + P_2 = f(r_1 + r_2)$ achieves the minimum. In particular the point where the time-sharing boundary touches the multi-access boundary minimizes the total transmitted power for the time-sharing scheme. When $s_1 \neq s_2$, the minimum is attained at one of the two corner points. For example when $s_1 < s_2$, the optimal power pair is $(f(r_1), f(r_1 + r_2) - f(r_1))$. In the AWGN channel case, this is $(P_1, (P_1 + \sigma^2)(2^{2r_2} - 1))$ where $P_1 = \sigma^2(2^{2r_1} - 1)$, which can be achieved by decoding user 2, subtracting its signal from the received signal, and then decoding 1. ■

Lemma 9 I*n an optimal multi-access offline schedule, the rate of a user need not change during an epoch.*

Proof: By definition, new data can only arrive at the start of a data epoch. So, at the beginning of an epoch, the two users together have a certain number of bits to be transmitted, and no new bits are added to this during the epoch. Now, let us focus on a generic data epoch in the optimal schedule. Assume without loss of generality that the epoch starts at $t = 0$ and ends at $t = t_e$, and that $1/s_1 = a > 1$, $1/s_2 = 1$. Assume that in this schedule the epoch is divided into k intervals, $[t_1 = 0, t_2)$, $[t_2, t_3)$, $\ldots$, $[t_k, t_e = t_{k+1})$, where the rates of both users are constant during an interval. Denote the first user's rate in interval $[t_i, t_{i+1})$ by r_{1i}, and the second's by r_{2i}. At optimal power settings (see Lemma 8) the total transmitted energy for the data epoch is given by $\mathcal{E} = \sum_{i=1}^{k}(t_{i+1} - t_i)((a-1)f(r_{1i}) + f(r_{1i} + r_{2i}))$. By convexity of f, it is easy to see that the total energy can be decreased by using the average rates $r_1 = \sum_{i=1}^{k} r_{1i}(t_{i+1} - t_i)/t_e$ and $r_2 = \sum_{i=1}^{k} r_{2i}(t_{i+1} - t_i)/t_e$ throughout the epoch. ■

We are now ready to state the minimum energy offline scheduling problem for the multi-access channel. For simplicity, consider equal sized packets each with B bits. The formulation and the results we obtain, however, can be readily generalized to packets of unequal size. Define the sequences $\{c_{1i}\}$ and $\{c_{2i}\}$ as the number of bits that have arrived at the beginning of epoch i for users 1 and 2, respectively. In the case of constant sized packets, this means that for $i \in \{1, \ldots, m\}$, $c_{ji} = B$ if there is a stream j arrival at the beginning of data epoch i, and 0 otherwise. By Lemma 9, the optimal multi-access offline scheduling problem reduces to finding a rate pair sequence $\{(r_{11}, r_{21}), (r_{12}, r_{22}), \ldots, (r_{1m}, r_{2m})\}$ that minimizes the total energy. The problem is then as follows:

Problem 3 : Multi-access channel offline scheduling

$$\begin{aligned}
&\textit{Minimize:} && \sum_{i=1}^{m} \xi_i((a-1)f(r_1) + f(r_1 + r_2)) \\
&\textit{subject to:} && \sum_{i=1}^{k} r_{ji}\xi_i \leq \sum_{i=1}^{k} c_{ji}\ ,\ k = 1, \ldots, m-1,\ j = 1, 2 \\
& && \sum_{i=1}^{m} r_{ji}\xi_i = \sum_{i=1}^{m} c_{ji},\ j = 1, 2.
\end{aligned}$$

Thus, similar to problems (1) and (2), multi-access channel scheduling is a convex optimization problem with linear constraints. We now show that in the symmetric case ($s_1 = s_2$) time-sharing is optimal and thus the problem can be optimally solved using MoveRight. For convenience, define $g(r_1, r_2) \triangleq (a-1)f(r_1) + f(r_1 + r_2)$, and note that g is convex in r_1 and r_2.

Theorem 4 I*n the symmetric case there exists a time-sharing schedule that minimizes total energy. In the asymmetric case, time-sharing is strictly sub-optimal.*

Proof. Consider the symmetric case first. We show that any schedule can be converted into a time-sharing schedule with equal or lower energy. First, note that from Lemma 2, it suffices to consider the schedule as a sequence of rate pairs, (r_{1i}, r_{2i}), one pair for each epoch. Also note that we can limit attention to the case where the received powers (P_{1i}, P_{2i}) are the optimal corner point of the feasible region for rates (r_{1i}, r_{2i}) (if not, the energy can be reduced without changing the schedule). Consider epoch i. From Lemma 8, in the symmetric case there is a point on the time-sharing curve that achieves the average rates (r_{1i}, r_{2i}) with minimum total energy. We can move to this time-sharing point by letting the first user transmit alone in a fraction α_i of the total interval, *i.e.*, for a duration $\alpha_i \xi_i$, using a rate $\frac{r_{1i}}{\alpha_i}$, and the second user transmit in the remaining with rate $\frac{r_{2i}}{1-\alpha_i}$, where $\alpha_i = \frac{r_{1i}}{r_{1i}+r_{2i}}$. Proceeding like this with other epochs, the schedule we were provided with has been converted to a time sharing schedule of equal or lower energy. Now, consider a packet from user 1 that is being transmitted across the epochs $(i, i+1, \ldots, i+l)$, as l chunks, with instantaneous rates $[\frac{r_{1i}}{\alpha_i}, \frac{r_{1(i+1)}}{\alpha_{i+1}}, \ldots, \frac{r_{1(i+l)}}{\alpha_{i+l}}]$. This packet has B bits, so $\sum_{j=0}^{l} r_{1(i+j)} \tau_{1(i+j)} = B$. The same amount of data can be transmitted by averaging user 1's rate over the l pieces, setting $\tilde{r} = \sum_{j=1}^{l} \frac{r_{1(i+j)}}{\alpha j} \beta_j$ where $\beta_j = \frac{\alpha_j \tau_{i+j}}{\sum_k \alpha_k \tau_{i+k}}$. By convexity of the power function f, energy is reduced. Now that the rate has been averaged out, one can always collect these l pieces together to transmit the packet of user 1 as a whole. All of the above can be repeated for user 2. The result is time sharing between the packets of user 1 and user 2, where rates (and powers) are set independently.

Now, consider the asymmetric case. Take any epoch where there is time-sharing. By Lemma 8 one can convert[2] this epoch's rates to the optimal corner point on the multi-access boundary, and strictly decrease energy. ∎

Problem 3 is a convex optimization problem and considering the conditions on $f(.)$, it

[2] If causality does not permit this, then pick another time-sharing interval.

is easy to see that it has a unique solution. However, except for the symmetric case, the problem has no closed form solution. In the following section we describe FlowRight, an algorithm for finding the optimal offline schedule. In Section 4.2.2 we compare the performance of time-sharing schedule using MoveRight to the optimal offline schedule computed by FlowRight.

4.2.1 FlowRight: An Algorithm for Optimal Offline Scheduling

FlowRight is an iterative algorithm. In the beginning, the transmission time of each packet is set to precisely the data epoch at the beginning of which the packet arrived. That is, packets begin transmission when they arrive, and end transmission when the next data epoch starts. Let the rates obtained in this way be $\{r_{1i}^0\}$ and $\{r_{2i}^0\}$, such that $r_{ji}^0 = \frac{c_{ji}}{\xi_i}$, $j = 1, 2$, $i = 1, 2, \ldots, m$. The FlowRight algorithm is based on doing local optimizations on pairs of epochs. Consider the first two data epochs. The total number of bits transmitted by users 1 and 2 in these two data epochs are $c_1 = r_{11}^0\xi_1 + r_{12}^0\xi_2$ and $c_2 = r_{21}^0\xi_1 + r_{22}^0\xi_2$, respectively. Keeping the number of bits fixed at c_1 and c_2, we update (r_{11}^0, r_{21}^0) to (r_{11}^1, r_{21}^1), where (r_{11}^1, r_{21}^1) is the allocation of rates to the first data epoch that minimizes the overall energy of the pair of data epochs. Obviously, when (r_{11}^1, r_{21}^1) are decreased (*i.e.*, at least one component is decreased and neither is increased), (r_{12}^1, r_{22}^1) will increase, since the bits that leave the first epoch go to the second[3]. Note that $r_{11}^1 \leq r_{11}^0$ and $r_{21}^1 \leq r_{21}^0$, since, from the initial condition, information can only flow to the right (otherwise causality would be violated.) We therefore have to reset (r_{12}^0, r_{22}^0) to new values which are larger (or equal to) their initial values.

Moving to the second pair of data epochs, this time optimally decrease (r_{12}^0, r_{22}^0) to (r_{12}^1, r_{22}^1), and reset the values of (r_{13}^0, r_{23}^0). Proceed in this way to obtain (r_{1i}^1, r_{2i}^1) for $i = 1, \ldots, n$. This completes the first *pass* of the algorithm (see Fig. 4.2.1 and 4.2.1 for an illustration.) It is easy to see that in the first pass, information can only flow to the right.

[3] We are allowing fractional numbers of bits to move between epochs.

Interestingly, we will later prove that information always flows right in the algorithm, and consequently, after each iteration the rates are closer to the optimal solution than they were before that iteration.

Figure 4.3: Illustration of the FlowRight algorithm

Figure 4.4: Illustration of the FlowRight algorithm

After the first pass is complete we start from the beginning and update the rates two data epochs at a time similarly to the above. Terminate after pass K, where $K = \min\{k : r_{ji}^k = r_{ji}^{k-1},\ i = 1, \ldots m, \text{ and } j = 1, 2\}$. A pseudo-code for the algorithm is given below.

```
passes=0;
for i=1:n
{
    r_{1i}^0 = c_{1i}/ξ_i;
    r_{2i}^0 = c_{2i}/ξ_i;
}
done=0;
while (done==0)
{
    passes=passes++;
    for i=1:n-1
    {
        [r_{1i}^k r_{1(i+1)}^k r_{2i}^k r_{2(i+1)}^k] = update ([r_{1i}^{k-1} r_{1(i+1)}^{k-1} r_{2i}^{k-1} r_{2(i+1)}^{k-1}]);
    }
    if (r_1^k==r_1^{k-1} and r_2^k==r_2^{k-1})
    {
        done=1;
```

```
    }
}
```

In each pass, the function `update` is run $m-1$ times, i.e., once for each consecutive pair of data epochs. Consider the k^{th} pass when `update` is running on epochs i, $i+1$. The problem is that of choosing (r_{1i}, r_{2i}), while the total number of bits on each stream is fixed at $\xi_i r_{1i}^{k-1} + \xi_{i+1} r_{1(i+1)}^{k-1} = b_{1i}^{(k-1)}$ and $\xi_i r_{2i}^{k-1} + \xi_{i+1} r_{2(i+1)}^{k-1} = b_{2i}^{(k-1)}$, so as to minimize the total energy of epochs i and $i+1$. For some (r_{1i}, r_{2i}), the total energy is $h^{(b_{1i}^{(k-1)}, b_{2i}^{(k-1)})}(r_{1i}, r_{2i}) \triangleq \xi_i g(r_{1i}, r_{2i}) + \xi_{i+1} g(\frac{b_{1i}^{(k-1)} - r_{1i}\xi_i}{\xi_{i+1}}, \frac{b_{1i}^{(k-1)} - r_{2i}\xi_i}{\xi_{i+1}})$. Observe that $h^{(b_{1i}^{(k-1)}, b_{2i}^{(k-1)})}(r_{1i}, r_{2i})$ is convex in r_{1i} and r_{2i}.

Now we prove that this algorithm finds the optimal offline schedule. We first present a lemma that will be useful in the proof. Consider two data epochs, 1 and 2, with durations ξ_1 and ξ_2. The first stream needs to transmit a total of b_1 bits in the two data epochs and the second stream needs to transmit b_2 bits. Of the b_1 bits, $b_{11} \leq b_1$ are available in the first data epoch, and of the b_2 bits, $b_{21} \leq b_2$ are available in the first data epoch. Let the rates chosen for the first epoch be r_{11}, r_{21}, hence the total energy is $h^{(b_1, b_2)}(r_{11}, r_{21})$.

Lemma 10 *The following hold.*

1. *The optimal rate pair $(\hat{r}_{11}, \hat{r}_{21})$ is unique, and satisfies $h_x(\hat{r}_{11}, \hat{r}_{21}) \leq 0$, and $h_y(\hat{r}_{11}, \hat{r}_{21}) \leq 0$, where h_x and h_y are the partial derivatives of h with respect to the first and second components, respectively. If $h_x(\hat{r}_{11}, \hat{r}_{21}) < 0$, then a new packet is starting transmission at epoch 2 on stream 1, and if $h_y(\hat{r}_{11}, \hat{r}_{21}) < 0$, a new packet starts on stream 2.*

2. *Suppose some additional bits are injected into epoch 1 on one or both streams, that is, set $r'_{11} = r_{11} + \Delta_{11}$, and $r'_{21} = r_{21} + \Delta_{21}$, $\Delta_{11} \geq 0$ and $\Delta_{21} \geq 0$. Also, some bits are ejected from the second epoch, i.e. set $r'_{12} = r_{12} - \Delta_{12}$, and $r'_{22} = r_{22} - \Delta_{22}$, $\Delta_{12} \geq 0$ and $\Delta_{22} \geq 0$. For the change to be non-trivial, we require that at least one of*

$\{\Delta_{11}, \Delta_{12}, \Delta_{21}, \Delta_{22}\}$ is non-zero. Notice that the constraint space of the problem has changed. In this case:

(a) If $h_x(\hat{r}_{11}, \hat{r}_{21}) = 0$ and $h_y(\hat{r}_{11}, \hat{r}_{21}) = 0$, then after the injection/ejection of the new bits, when `update` *is run again, there will be a right push, i.e. a non-negative amount of information will move from epoch 1 to epoch 2 on both streams.*

(b) If $h_x(\hat{r}_{11}, \hat{r}_{21}) < 0$, the first stream was limited by causality before the injection/ejection, hence it is limited by causality again (because no information has crossed from epoch 1 into epoch 2). If $h_x(\hat{r}_{11} + \Delta_{11}, \hat{r}_{21} - \Delta_{21}) > 0$, there will be a push on stream 1. Otherwise, due to causality, re-optimization will not result in any move on that stream (i.e. no push or pull). Similarly, if $h_y(\hat{r}_{11}, \hat{r}_{21}) < 0$, on the second stream there may only be a push or no movement at all.

Proof:

1. For simplicity, we shall drop the superscript and refer to $h^{(b_1,b_2)}(r_{11}, r_{21})$ as $h(r_{11}, r_{21})$. This function is strictly convex in both variables, and $(\hat{r}_{11}, \hat{r}_{21})$ is the result of minimizing it over a bounded region. Hence the solution is unique. The solution, $(\hat{r}_{11}, \hat{r}_{21})$, is either at the boundaries of the region defined by $0 \leq r_{11} \leq b_{11}/\xi_1$, $0 \leq r_{21} \leq b_{21}/\xi_1$, or inside. If it is inside, it must satisfy $h_x(\hat{r}_{11}, \hat{r}_{21}) = 0$ and $h_y(\hat{r}_{11}, \hat{r}_{21}) = 0$.

   Due to convexity, partial derivatives of h are monotonic increasing. At the point $\hat{r}_{11} = \hat{r}_{21} = 0$, h_x and h_y are both negative (this can be seen by substituting the values). If $h_x = 0$ is not achieved in the region, then due to monotonicity, $h_x(b_{11}/\xi_1, r_{21}) < 0$ for all permissible values of r_{21}. In this case, increasing the rate r_{11} further than the boundary would decrease total energy, but this cannot be done due to *causality* constraints, so $\hat{r}_{11} = b_{11}/\xi_1$. Similarly, if $h_y = 0$ is not achieved inside the region, then $\hat{r}_{21} = b_{21}/\xi_1$.

2. First consider case (a), i.e. $h_x(\hat{r}_{11}, \hat{r}_{21}) = 0$ and $h_y(\hat{r}_{11}, \hat{r}_{21}) = 0$. Define $b_j^{'} = \xi_1 r_{j1}^{'} + \xi_2 r_{j2}^{'}$ for $j = 1, 2$. Due to strict convexity (hence the monotonicity of the derivative)

it can be easily shown that $h_x^{(b_1',b_2')}(\hat{r}_{11} + \Delta_{11}, \hat{r}_{21} + \Delta_{21}) \geq h_x^{(b_1,b_2)}(\hat{r}_{11}, \hat{r}_{21}) = 0$, and that the inequality is *strict* unless $b_1' = b_1$, $b_2' = b_2$, $\Delta_{11} = 0$, *and* $\Delta_{21} = 0$. But $h_x^{(b_1',b_2')}(0,0) < 0$. Hence, energy is uniquely minimized by a point $[\tilde{r}_{11}, \tilde{r}_{21}]$ that satisfies $0 < \tilde{r}_{11} < \hat{r}_{11} + \Delta_{11}$ and $0 < \tilde{r}_{21} < \hat{r}_{21} + \Delta_{21}$. This solution results from *pushing* a nonzero amount of information *right*, from data epoch 1 to data epoch 2. In case (b), $h_x(\hat{r}_{11}, \hat{r}_{21}) < 0$ or $h_y(\hat{r}_{11}, \hat{r}_{21}) < 0$. So when the injection and subtraction is done, these derivatives can remain negative or become positive. In the case that they become positive, there will be a right push. If either of these, say h_x, remains negative, then a right push (*i.e.*, decreasing r_{11}) on that stream can only increase the total energy. That stream was shown in part 1 to be limited by causality, and it still is, because injection brought only bits from the left. So we cannot pull any bits from epoch 2 to epoch 1 on this stream (*i.e.*, increase r_{11}). Hence, there will be no move on this stream. ■

Now, let $\{r_{1i}^{opt}\}$ and $\{r_{2i}^{opt}\}$ be the pair of *optimal* rate sequences. We will sometimes use the shorthand r^{opt} to refer to this pair. Such a unique solution exists because of the convexity of the problem and the compactness of the search space. In the following it will be proved that the algorithm **FlowRight** results in $\{r_{1i}^{opt}\}$ and $\{r_{2i}^{opt}\}$. In order to show that, we first argue that the algorithm stops at pair of sequences $\{r_{1i}^{\infty}\}$, $\{r_{2i}^{\infty}\}$. We then show that this is identical to r^{opt}. The following results are proved similarly to Theorem 1 in [26].

Theorem 5 T*he following statements hold:*

1. *As the algorithm FlowRight runs, information always flows right.*

2. *FlowRight stops, and returns two sequences* $\{r_{1i}^{\infty}\}$ *and* $\{r_{2i}^{\infty}\}$.

3. $\{r_{1i}^{\infty}\} = \{r_{1i}^{opt}\}$, *and* $\{r_{2i}^{\infty}\} = \{r_{2i}^{opt}\}$.

Proof:

1. The claim is that throughout the running of FlowRight, all pushes are to the right. We now prove this by induction. In the first pass, the claim is trivially true, since all left pushes are impossible due to causality. Now, suppose that we are on the k^{th} pass of the algorithm, and so far `update` has operated on all epoch pairs up to and including the pair $(i-1, i)$, and all pushes so far have been to the right. We will show that the next push will be to the right. On the $(k-1)^{st}$ run, `update` performed a local optimization on epochs $(i, i+1)$. Let us call the two pairs of rates resulting from this optimization $(\hat{r}_{1i}, \hat{r}_{2i})$, and $(\hat{r}_{1(i+1)}, \hat{r}_{2(i+1)})$. The number of bits transmitted in the i^{th} epoch on streams 1 and 2 are $b_{1i} \triangleq \hat{r}_{1i}\xi_i$ and $b_{2i} \triangleq \hat{r}_{2i}\xi_i$. Similarly, define $b_{1(i+1)}$ and $b_{2(i+1)}$ to be the bits transmitted the next epoch, and define $b_1 \triangleq b_{1i} + b_{1(i+1)}$ and $b_2 \triangleq b_{2i} + b_{2(i+1)}$. From Lemma 10, part 1, $h_x^{(b_1,b_2)}(\hat{r}_{1i}, \hat{r}_{2i}) \leq 0$ and $h_y^{(b_1,b_2)}(\hat{r}_{1i}, \hat{r}_{2i}) \leq 0$. Now, as the $(k-1)^{th}$ pass progresses, `update` performs a local optimization on $(i+1, i+2)$, and this, by hypothesis, results in a right push (i.e. a push from $i+1$ onto $i+2$), which changes $(r_{1(i+1)}, r_{2(i+1)})$ to $(\hat{r}_{1(i+1)} - \Delta_{1(i+1)}, \hat{r}_{2(i+1)} - \Delta_{2(i+1)})$, where $\Delta_{j(i+1)} \geq 0$ for $j = 1, 2$. Continuing to the present time, on the k^{th} pass there is a right push (again, by the induction hypothesis), from $i-1$ to i resulting in $(\hat{r}_{1i} + \Delta_{1i}, \hat{r}_{2i} + \Delta_{2i})$, where $\Delta_{ji} \geq 0$ for $j = 1, 2$. By part 2 of Lemma 10, there can only be a right push (if any) from i to $i+1$ on the k^{th} iteration.

2. Consider $r_{11}^k \xi_{11} = b_{11}^k < \infty$, i.e. the total number of bits of stream 1 on epoch 1 after the k^{th} iteration. Since all pushes are to the right, b_{11}^k is monotonically non-increasing. Also, it's obviously bounded from below by zero. Therefore $b_{11}^k \downarrow b_{11}^\infty$, and therefore $r_{11}^k \downarrow r_{11}^\infty$. Similarly, $b_{11}^k + b_{12}^k$ (the total number of bits in epochs 1 and 2 on stream 1) is monotonic non-increasing and bounded below by zero. Hence, this sum tends to a limit; $(b_{11}^k + b_{12}^k) \downarrow (b_{11}^\infty + b_{12}^\infty)$. Therefore, $b_{12}^k \downarrow b_{12}^\infty$, and $r_{12}^k \downarrow r_{12}^\infty$. Proceeding like this, we will see that $r_{ji}^k \downarrow r_{ji}^\infty$, $j = 1, 2$, for all i. Hence the sequences of rates converge.

3. First, observe that $h_x(r_{1i}^\infty, r_{2i}^\infty) \leq 0$ and $h_y(r_{1i}^\infty, r_{2i}^\infty) \leq 0$ for all i. To see why this is true, suppose it is not. That is, let $h_x(r_{1\hat{i}}^\infty, r_{2\hat{i}}^\infty) > 0$ for some $\hat{i}$. Then, if we run FlowRight on this sequence, there will be a right push. This contradicts the fact that

$\{(r_{1i}^{\infty}, r_{2i}^{\infty})\}$ is a fixed point. Hence we have $h_x(r_{1i}^{\infty}, r_{2i}^{\infty}) \leq 0$ and $h_y(r_{1i}^{\infty}, r_{2i}^{\infty}) \leq 0$ for all i, and using that we will show that $\{(r_{1i}^{\infty}, r_{2i}^{\infty})\}$ satisfies the Karush-Kuhn-Tucker (KKT) conditions [13].

Recall that our problem is a convex problem with linear inequality constraints, so the KKT conditions are sufficient for optimality in this case. The first of those conditions is feasibility, of course, but we already know that $\{(r_{1i}^{\infty}, r_{2i}^{\infty})\}$ is a feasible solution (FlowRight always respects feasibility). Then we need only check if for our solution there is a set of Lagrange multipliers with the properties specified by the KKT conditions. Differentiating the Lagrangian for the problem provides us with $2m$ equations:

$$\xi_i g_x(r_{1i}, r_{2i}) + \sum_{l=i}^{m} \lambda_l \xi_i = 0, \qquad 1 \leq i \leq m, \text{and}$$
$$\xi_i g_y(r_{1i}, r_{2i}) + \sum_{l=i}^{2m} \lambda_l \xi_{i-m} = 0, \qquad m+1 \leq i \leq 2m,$$

where λ_l, $l = 1, \ldots, 2m$ are the Lagrange multipliers. Now we need to inquire about the values of these Lagrange multipliers. By subtracting the m^{th} equation from the $(m-1)^{th}$, we obtain $\lambda_{m-1} = -[g_x(r_{1(m-1)}, r_{2(m-1)}) - g_x(r_{1m}, r_{2m})] = -h_x(r_{1(m-1)}, r_{2(m-1)})$. Now, substitute $(r_{1(m-1)}^{\infty}, r_{2(m-1)}^{\infty})$ for $(r_{1(m-1)}, r_{2(m-1)})$, and by what we showed above, we obtain $\lambda_{m-1} \leq 0$. Further, using Lemma 10, $\lambda_{m-1} > 0$ if the $(m-1)^{th}$ constraint is active (i.e. all the bits that are present by that time have been transmitted by the end of epoch $m-1$ on stream 1), and $\lambda_{m-1} = 0$ otherwise. Proceeding this way, we obtain that $\lambda_l > 0$ if the l^{th} constraint is active (i.e., again, all the bits that are present by that time have been transmitted by the end of epoch l on stream 1), and $\lambda_l = 0$ otherwise, for $1 \leq l \leq m$, and $\lambda_l > 0$ if the $(m+l)^{th}$ constraint is active (i.e., causality is met at the end of epoch l on stream 2), and $\lambda_l = 0$ otherwise, for $1 \leq l \leq m$. But with these, we have the KKT conditions completely satisfied. This proves that $\{(r_{1i}^{\infty}, r_{2i}^{\infty})\}$ is the globally optimal solution of our problem. ■

4.2.2 Time-sharing Versus Optimal Multi-Access

As we have pointed out before, one can calculate the best time-sharing offline schedule by running the MoveRight algorithm on the joint sequence of packet arrivals of all users. It is interesting to find out how time-sharing compares to the optimal solution. To that end, in this section we compare the average energies consumed by MoveRight and FlowRight on the same arrival sequence.

The experiment setup is as follows: The two users' packets arrive according to two independent Poisson processes with identical rates, with a combined rate of λ arrivals per unit time (unit time is a symbol time). For each value of λ, 1000 arrivals are generated, and T is set to $t_{1000} + 1/\lambda$. Then, both FlowRight and MoveRight are run on this sequence, and the average energy per packet, and average delay per packet for both users are calculated. Here, $a_1 = 32$ and $a_2 = 1$, these values were chosen because the resulting average energy and delay values are very similar for the two users. In Figure 4.5, the average energy and delay values in both time-sharing and optimal solutions are plotted.

Figure 4.5: Comparison of offline Time-Sharing and Multi-Access schedules as obtained by the MoveRight and FlowRight algorithms for a two-user multi-access channel. The users' packets arrive according to two independent Poisson processes with identical rates, and the combined arrival process is at rate λ. The energy values correspond to 10^3-bit packets. The signaling rate is 10^6 transmissions/sec, and the nominal rate is 6 bits/transmission. That rate is obtained when a packet takes 1 time unit, *i.e.*, $\frac{10^{-2}}{6}$ seconds, to transmit.

4.2.3 Extension to the Broadcast Channel

Consider an AWGN broadcast channel with one sender and two receivers. The sender has two streams of packet arrivals, with one stream destined to each receiver. As before (see Figure 4.1) we merge the packets into a single sequence.

The received signal at the ith receiver at time k is given by:

$$Y_i[k] = \sqrt{s_i} X[k] + Z_i[k], \tag{4.3}$$

where $X[k]$ is the transmitted signal with average power constraint P, $\sqrt{s_i}$ is the channel gain and the $Z_i[k]$'s are i.i.d. zero-mean Gaussian noise with variance σ^2. The capacity region of the channel (see [23]) assuming $s_1 < s_2$, is the set of rate pairs (r_1, r_2) such that

$$\begin{aligned} r_1 &\leq \frac{1}{2}\log_2\left(1 + \frac{\alpha s_1 P}{\sigma^2}\right) \\ r_2 &\leq \frac{1}{2}\log_2\left(1 + \frac{(1-\alpha) s_2 P}{\alpha s_2 P + \sigma^2}\right), \end{aligned}$$

for some $0 \leq \alpha \leq 1$. Now, we express the minimum average power for a given rate pair, where the above inequalities are replaced by equalities, as follows. Rewrite the first equality as $\alpha s_1 P/\sigma^2 = 2^{2r_1} - 1 = f(r_1)$. Hence, $\alpha = (\sigma^2/P s_1) f(r_1)$. Substituting into the second inequality and rearranging we obtain

$$P = \sigma^2 \left(\frac{f(r_1)}{s_1} + \frac{f(r_2)}{s_2} + \frac{f(r_1) f(r_2)}{s_1} \right).$$

Consider epochs defined in the same way as in section 4.2, as the packet inter-arrival times of the merged sequence. Again, making the observation that in an optimal broadcast schedule rates do not need to change during an epoch,the offline scheduling problem is as follows.

Problem 4 : Broadcast channel offline scheduling

$$\begin{aligned}
&\textit{Minimize:} && \sum_{i=1}^{m} \xi_i \left(\frac{f(r_1)}{s_1} + \frac{f(r_2)}{s_2} + \frac{f(r_1)f(r_2)}{s_1} \right) \\
&\textit{subject to:} && \sum_{i=1}^{k} r_{ji}\xi_i \leq \sum_{i=1}^{k} c_{ji}, \;\; k = 1, \ldots, m-1, \;\; j = 1, 2 \\
& && \sum_{i=1}^{m} r_{ji}\xi_i = \sum_{i=1}^{m} c_{ji}, \;\; j = 1, 2.
\end{aligned}$$

Note that this is the same as Problem 3 in Section 4.2, except that the objective function is different. But the objective function is still convex, monotonically increasing and differentiable in both r_1 and r_2. Also, it is expressed as a sum of epoch energies. Hence:

Theorem 6 F*lowRight finds the unique solution to Problem 4.*

4.3 Scheduling over Slow Fading Channels

Consider the AWGN channel as specified by Equation (4.2). We make the block-fading assumption where the power gain $s_i[k]$ changes every T_c channel uses (a“coherence window”). Further assume that fading is slow with respect to codeword lengths. Initially, consider the single user case, *i.e.*, $k = 1$. We assume that both the transmitter and the receiver have perfect channel state information at the beginning of each coherence window.

As before, consider packets coming at arbitrary instants in $[0, T)$, all of which need to be transmitted within this same time period. The optimal offline schedule is the one that minimizes the total packet transmission energy given perfect knowledge of the packet arrival instants and channel state values for the entire duration $[0, T)$, at time 0.

Define an “epoch” to be a time interval that begins with either a packet arrival or a change in the channel state, and continues until the next arrival or state change. The first epoch starts at $t_1 = 0$, and continues until t_2 or T_c, whichever is smaller, at which point the second epoch starts, and so forth. Let c_j denote the number of bits that have arrived at the beginning of epoch j, so $c_j > 0$ if the j^{th} epoch starts with a packet arrival, and $c_j = 0$ otherwise. Let the duration of epoch j be ξ_j.

Lemma 11 I*n an optimal schedule, rate is constant during an epoch.*

Proof: Noting that the channel state and the number of available bits are constant during an epoch by definition, the proof follows very similarly to the proof of Lemma 9. Suppose rate is r_1 in the first τ_1 time units of an epoch of length t, and r_2 during the remaining $t-\tau_1$. The transmit energy in this epoch is then $\tau_1 f(r_1)/s + (t-\tau_1)f(r_2)/s$ where s is the fading state during the epoch. The same number of bits can also be transmitted using the uniform rate $(r_1\tau_1 + r_2(t-\tau_1))/t$ for the whole time t. This new rate results in a total energy $tf((r_1\tau_1 + r_2(t-\tau_1))/t)/s$, which, by convexity of f, is strictly lower than previous, unless $r_1 = r_2$. ■

From Lemma 11, $\{r_j\}_{j=1}^m$ (m is the number of epochs) is sufficient to characterize the optimal schedule:

Problem 5 : Offline scheduling for the slow fading channel

$$\begin{aligned} \textit{Minimize:} \quad & \sum_{i=1}^{m} \xi_i f(r_j)/s_j \\ \textit{subject to:} \quad & \sum_{i=1}^{k} r_i \xi_i \leq \sum_{i=1}^{k} c_i \quad k = 1, \ldots, m \\ & \sum_{i=1}^{m} r_i \xi_i = \sum_{i=1}^{m} c_i. \end{aligned}$$

This convex optimization problem can be solved by the FlowRight algorithm: Initially, the rates are set to $r_j^0 = c_j/\xi_i$, $i = 1, 2, \ldots, m$. The first two epochs are then considered. The total number of bits transmitted in these two data epochs is $r_1^0\xi_1 + r_2^0\xi_2$. Keeping the total number of bits fixed, r_1^0 is updated to r_1^1, the value that minimizes the total energy of the first pair of data epochs. Note that $r_1^1 \leq r_1^0$, since from their initial condition information can only be moved to the right (otherwise causality would be violated.) Therefore r_2^0 is reset to a new value that is larger than (or equal to) its initial value. Moving to the second pair of epochs, this time r_2^0 is optimally decreased to r_2^1, and the value of r_3^0 is reset. Proceeding in this way we obtain r_i^1 for $i = 1, \ldots, n$. This completes the first *pass* of the algorithm. The algorithm then repeats the same procedure and terminates after K passes, where $K = \min\{k : |r_i^k - r_i^{k-1}| < \epsilon\}$, $i = 1, \ldots m$, for small enough $\epsilon > 0$.

Theorem 7 T*he following statements hold.*

1. *As the algorithm FlowRight runs on $\{r_i\}$, information always flows to the right.*

2. *FlowRight stops, and returns a sequence $\{r_i^\infty\}$.*

3. *$\{r_i^\infty\} = \{r_i^{opt}\}$.*

The only difference between this theorem and Theorem 5 is that the energy function, due to scaling with channel gain, does not have the same form for each epoch pair: $h(r_i) = \xi_i f(r_i)/s_i + \xi_{i+1} f(r_{i+1})/s_{i+1}$. Note, however, that this does not affect any of the steps of the proof of Theorem 5, and therefore the proof of this theorem follows from the proof of Theorem 5.

4.3.1 Extension to Multi-access and Broadcast Channels with Fading

Now we extend the slow fading single user offline scheduling results to multiple-access and broadcast channels. To formulate the offline scheduling problem we first merge all users' packet arrival sequences and the times at which channel states change to obtain m epochs and note as before that in an optimal schedule rates do not need to change during an epoch.

Problem 6 : Offline scheduling for the slow fading multi-access channel

$$
\begin{aligned}
\textit{Minimize:} \quad & \sum_{i=1}^{m} \xi_i \left((\frac{1}{s_{k_1 i}} - \frac{1}{s_{k_2 i}}) f(r_{k_1 i}) + \frac{1}{s_{k_2 i}} f(r_{k_1 i} + r_{k_2 i}) \right) \\
\textit{subject to:} \quad & \sum_{i=1}^{k} r_{ji}\xi_i \leq \sum_{i=1}^{k} c_{ji} \quad k = 1, \ldots, m-1, \;\; j = 1, 2 \\
& \sum_{i=1}^{m} r_{ji}\xi_i = \sum_{i=1}^{m} c_{ji} \quad j = 1, 2,
\end{aligned}
$$

where $k_1 i = \arg\min_{k \in \{1,2\}}(s_{1i}, s_{2i})$ and $k_2 i = \{1, 2\} - \{k_1 i\}$, and where $c_{ji} = B$ if a packet for user j arrives at the beginning of epoch i, and 0 otherwise. This is a convex optimization problem with linear constraints and can be solved by FlowRight.

Note that the broadcast scheduling problem in the slow fading channel can be stated and solved similarly.

4.3.2 Fast Fading Channels

Now, suppose that fading is fast with respect to our codeword lengths, so that a codeword will experience many channel realizations. In this setting it is natural to assume that the transmitter does not have CSI; by the time feedback from the receiver about channel state reaches the receiver, the channel has already changed. It is well known (see, *e.g.*[14]) that the ergodic capacity of this (single-user) Gaussian fading channel is given by:

$$
\mathrm{E}_{\mathbf{s}} \left(\frac{1}{2} \log(1 + sP) \right) \triangleq C(P) \tag{4.4}
$$

where $\mathrm{E}_{\mathbf{s}}$ denotes expectation over the channel state $\mathbf{s}$, and P is the average transmit power.

Notice that $C(P)$ is a monotonically increasing and concave function in P. Hence, it is invertible, with inverse $C^{-1}(r)$ monotonically increasing and convex in r. Reliable communication at rate r is possible if received power is in the set: $\{P : C(P) \geq r\} = \{P : P \geq C^{-1}(r)\}$.

Therefore, the power needed to communicate at rate r is $C^{-1}(r)$. This problem can be written in the style of Problem 5, by defining epochs as packet inter-arrival times, of durations ξ_i, $1 \leq i \leq m$.

Problem 7 : Offline scheduling for the fast fading channel

$$\begin{aligned} \textit{Minimize:} \quad & \sum_{i=1}^{m} \xi_i C^{-1}(r_i) \\ \textit{subject to:} \quad & \sum_{i=1}^{k} r_i \xi_i \leq k\ B\ ,\ \ k = 1, \ldots, m-1 \\ & \sum_{i=1}^{m} r_i \xi_i = m\ B. \end{aligned}$$

Note that the definition of Problem 7 does not involve channel states. This is because the transmitter does not track the channel state, which is assumed to vary over a codeword resulting in channel capacity to be constant (Equation (4.4)). This is unlike the slow-fading case where we assumed the transmitter can obtain channel state information and code accordingly, and thus the capacity changes in time. Clearly this problem can be solved using FlowRight. Note, however, that it is much simpler than Problem 5, since we do not need to consider channel state change instants when defining epochs. In fact, this problem is identical to Problem 1 and thus its closed-form solution is given by (4.1).

4.4 Inclusion of a Constant Power Component

It was pointed out in Chapter 3 that the total power consumed by a tranceiver in practice may be modeled as the sum of transmit power and a constant power term. So far in this chapter we focused on transmisson power only. It is interesting to explore how the results of this chapter would change if the power function was changed to $f(r) + c$, to include a constant term c.

Lemma 8 changes significantly. In the symmetric case, there is a unique time-sharing point that achieves minimum energy and multiaccess is strictly sub-optimal. It is simple to see this: The objective now is to minimize $a\tau(P_1 + c) + a\tau(P_2 + c)$ with multiaccess, and $a\tau P_1 + a\tau P_2 + a\tau c$ with time-sharing. The latter is clearly smaller. In the asymmetric case, the objective is to minimize $a_1\tau(P_1 + c) + a_2\tau(P_2 + c)$ with multiaccess, and $a_1\tau P_1 + a_2\tau P_2 + a_1\alpha\tau c + a_2(1-\alpha)\tau c$, for $0 \le \alpha \le 1$ with time-sharing. Whether time-sharing or multiaccess is optimal depends on the values of the constants a_1, a_2, c, r_1 and r_2.

4.5 Conclusion

In this chapter, we extended the formulation in Chapter 3 to transmission scenarios where the channel is time-varying due to interference and/or fading. We showed that offline scheduling for classes of multi-access and broadcast channels with and without fading can be reduced to convex optimization problems with linear constraints and devised an iterative algorithm, FlowRight, that finds the optimal offline schedule.

A host of model variations motivated by practical networking situations are solved readily with FlowRight. To illustrate, consider the following two cases: (a) each packet has its individual latency constraint, (b) buffer size is finite, and packet drops are not allowed. First, note that case (b) reduces to case (a): Suppose buffer size is A. This means, packet i must be transmitted before packet $A+i$ arrives. So t_{A+i} is the transmission deadline for packet i, and $t_{A+i} - t_i$ is its latency constraint. Now place epoch end-points at all these packet deadlines. Suppose, for example, packet i's deadline marks the end of epoch k_i. Then, the following linear inequality constraints are obtained: $\sum_{j=1}^{k_i} \xi_j r_j \ge iB$, for all i.

Chapter 5

Online Adaptation to Queue and Channel

Consider the following generic data communication scenario: the transmitter and receiver engage in a session which may be a video conference or a web session, or may alternate between the two. Different types of applications generate data at different rates, and the data packets are collected in the transmitter's buffer to be sent to the receiver. Let the rate at which packets arrive into the transmitter's buffer at time t be $\lambda(t)$ packets/second. They are transmitted to the receiver at a rate $\mu(t)$ packets/second. Assume we set $\mu(t) = \mu$, a constant that is large enough, say, for a high rate streaming video session. When the required rate drops, for example because the user switches to a lower-rate web session where $\lambda(t) \ll \mu$, the transmitter will idle a significant fraction of time and unnecessarily transmit at a high rate for the rest of the time.

Schemes that adapt solely to the channel state can maximize the throughput for a given energy constraint; but since they cannot track the value of $\lambda(t)$, they do not have control over delay. In order to guarantee finite average delay, they need to be set for the largest possible value of $\lambda(t)$, which causes them to be energy-inefficient. A better performance in terms of energy and delay can be obtained through adaptation to the variation in the

Figure 5.1: An illustration of look-ahead scheduling

data rate. It was shown in the previous chapters via offline analysis that such adaptation can result in significant transmission energy savings. This chapter will focus on online scheduling.

As opposed to previously, algorithms here are heuristic. They are based on the idea of a look-ahead buffer, which, while being very simple to implement, also performs very competitively to optimal. For the fading channel, the Look-ahead Water-filling algorithm will be presented. This is a heuristic for adapting jointly to the channel state and data arrival rate. It will be observed to be significantly more energy-efficient than the well-known Water-filling algorithm that adapts solely to the channel state, and to perform closely to the offline optimal benchmark provided by FlowRight. The results generalize to multi-access and broadcast channels.

5.1 Online Scheduling

A way to carry the offline scheduling algorithms to the online setting is to use a *look-ahead* buffer. If it was possible to look ahead into the future of the arrival process for a certain duration, offline algorithms could be used to schedule optimally over that duration. This idea can be mimicked in the real-time setting by observing the arrival process in a certain time window, buffering all packets that arrive, and scheduling them optimally at the end of this time window.

Let's think of having a switch between the buffer and the scheduler which closes at certain times (Figure 5.1). The case when the switch closes every L time units, will be called *fixed look-ahead*. In this case, packets arriving in $[0, L)$ are scheduled for departure

Figure 5.2: The fixed look-ahead algorithm operating on packets with different energy functions. Packet arrival instants are marked by crosses. Colored bars depict packet transmission durations.

in the interval $[L, 2L)$, and so on, as illustrated in Figure 5.1.[1] The case when the switch closes after the completion of each packet transmission will be called *variable look-ahead*. In this case, packets in the buffer are optimally scheduled for the next L time units, but only the first packet is transmitted, after which packets are scheduled again.

More precisely, the variable look-ahead algorithm sets packet transmission duration to $q(t)/L$ where $q(t)$ is the queue size at time t, equivalently, sets transmission rate to $q(t)/L$ packets/sec. We further modify the algorithm to set the rate as $\min(q(t)/L, \lambda_{\max})$, if it is known that packet arrival rate is upper bounded by $lambda_{\max}$. Clearly the larger the choice of L, the better the online algorithm can track the arrival process, and thus the more competitive it will be to the offline optimal in terms of energy per packet. On the other hand, packet delay will increase with L, roughly linearly as seen in Figure 5.3. A useful property of look-ahead scheduling is stability under any arrival rate λ. This is obvious for fixed look-ahead, as the queue is emptied every L time units. The stability of variable look-ahead is noted below.

Lemma 12 *The variable look-ahead algorithm, which sets transmission rate to* $\min(q(t)/L, \lambda_{\max})$, *is stable under any arrival process of rate* $\lambda < \lambda_{\max}$.

*P*roof. Suppose the claim is false. Consider the Markov process where the state at time t_n, $n = 0, 1, 2, \ldots$, is $q(t_n) > 0$, and state transitions occur whenever there is a packet arrival or departure. Let $q(t_0) = 0$. The state evolves as follows:

$$q(t_{n+1}) = q(t_n) + \text{arrivals in } (t_n, t_{n+1}) - \text{departures in } (t_n, t_{n+1}) \tag{5.1}$$

[1] In this example, packets have different energy functions hence they are allocated different transmission durations.

Since $q(t_0)$ is finite and t_n increases at most linearly with n, the expectation $\mathrm{E}(t_n)$ is defined for every n. The expected[2] number of arrivals in (t_n, t_{n+1}) is λ, and the expected number of departures is $\mathrm{E}(q(t_n)/L \mid q(t_n)/L < \lambda_{\max})\mathrm{Pr}(q(t_n)/L < \lambda_{\max}) + \lambda_{\max}\mathrm{Pr}(q(t_n)/L \geq \lambda_{\max})$. Summing both sides of Equation (5.1) over n, and telescoping, we obtain

$$\begin{aligned} \frac{1}{N+1}\mathrm{E}(q(t_N)) = \lambda \quad &- \quad \frac{1}{N+1}\sum_{n=0}^{N}\mathrm{E}(q(t_n)/L \mid q(t_n)/L < \lambda_{\max})\mathrm{Pr}(q(t_n)/L < \lambda_{\max}) \\ &+ \quad \frac{\lambda_{\max}}{N+1}\sum_{n=0}^{N}\mathrm{Pr}(q(t_n)/L \geq \lambda_{\max}). \end{aligned} \tag{5.2}$$

Our hypothesis that the queue is not stable implies that the Markov chain is transient, hence eventually any finite set of states has probability zero. But then the limiting probability of the event $\{q(t_n)/L < \lambda_{\max}\}$ is zero. Using this fact in Equation (5.2),

$$\lim_{N\to\infty} E(q(t_N)) = \lambda - \lambda_{\max},$$

which is negative. This is a contradiction. ∎

Figure 5.3: Variation of energy and delay per packet for fixed and variable look-ahead schemes as the size of the look-ahead window (L) increases. Packets have either of the two energy functions $10^4/6\tau(2^{12/\tau} - 1)$ and $16 \times 10^4/6\tau(2^{12/\tau} - 1)$. Packet generation process is Poisson at rate $\lambda = 0.6$, the duration is $T = 10000$. The optimal offline schedule gives energy 2.5×10^6, and delay of 37.56.

5.2 Multiuser Online Scheduling

It is straightforward to apply look-ahead algorithms in the multiuser case: Observe all users' arrival processes for the same duration, and at the end of the time period schedule these

[2]Here the expectation is over packet arrival statistics. When proving the stability of the look-ahead water-filling algorithm, the expectation will be over the statistics of both the packet arrival process and the channel state.

packets jointly.

Consider the uplink. One way to schedule the packets in the look-ahead buffer is to time-share between users. The best time sharing schedule can be obtained by running MoveRight as discussed in Chapter 4. Figure 5.4 plots the performance of look-ahead with time-sharing and that of look-ahead with multiuser coding (obtained by running FlowRight on the look-ahead buffer). The experiment setup is similar to the one in Section 4.2.2 in Chapter 4. The look-ahead window size was held at $L = 25$ time units. Note, from Figure 5.3 that a window of 20-30 time units is a good choice, in the sense that most of the achievable decrease in energy is obtained. Because of this choice of L, as the online algorithms "adapt" to the arrival rate, average delay remains around 25 time units. The resulting average energy per packet is reasonably close to the optimal, also plotted in Fig. 5.4. Consider packet delay: The optimal schedule has lower delay that varies with λ, while the delay incurred by the online algorithms are almost constant and a little higher than L, as expected. This points to the nice property of the look-ahead heuristic that by incurring some predictable delay, it competes very well with the offline optimal in terms of energy.

Another point of interest to note from Figure 5.4 is that the performance difference of online time-sharing and online multiuser coding is small at low arrival rates. As the system gets more congested, multiuser coding becomes more advantegeous. Finally, downlink scheduling can be similarly done by utilizing FlowRight.

Figure 5.4: Comparison of the online algorithms Look-Ahead Time-Sharing and Look-Ahead Multi-Access for a two-user multi-access channel. 10^3-bit packets arrive according to two independent Poisson processes with identical rates. $L = 25$, the signaling rate is 10^6 transmissions/sec, and the nominal rate is 6 bits/transmission achieved when a packet takes 1 time unit, *i.e.*, $\frac{10^{-2}}{6}$ seconds, to transmit.

5.3 Online Scheduling in the Slow Fading Channel

Assume that the packet input process into the transmitter buffer is stationary and ergodic. The time average arrival rate $\lambda \triangleq \lim_{T\to\infty} \frac{1}{T}\int_0^T \lambda(t)\mathrm{d}t$ is bounded such that $\lambda < \lambda_{\max}$ with probability 1. We are interested in schedules that are stable, *i.e.*, scheduling algorithms that ensure that the number of packets in the buffer is finite with probability 1.

Future arrivals, channel states, or λ are not known. The channel has slow ergodic fading with known statistics where the power gains of different coherence windows are independent and identically distributed. The transmitter knows the present value of the channel gain just before transmitting a packet. The bound on packet arrival rate, $\lambda_{\max}$, is also known.

A natural candidate to compare look-ahead online scheduling against is an algorithm that adapts to the fading state but not the arrival process. As described in Chapter 2, optimal power/rate adaptation to the single-user ergodic fading channel with perfect transmitter and receiver CSI is *water-filling in time*. Below, this algorithm is adapted to the present setting. Next, Look-ahead Water-filling, an algorithm that simultaneously adapts to both the channel and the data arrival rate, is described.

5.3.1 Water-filling in Time

The power control policy that minimizes the long term average power while achieving a target rate R^* is water-filling on the fading states [69]:

$$P(s) = \begin{cases} N(\frac{1}{s_o} - \frac{1}{s}), & \text{if } s \geq s_o \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

where $f_S(s)$ is the probability density function of the channel gain S, and s_o is the solution of

$$\frac{1}{2}\int_{s_o}^{\infty} \log_2\left(\frac{s}{s_o}\right) f_S(s)ds = R^*$$

Figure 5.5: Water-filling algorithm (WF) for adapting to channel state.

The rate adaptation is then

$$R(s) = \frac{1}{2}\log_2(1 + \frac{sP(s)}{N}) \tag{5.4}$$

To use water-filling (WF) adaptation in the present setting, the average rate is set equal to $R^* = \lambda_{\max}$ (pkts/time unit) $\times$ B(bits/pkt) $\times$ T_s (time units/symbol) to ensure stability. Then s_o is computed (hence determining the long term average power) such that the capacity is equal to this target average rate. Before each packet transmission, the instantaneous power and rate are set according to the channel gain s, which is assumed constant during packet transmission. The algorithm is illustrated with a flowchart in Figure 5.3.1.

5.3.2 Look-ahead Water-filling Algorithm

While the water-filling scheduling algorithm optimally adapts to the channel state in the sense of minimizing power for a given target rate $\lambda_{\max}$, this algorithm will clearly be wasteful if the packet arrival rate decreases below $\lambda_{\max}$. Now we describe an online algorithm, the Look-ahead Water-filling algorithm (LW), that adapts to the arrival rate as well as to the channel.

The algorithm is as follows: suppose just before time t, a packet transmission ended. Let the backlog at time t be $Q(t)$. If $Q(t) > 0$, begin transmitting the packet at the head of the queue at time t (otherwise, wait until there is a packet in the queue). Set the target transmission rate to $\hat{\mu} = \min\{Q(t)/L, \lambda_{\max}\}$ packets/time unit for some constant $L > 0$. Given $\hat{\mu}$, determine the instantaneous transmission rate according to water-filling. That is, the optimal cutoff value s_o is computed as in section 5.3.1, which corresponds to an average power for which the capacity is $BT_s\hat{\mu}$. The instantaneous power and code rate are then determined from (5.3) and (5.4). The algorithm is summarized in Figure 5.3.2.

Figure 5.6: Look-ahead Waterfilling (LW) Algorithm for adapting jointly to buffer and channel states

In the LW algorithm, the target packet transmission rate $\hat{\mu}$ never exceeds $\lambda_{\max}$, yet the queue is stable. This is noted in the following lemma (for the proof, see the proof of Lemma 12.)

Lemma 13 T*he LW algorithm is stable,* i.e., *given any t, with probability one there exists t_1, $t < t_1 < \infty$, such that $Q(t_1) = 0$.*

Simulation experiments were conducted to observe the performance of LW. The setting is as follows: 1 Kbit packets arrive at the buffer at a rate $\lambda < 1$ arrivals/time unit. A time unit is 1/6 msec, which corresponds to the transmission duration of a packet if it is transmitted at $r = 6$ bits/symbol (symbol rate is constant at 10^6 symbols/sec). The packet arrival process is a Markov Modulated Poisson Process for which $\lambda(t) = \beta\lambda$ with probability $0.9/\beta$, and $\lambda(t) = \frac{1}{10-9/\beta}\lambda$ otherwise. The parameter $\beta \geq 1$ is chosen such that the process is ergodic with expected rate λ. Note that when $\beta > 1$, the arrival process is bursty, and for $\beta = 1$ it reduces to a Poisson process at rate λ.

Figure 5.7: The top plot shows a sequence of packet arrivals ("$\times$") and channel gains. The lower three plots show the instantaneous rates used by the online algorithms Water-filling and Look-ahead Water-filling, and the Optimal Offline Schedule, respectively, as they run on this sequence of packet arrivals. The average energy per packet values are normalized for a noise power of unity.

Figure 5.7 shows an example run of bursty packet arrivals at $\lambda \simeq 0.5$, scheduled by the three algorithms WF (Water-Filling), LW (Look-ahead Water-filling) and OPT (Optimal Offline). Notice that WF transmits with much higher rate than the other two algorithms, thus quickly finishes its backlog and idles a significant amount of the time. LW, on the other hand, spreads its rate more uniformly over time, almost as uniformly as OPT which

Figure 5.8: Energy per packet as arrival rate changes for $\beta = 1$ in Rayleigh fading; $L = 25$.

has the lowest rate transmission.

Figures 5.8 and 5.9 explore the energy and delay performance of these algorithms. The water-filling schedule has constant energy for all arrival rates, since the rate it assigns to packets is independent of λ. This energy is much higher than the average energy values achieved by LW when λ is small; both for bursty and non-bursty arrival processes. Of course the energy efficiency is achieved at the expense of an increase in delay. The delay of LW is essentially lower bounded by L, as it allows this time to monitor the arrival process. However, as can be observed from Figure 5.8, the variation of its delay is much smaller than that of WF. In the figure, the delay of WF varies by about 7000% as λ is varied from 0 to $\lambda_{\max} = 1$, while the delay of LW varies only about 60%. The fact that the *delay jitter* is so much smaller makes the backlog-adaptive algorithm attractive for data applications, especially streaming media.

Figure 5.9: Average energy per packet as arrival rate changes for $\beta = 2$ in Rayleigh fading; $L = 25$.

5.3.3 Extension to Multi-access and Broadcast Channels with Fading

In the uplink, if the fading processes of users are i.i.d., and the goal is to maximize the sum rate with respect to a total power constraint, the important result of Knopp and Humblet [55] says that the optimal power control scheme allows only the user with the best channel to transmit at any given time. The rate of that user is then determined by water-filling across the channel states. Tse and Hanly [69] exhibit the optimal power control when users are not necessarily symmetric, and the goal is to maximize a weighted sum of the rates. They present an online "Greedy Algorithm", shown to be optimal under transmitter CSI.

A “look-ahead greedy” online schedule that uses a look-ahead buffer to adapt to both the backlog and the channel state (similar to Look-ahead Water-filling) can be obtained as follows: Each user’s required rate is estimated from the current backlogs. The power allocation is then determined using the Greedy algorithm in [69]. Finally, in the downlink case, online scheduling is analogous.

5.4 Conclusion

This chapter presented an online heuristic for adapting to data generation rate. The heuristic uses a simple look-ahead buffer. For the fading channel, the look-ahead idea has been augmented with a water-filling power control policy, to obtain algorithm LW, which adapts to the fading state as well as the buffer state. Despite their simplicity, these algorithms have experimentally been seen to perform very competitively to optimal offline schedules. The energy consumed is orders of magnitude less than a well-known benchmark which adapts to fading state only. Another apparent advantage of look-ahead-based schedules is the predictable delay: the delay is almost constant around the look-ahead buffer size, and does not vary with the packet arrival rate λ. Results were observed to hold under different arrival process statistics (including bursty ones). We believe that the energy efficiency combined with the predictable, jitter-free delay is attractive for many wireless data applications, especially multimedia.

Chapter 6

Conclusions and Future Directions

This chapter will summarize this work and outline directions for further research.

6.1 Summary

The goals of this work were essentially twofold: first, understanding how efficiently one can communicate discontinuously generated data, and second, finding online data transmission algorithms that achieve close to optimal energy-efficiency. The first goal was achieved through the development of a theoretical model that captures essential elements such as the arrival of messages at arbitrary times. The second goal was achieved by developing (heuristic) online algorithms that are suitable for easy implementation, while being robust to data generation.

The centerpiece of the model was the assumption that the energy to transmit a packet is a convex, decreasing function of the transmission duration. The model and the analysis is quite general in that they apply to any coding scheme and channel model that gives rise to an energy function that satisfies those properties. Our methodology was offline analysis. Offline analysis provided an understanding of optimal scheduling, as well as benchmarks for the performance of practical (online) schedules.

The strength of this model is in its ability to handle various situations that may arise in a wireless network: The single transmitter-single receiver optimal offline scheduling problem was solved for both finite time-horizon and infinite time horizons. The uplink and downlink scheduling problems were solved by the iterative FlowRight algorithm. FlowRight readily provided the optimal offline solution to single and multiuser problems under channel fading. Variations such as each packet having individual deadlines, finite buffer size, etc., were handled straightforwardly.

Online scheduling algorithms were developed using a heuristic device called a "look-ahead buffer". By incurring a constant, predictable delay, energy close to optimal was attained. The online algorithms were seen to be very competitive to offline optimal such that there is little room for improvement. Searching for optimal online algorithms therefore seems to be of little practical value. Moreover, the simplicity of look-ahead scheduling in addition to the predictable, constant delay is very attractive for modern wireless data applications.

6.2 Future Directions

There are many possible directions for further work in this area. Those that are immediate extensions of this work were mentioned in the previous chapters. Here, we will discuss some others. First of all, this work focused on transmission energy. As noted, incorporating other components into the models is straightforward, but has not been attempted in this thesis since the relative importance of the energy components is highly system-dependent. We submit that such extensions could easily be done for specific applications. In fact, implementation of energy-efficient packet scheduling algorithms in real-life networks would be very interesting. Developers of sensor networks and other energy-constrained networks are very interested in doing just that, as more adaptive hardware becomes available.

An important direction for future work is generalizing the models to the multi-hop network setting. The design of energy-efficient network protocols can benefit greatly from the understanding that will result from such modeling and analysis.

This work only scratched the surface of the energy-delay tradeoff which is fundamental to energy-constrained networks. Exploring the throughput/delay/energy space further would be very interesting. For example, throughput limits in mobile networks under energy and delay constraints is an important question. Also interesting is the notion of using energy as a metric in routing topology establishment in ad-hoc networks.

Bibliography

[1] P. Abshire, A. G. Andreou, “Relating Information Capacity to a Biophysical Model for Blowfly Photoreceptors”, in *Neurocomputing 32-33* (2000) 9-16.

[2] F. Adler, “Minimum energy cost of an observation”, *IRE Transactions- Information Theory*, pp. 28 - 32, 1955.

[3] S. Appadwedula, M. Goel, N. Shanbhag, D. Jones, and K. Ramchandran, ”Total System Energy Minimization for Wireless Image Transmission”, Journal of VLSI Signal Processing Systems, vol. 27, no. 1/2, pp. 99–117, February 2001.

[4] B. Ata, J.M. Harrison and L.A. Shepp, ”Drift Rate Control of a Brownian Processing System,” preprint, 2002.

[5] B. Ata, *Dynamic Control of Stochastic Networks: Heavy Traffic Approximations and Asymptotic Optimality*, Ph.D. thesis, Graduate School of Business, Stanford University, 2003.

[6] N. Bambos and G. Pottie, “Power control based admission policies in cellular radio networks”, *GLOBECOM ’92*

[7] N. Bambos, “Towards Power-Sensitive Network Architectures in Wireless Communications”, *IEEE Personal Communications*, vol. 5, no. 3, 1998.

[8] N. Bambos, S. C. Chen, G. J. Pottie, “Channel Access Algorithms with Active Link Protection for Wireless Communication Networks with Power Control”, in *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 583-597, 2000.

[9] N. Bambos and S. Kandukuri, “Power Control Multiple Access (PCMA) in Wireless Communication Networks”, in *Proc. IEEE INFOCOM*, 2000, pp. 386-395.

[10] S. Bansal, R.Gupta, R. Shorey, I. Ali, A. Razdan, A. Misra, “Energy efficiency and throughput for TCP traffic in multi-hop wireless networks,” in *Proc. IEEE INFOCOM 2002*, vol. 1, pp: 210 -219, 2002.

[11] T. Berger, “Living Information Theory”, *“Shannon Lecture”*, presented at IEEE Int. Symp. Information Theory, Lausanne, Switzerland, July 2002.

[12] R. A. Berry, *Power and Delay Trade-offs in Fading Channels*, Ph.D. Thesis, MIT, Cambridge, MA, December 2000.

[13] D. P. Bertsekas. *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1995.

[14] E. Biglieri, J. Proakis and S. Shamai (Shitz). “Fading Channels: Information-Theoretic and Communications Aspects,” *IEEE Transactions on Information Theory*, vol. 44, No. 6, October 1998: pp. 2619-2692.

[15] G. Caire and S. Shamai (Shitz), “On the capacity of some channels with channel state information”, in *Proc. 1998 IEEE Int. Symp. Information Theory*, (Cambridge, MA, Aug. 17-21, 1998), p. 42.

[16] C-T. Chou, and K.G. Shin, ”Analysis of Combined Adaptive Banwidth Allocation and Admission Control in Wireless Networks”, in *Proc.IEEE INFOCOM’02*, New York City, New York, June 2002.

[17] G. Caire, G. Taricco, and E. Biglieri, “Minimum information outage rate for slowly fading channels”, in *Proc. IEEE Int. Symposium on Information Theory*, (Cambridge, MA ,Aug. 17-21, 1998), p. 7.

[18] G. Caire, G. Taricco, and E. Biglieri, "Optimum Power Control over Fading Channels", in *IEEE Trans. on Information Theory*, vol. 45, no. 5, pp. 1468-1489, July 1999.

[19] A. Cerpa, D. Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies" in *Proc. IEEE INFOCOM 2002*, vol. 3, pp.: 1278 -1287, 2002.

[20] A. Chockalingam, M. Zorzi, "Energy Efficiency of Media Access Protocols for Mobile Data Networks," *IEEE Trans. Comm.*, vol. 46, pp. 1418-21, 1998.

[21] T. Coleman and M. Medard, "Trade-off Between Energy and Delay in Wireless Packetized Systems", presented at the Thirty-Ninth Annual Allerton Conference on Communication, Control, and Computing, 2001.

[22] B. Collins and R. Cruz. "Transmission Policies for Time Varying Channels with Average Delay Constraints", in *Proc. 1999 Allerton Conf. on Commun., Control & Comp.*, Monticello, IL, 1999.

[23] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons, Inc., 1991.

[24] R. Durrett, *Probability: Theory and Examples*, Duxbury Press, 1996.

[25] A. El Gamal and T. M. Cover, "Multiple User Information Theory", in *Proc. IEEE*, vol. 68, no. 12, pp. 1466-1483, Dec. 1980.

[26] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks", in *Proc. IEEE INFOCOM 2002*, 2002, pp. 1773-1782.

[27] A. Ephremides and B. Hajek, "Information theory and communication networks: an unconsummated union," in *IEEE Trans. Info. Theory*, 44(6): 2416-2434, 1998.

[28] E. Erkip, X. Lu, Y. Wang and D. Goodman, "Total Power Optimization for Wireless Multimedia Communication", Presented in IMPACCT 2002, Integrated Management of

Power Aware Communications, Computing and Networking Workshop, in conjunction with ICC 2002, New York, New York, May 2002.

[29] L. M. Feeney and M. Nilsson, "Investigating the Energy consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", *IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[30] P. Fleming and A. Stolyar, "Convergence properties of the CDMA reverse link power control algorithm", 1994.

[31] G.J. Foschini, "A Simple Distributed Autonomous Power Control Algorithm and its Convergence", *IEEE Trans. on Vehicular Technology*, vol. 42, no. 4, 1993.

[32] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence" in *IEEE Trans. Vehic. Technol.*, vol. 40, no. 4. pp. 641–646, 1993.

[33] A. Fu, E. Modiano, and J. Tsitsiklis, "Optimal Energy Allocation and Admission Control for Communications Satellites", *IEEE INFOCOM*, New York, NY, June 2002.

[34] R. Gallager, "A perspective on multiaccess channels," in *IEEE Trans. Info. Theory*, vol. 31, no. 2, pp. 124-142, 1985.

[35] R.G. Gallager, *Energy Limited Channels: Coding, Multiaccess and Spread Spectrum*, M.I.T. LIDS Report LIDS-P-1714, November 1987.

[36] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, Boston, 1995.

[37] D. Ganesan, R. Govindan, S. Shenkar, and D. Estrin, "Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks," in *Mobile Computing and Communications Review*, vol. 4, October 2001.

[38] K.S. Gilhausen, I.M. Jacobs, R. Padovani, A.J. Viterbi, L.A. Weaver, and C.E Wheatley, “On the Capacity of a Cellular CDMA System”, in *IEEE Transactions on Vehicular Technology*, vol. 40, no. 2, pp. 303-311, 1991.

[39] S. Giordano, W. W. Lu, “Guest editorial - Challenges in mobile Ad Hoc networking”, in *IEEE Communications Magazine*, no. 6, June 2001 pp. 129-129.

[40] S. Gitzenis and N. Bambos, ”Power-controlled Data Prefetching/Caching in Wireless Packet Networks”, *IEEE INFOCOM*, New York, NY, June 2002.

[41] A. J. Goldsmith and P. Varaiya, “Capacity, Mutual Information and Coding for Finite-State Markov Channels”, in *IEEE Trans. on Inform. Theory*, vol. 42, pp. 868-886, May 1996.

[42] A. J. Goldsmith and P. Varaiya, “Capacity of Fading Channels with Channel Side Information”, in *IEEE Trans. on Inform. Theory*, vol. 43, pp. 1986-1992, November 1997.

[43] A. Goldsmith, “Capacity and dynamic resource allocation in broadcast fading channels,” 33^{th} *Annu. Allerton Conf. Communication, Control and Computing*, pp. 915-924.

[44] A. J. Goldsmith, “Design and Performance of High-Speed Communication Systems over Time-Varying Radio Channels”, Ph.D. Dissertation, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley, 1994.

[45] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” in *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388-404, March 2000.

[46] S. Grandhi, R. Vijayan, D. J. Goodman and J. Zander, “Centralized Power Control in Cellular Radio Systems”, in *IEEE Trans. on Vehicular Technology*, vol. 42, pp. 466-468, Nov. 1993.

[47] S. Grandhi, J. Zander and R. Yates, “Constrained power control”, *International Journal of Wireless Personal Communications*, vol. 1, no. 4, 1995.

[48] S. Hanly and D. Tse, "Power Control and Capacity of Spread-spectrum Wireless Networks", *Automatica*, vol. 35, no. 12, 1999.

[49] S.V. Hanly, "An algortihm for combined cell-site selection and power control to maximize cellular spread spectrum capacity", in *IEEE JSAC*, vol. 13, pp. 1332-1340, September 1995.

[50] S. Hanly and D. Tse. "Multi-Access Fading Channels: Part II: Delay-Limited Capacities," *IEEE Transactions on Information Theory*, vol. 44, No. 7, November 1998: pp. 2816-2831.

[51] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proceedings of 5th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, August 1999.

[52] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," in *Proceedings of 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, November 2000.

[53] F. P. Kelly, *Reversibility and Stochastic Networks, Wiley Series in Probability and Mathematical Statistics.* New York: Wiley, 1979.

[54] T. E. Klein, *Capacity of Gaussian Noise Channels with Side Information and Feedback*, Ph.D. Thesis, MIT, Cambridge, MA, February 2001.

[55] R. Knopp and P.A. Humblet, "Information capacity and power control in single-cell multiuser communications," *Intl. Conference on Communications*, Seattle, WA, June 1995.

[56] D. Lal, A. Maneshwar, F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian, "Measurement and Characterization of Link Quality in Energy-Constrained Wireless Sensor Networks," submitted to IEEE Globecom 2003.

[57] Q. Li, J. Aslam, and D. Rus, "Heirarchical Power-aware Routing in Sensor Networks," in *Proceedings of the DIMACS Workshop on Pervasive Networking*, May 2001.

[58] D.A. Maltz, J. Broch, D.B. Johnson, "Quantitative lessons from a full-scale multi-hop wireless ad hoc network testbed," in *Proc. IEEE WCNC 2000*, vol. 3, pp. 992 -997, 2000.

[59] M. Medard, S. P. Meyn, J. Huang, and A. J. Goldsmith, "Capacity of Time-Slotted ALOHA packetized Multiple-Access Systems", 2001.

[60] D. Mitra, "An asynchronous distributed algorithm for power control in cellular radio systems", in *Fourth WINLAB Workshop on Third Generation Wireless Information Networks*, 1993.

[61] J. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 1, pp. 1–11.

[62] M. Neely, E. Modiano, and C. Rohrs, "Power and Server Allocation in a Multi-Beam Satellite with Time Varying Channels," *IEEE INFOCOM*, New York, NY, June 2002.

[63] R. W. Nettleton and H. Alawi, "Power Control for a Spread Spectrum Cellular Mobile Radio System", in *Proc. VTC-83, IEEE Vehicular Technology Conf.*, pp. 242-246, 1983.

[64] G.J. Pottie, "Wireless sensor networks", *Information Theory Workshop '98*, Killarney, Ireland. June 1998.

[65] B. Prabhakar, E. Uysal-Biyikoglu and A. El Gamal, "Energy-efficient Transmission over a Wireless Link via Lazy Scheduling", in *Proc. IEEE INFOCOM 2001*, 2001, pp. 386-394.

[66] J.G. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice-Hall, 1994.

[67] T. Simunic, *Energy Efficient System Design and Utilization*, Ph.D. Thesis, Stanford University, 2001.

[68] I. E. Telatar and R. Gallager, "Combining Queuing Theory with Information Theory for Multiaccess", in *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, August 1995, pp. 963-969.

[69] D. Tse and S. Hanly. "Multi-Access Fading Channels: Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities", IEEE Transactions on Information Theory, v. 44, No. 7, November 1998: pp. 2796-2815.

[70] Y-C. Tseng, C-S Hsu, T-Y Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks" in *Proc. IEEE INFOCOM 2002*, vol. 1, pp: 200 -209, 2002.

[71] E. Uysal-Biyikoglu, B. Prabhakar and A. El Gamal, "Energy-efficient Packet Transmission Over a Wireless Link," in *IEEE/ACM Trans. on Networking*, vol. 10, August 2002.

[72] E. Uysal-Biyikoglu and A. El Gamal, "Energy-efficient packet transmission over a multi-access channel," in *Proc. 2002 Intl. Symposium on Information Theory*, 2002, p. 153.

[73] E. Uysal-Biyikoglu, A. El Gamal and B. Prabhakar, "Adaptive Transmission of Variable-Rate Data over a Wireless Channel for Energy-Efficiency", in IEEE GLOBECOM 2002, Taipei, Taiwan, Nov. 2002.

[74] S. Verdu, "On Channel Capacity per Unit Cost", *IEEE Transactions on Information Theory*, vol. 36, no. 5, September 1990.

[75] P. Viswanath, V. Anantharam and D. Tse, "Optimal Sequences, Power Control and Capacity of Synchronous CDMA Systems with Linear MMSE Multiuser Receivers", *IEEE Transactions on Information Theory*, vol. 45(6), Sept., 1999.

[76] J. Walrand, *An Introduction to Queuing Networks*, Prentice Hall, 1988.

[77] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-limited Wireless Networking with Directional Antennas: The Case of Session-Based Multicasting", in *IEEE INFOCOM*, New York, NY, June 2002.

[78] A. Woo and D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," in *Proceedings of 7th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2001)*, July 2001.

[79] M. D. Yarvis, W. S. Conner, L. Krishnamurthy, J. Chhabra, B. Elliot, and A. Mainwaring, "Real-World Experiences with an Interactive Ad Hoc Sensor Network," in *Proceedings of International Workshop on Ad-Hoc Networking (IWAHN 2002)*, August 2002.

[80] R. D. Yates, "A Framework for Uplink Power Control in Cellular Radio Systems", in *IEEE JSAC*, vol. 13, no. 7, pp. 1341-1348, September 1995.

[81] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *IEEE INFOCOM 2002*, June 2002.

[82] E. M. Yeh, *Multiaccess and Fading in Communication Networks*, Ph.D. Thesis, Massachusetts Institute of Technology, September 2001.

[83] W.S. Yoon and T.E. Klein, "Delay-Optimal Power Control for Wireless Data Users with Average Power Constraints," in *Proc. 2002 Int. Symposium on Information Theory*, 2002, p. 53.

[84] E. M. Yeh, "An Inter-Layer View of Multiaccess Communication", in *Proc. 2002 Int. Symposium on Information Theory*, 2002, p. 112.

[85] J. Zander, "Optimum Global Transmitter Power Control in Cellular Radio Systems", in *Proc. IEEE International Symp. on Personal, Indoor and Mobile Radio Communications*, London, Sept. 1991.

[86] J. Zander, "Transmitter power control for co-channel interference management in cellular radio systems", in *Proc. 4^{th} WINLAB Workshop*, New Brunswick, NF, USA, October 1993.