Delay-Bounded Packet Scheduling of Bursty Traffic Over Wireless Channels

Dinesh Rajan, *Member, IEEE*, Ashutosh Sabharwal, *Member, IEEE*, and Behnaam Aazhang, *Fellow, IEEE*

Abstract—In this paper, we study minimal power transmission of bursty sources over wireless channels with constraints on mean queuing delay. The power minimizing schedulers adapt power and rate of transmission based on the queue and channel state. We show that packet scheduling based on queue state can be used to trade queuing delay with transmission power, even on additive white Gaussian noise (AWGN) channels. Our extensive simulations show that small increases in average delay can lead to substantial savings in transmission power, thereby providing another avenue for mobile devices to save on battery power. We propose a low-complexity scheduler that has near-optimal performance. We also construct a variable-rate quadrature amplitude modulation (QAM)-based transmission scheme to show the benefits of the proposed formulation in a practical communication system. Power optimal schedulers with absolute packet delay constraints are also studied and their performance is evaluated via simulations.

Index Terms—Packet scheduling, power control, queuing delay, traffic regulation, wireless channels.

I. INTRODUCTION

THE current and future wireless systems will support a multitude of services with a wide range of delay, rate, and reliability requirements. The task of delivering wide variety of services is complicated by the hostile time-varying nature of wireless channels. The limited battery resources at the mobile devices add yet another dimension in the challenge of reliable content delivery. Thus, it is imperative that methods for wireless transmission should be designed to achieve target delay and throughput with minimal power consumption. Recognizing that transmission power is one of the major battery consumers[1] in a mobile device, in this paper, we study the tradeoff between transmission power and communication delay in wireless channels.

Delaying communication based on channel conditions to save transmission power is commonly used in wireless systems. The transmission scheme which maximizes long-term throughput transmits more power and information in good channel states, and less in poor conditions [1]. Thus, it delays some parts of input traffic to wait for good channel states, buying more utility for the available power resources to achieve the maximal long-term throughput. The information-theoretic concept of power and rate control is reflected in the INFOSTATION [2] based architecture, where mobile nodes transmit data only when they are close to base stations. By waiting for extremely good conditions, more or less guaranteed by the proximity to base stations, mobile devices ensure that transmission power is most efficiently used. But the wait to reach close to base stations (often sparsely placed) can result in large transmission delays.

In this paper, we address the problem of scheduling transmission of packet data over a time-slotted single-user wireless link. We incorporate queuing delay[2] as one of our main design constraint. With a bound on mean queuing delay, our objective is to minimize the average transmission power. The inclusion of explicit delay constraint is motivated by the need to support applications with different delay sensitivities. The solution to the proposed formulation leads to methods which perform power and rate control, even in additive white Gaussian noise (AWGN) channels if the traffic is bursty. Note that, much like [1], rate and power adaptation is done based on the Gaussian channel coding theorem [3]; the relation between instantaneous rate and power is

$$R = \log\left(1 + \frac{P}{\sigma^2}\right) \implies P = \sigma^2\left(e^R - 1\right).$$

The following example clarifies how the convex relation between power and rate leads to reduced power requirements as the delay increases.

Consider the transmission of an *On–Off* source over an AWGN channel with noise variance $\sigma^2 = 1$. The *On–Off* source produces packets at a constant rate r packets per time slot, in the *On* state and no packets in the *Off* state. Let the source arrivals be independent and identically distributed (i.i.d.) across time. First, consider a scheme in which all packets in the system are transmitted as soon as they arrive, the average power required to guarantee error-free reception equals

$$P_0 = \frac{1}{2} \cdot 0 + \frac{1}{2}\left(e^r - 1\right) = \frac{1}{2}\left(e^r - 1\right)$$

[1]In a typical time-division multiple-access (TDMA) phone, approximately 60% of the battery consumption can be attributed to transmission RF amplifier.

[2]There are many components that affect the total transmission delay. In this paper, we only consider queuing delays. The delay due to encoding, propagation, and decoding are assumed to be small and nearly constant, and hence neglected.

(assuming that r packets require transmission at r bits per second per hertz (bits/s/Hz)). Now, consider an alternate transmission scheme which schedules packets as follows. When there are $\frac{r}{2}$ or more packets available, $\frac{r}{2}$ of those packets are transmitted (assuming r is an even integer) and the remaining packets are buffered for future transmission. If the queue has less than $\frac{r}{2}$ packets, all packets are buffered. In the second transmission scheme, the average power required to guarantee error-free reception is $\left(e^{\frac{r}{2}} - 1\right)$, which is lower than P_0 due to convexity of exponential function e^x. The power savings in the second scheme can be attributed to increased packet delay in the system. The above simple example illustrates the average power savings due to power and rate control combined with queuing-based communication delay. In the sequel, we formalize the joint power and rate control schemes, which we generically label as *schedulers*, and derive minimal power methods with delay bounds.

The power gain from additional communication delay was independently recognized in [4], and lazy packet scheduling was proposed to reduce average power consumption. Though related in essence, our formulation differs considerably from the work in [4]. In [4], optimal off-line and near-optimal online schedulers were proposed using both finite and infinite horizon optimization. An indirect bound on packet delay was imposed by requiring the schedulers transmit all packets arriving in time interval $[0, T]$ to depart no later than T. Thus, the scheduler flushes the queue every T seconds. The emphasis in [4] was to demonstrate that additional delay can help save power, hence no specific constraints on target delay were imposed. Schedulers satisfying explicitly imposed packet delay bounds were derived in [5], where the authors found minimal power transmission schemes for a two-state Gilbert–Elliot channels. Though the work in [5] is closest to our formulation, the authors used a more optimistic relation between power and rate—the power was assumed to depend linearly on the rate. As will become evident from the subsequent development in this paper, the gain due to additional delay in [5] is only due to channel time variation, and not the source burstiness. In [6], the authors propose dynamic power control policies in time-varying channels under energy and three different delay constraints; the optimal policies are based on thresholding received signal strength and residual battery energy. Again, the work in [6] notes the impact of increasing delay on energy efficiency but it differs considerably from our formulation of the delay-constrained transmission. Power savings in fading channels with additional delay has also been investigated in [7], [8].

The main contributions in this paper are as follows.

- For AWGN channels, we completely characterize the achievable delay–power region for the set of randomized stationary schedulers which make their transmission decisions based on queue state. A randomized scheduler can take one of several possible actions for a given system state with finite probability, and generalizes deterministic schedulers which take the same action for a given system state. As the first important result, we show that for delay–power characterization, deterministic schedulers form a basis of the set of randomized schedulers.

Throughout this paper, the analysis is performed for finite buffer size systems with no permissible buffer overflow.

As another application of the proposed formulation, we derive optimal delay-bounded schedulers for transmission of constant-rate traffic over finite-state fading channels. Depending on the delay constraint, the scheduling decisions exhibit a mix of time water-filling [1] and outage-minimizing power control [9]. If the delay bound is small, the scheduler transmits packets in poor channel conditions like the outage power control while the scheduler uses only good channel states for large delays.

- The optimal power-minimizing scheduler requires a dynamic programming based optimization, which is computationally cumbersome for large systems (sources with large bursts or systems with large buffer sizes). Based on the empirically observed properties of the optimal schedulers, we propose a single-parameter scheduler, labeled *log-linear* scheduler, with near-optimal performance. The empirically observed near-optimal performance of the log-linear scheduler is used to derive an approximate relation between mean queuing delay and average transmit power. The approximate relation is found to accurately predict the performance of optimal schedulers for moderate to high delays.

- We show that similar results can be obtained if the mean queuing delay bound is replaced by an absolute delay bound such that no packet suffers more than a prespecified maximum delay.

- To demonstrate the practical application of the proposed scheduling, we construct a convolutionally coded variable-rate quadrature amplitude modulation (QAM) system, and show that small additional delay can lead to substantial power savings. In variable-rate QAM systems, where the transmission rate and power is controlled by the transmitter, the receiver is unaware of the transmitted rate. We show that the transmission rate information (required for appropriate decoding) can be derived *without* requiring any protocol information.

All results in this paper assume i.i.d. packet arrivals in each slot for simplicity. The analysis can be extended to Markov arrival processes which are more appropriate models for multimedia applications such as video transmission.

The main results in the paper can be attributed to inclusion of finite delay constraints along with a bursty traffic model. If delay is not a consideration, adequate queuing delay can completely remove all burstiness and standard information-theoretic analysis applies [10]. The importance of incorporating the traffic model in the design of wireless communication systems has been well recognized; for insightful reviews, see [11], [12]. The effect of queuing delay on probability of error is studied in [13] using a simplified Gaussian multiple-access channel model.

In this paper, we study only single-user systems and schedule packets over time. Scheduling is more commonly referred to selecting packets when there are multiple input flows [14]; e.g., first come first serve (FCFS), last come first serve (LCFS), earliest deadline first (EDF), and weighted fair queuing (WFQ).

There is extensive work on scheduling over wireless channels [15]–[19]. Again, the main idea is to efficiently use system resources, often with an aim of fair division of resources. One of the interesting results in this paper is that efficient scheduling of packets helps even in a single-user single flow system. In an actual wireless network, the proposed packet scheduler can be combined with a multiflow service discipline, like WFQ, to achieve fairness with efficient power usage. The single-user problem provides methods for traffic regulation in circuit-switched TDMA systems, and the multiuser problem forms the basis for scheduling in code-division multiple-access (CDMA) systems. In our related work [20], we focus on the multiuser problem.

In this paper, we focus on schedulers which do not drop or loose packets, and seek to minimize their average power utilization. If the minimal power scheduler requires more power than available, then the transmitter is forced to drop packets. A power-limited outage based formulation is studied in [21]–[23], where we show that additional delay helps reduce probability of outage.

The rest of the paper is organized as follows. In Section II, we introduce the system model and set up the scheduling problem of interest. We provide a technique to find power-efficient schedulers under average delay constraints and discuss some of their properties in Section III. Scheduling under absolute delay constraints is addressed in Section IV. We demonstrate scheduling gains in a simple practical system in Section V. We conclude in Section VII.

II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we introduce the time-slotted system model used in the paper. We define the queue, source, and channel model in Section II-A. Our emphasis in this paper is on design of schedulers which guarantee information-theoretic reliability, discussed in Section II-B. Finally, a class of reliable schedulers, labeled zero-outage schedulers, is presented and characterized in Section II-C.

A. System Model

We consider a single-user time-slotted system with an input buffer, shown in Fig. 1; the period of each time slot is assumed to be T_s seconds. The source produces packets at an average rate of λ packets per time slot. In time slot n, the source produces a_n packets, each of size S bits, where a_n has a distribution $p(a_n)$ with a finite support $[0, \ldots, M]$, i.e., the largest number of packets that arrive in a time slot is M. The arrival process $\{a_n\}$ is assumed to be i.i.d. from one time slot to another. The arriving packets are queued in the input buffer, which can store a maximum of L packets.

The number of queued packets in the buffer at the beginning of the nth time slot are denoted by x_n. The scheduler chooses u_n packets for transmission at the beginning of the time slot, and uses power P_n for transmission. Since the length of the time slot is fixed, the rate of transmission is varied based on the selected number of packets u_n. Denoting the transmitted signal by X_n, the received signal is given by
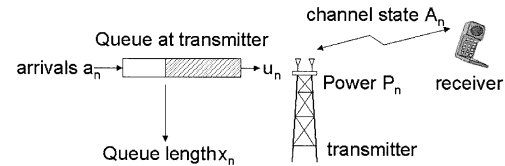
$$Y_n = A_n X_n + \varepsilon_n \tag{1}$$

Fig. 1. Schematic of system model.

where ε_n is the complex circularly symmetric AWGN with zero mean and variance σ^2. The channel gain A_n is assumed to be constant over the period of a time slot, i.e., the coherence interval of the channel is the same as the length of the time slot. Two channel models are considered: In the first model, $A_n \equiv 1$, thus, the channel model (1) reduces to a Gaussian channel. In the second model, A_n is assumed to be a finite-state Markov channel, i.e., A_n forms a Markov chain. Both the transmitter and receiver are assumed to have perfect knowledge of the channel gain A_n.

The transmitted signal X_n is a function of the number of packets transmitted u_n, the channel coding, modulation, and waveform used. We use information theoretically optimal coding and modulation to obtain X_n, which serves two purposes. First, we can get closed-form relationship between u_n and power P_n needed for reliable transmission. Second, the power we obtain serves as a universal lower bound on power required by any scheduler designed for a specific coding and modulation scheme.

At the transmitter, the buffer update is given by

$$x_{n+1} = \min\left(x_n + a_n - u_n, L\right). \tag{2}$$

The average packet delay is related to the average buffer length via Little's theorem [24] as follows:

$$D_{\text{avg}} = \frac{1}{\lambda} \mathbb{E}\left\{x_n\right\} \tag{3}$$

where $\lambda = \mathbb{E}\left\{a_n\right\}$ is the average packet arrival rate. We assume that all packets that arrive in time slot n can be transmitted only in time slot $(n+1)$ or later. A natural constraint on u_n is that $0 \leq u_n \leq x_n$, i.e., we cannot transmit more packets than available in the queue. The smallest average delay in the system is achieved when all buffered packets are transmitted in each time lot, i.e., $u_n = x_n$, which implies that $D_{\text{avg}} = \frac{1}{\lambda} \mathbb{E}\left\{x_n\right\} = 1$, assuming that $L \leq M$. In other words, the buffer is flushed at all time instants, and all packets are transmitted in the slot immediately after which they arrive.

A scheduler is a mapping from the current buffer state x_n and channel state A_n to the number of packets transmitted. u_n and transmit power P_n. (See Section II-C for more details.) We will restrict our attention to a class of schedulers which do not drop any packet and achieve a predetermined minimum reliability for every transmitted packet; a scheduler which does not lose any packets is labeled as a zero-outage scheduler. The objective is to characterize the achievable delay–power region which is defined as follows.

Definition 1 (Delay–Power Region): A point (D, P) on the average delay versus average transmit power plane is achievable if there exists a zero-outage scheduler which achieves an average delay no greater than D and average power no greater than P.

The delay–power region can be characterized by finding the minimal average transmit power required for any delay over the class of all randomized zero outage schedulers Θ (formally defined in Section II-C). Formally

$$P^*(D_o) = \min_{\substack{\Theta \\ D_{\mathrm{avg}} \leq D_o}} \lim_{n \to \infty} \mathbb{E}\{P_n\} \qquad (4)$$

where D_o is the average packet delay bound and $P^*(D_o)$ is the minimum average power required to achieve average delays no greater than D_o. Given the solution $P^*(D_o)$, all points (D_o, P) such that $P \geq P^*(D_o)$ belong to the delay–power region.

B. Power Control

In this paper, we restrict our attention to schedulers which guarantee a certain level of reception reliability. Our notion of reliability is motivated by the concept of outage proposed in [25], where a packet is reliably received if the instantaneous mutual information is higher than the required rate. To invoke mutual information based reliability, we require that T_s be sufficiently long to transmit M packets close to channel capacity.

If the scheduler chooses to transmit u_n packets of length S bits in a time slot of length T_s seconds, then the required transmission rate is given by $\frac{u_n S}{T_s \log_2(e)}$ nats per second. Using the results in [3], we define the capacity function as

$$\rho(P_n, A_n) = \frac{T_s}{S} \log\left(1 + \frac{|A_n|^2 P_n}{\sigma^2}\right) \text{ packets per time slot} \qquad (5)$$

where, for simplicity, we have assumed that the system bandwidth is 1 Hz. For simplicity, we also assume $T_s = S \log_2(e)$. The quantity $\rho(\cdot)$ is the Shannon capacity of a Gaussian channel with noise variance $\frac{\sigma^2}{|A_n|^2}$ and an average power of P_n, and represents the maximum reliably achievable rate. Using (5), the appropriate power control to ensure reliable reception of packets follows immediately. To transmit u_n packets, the scheduler selects the power P_n such that $u_n = \rho(P_n, A_n)$, leading to the following power control:

$$P_n(u_n, A_n) = \frac{\sigma^2}{|A_n|^2}\left(e^{u_n} - 1\right). \qquad (6)$$

Note that P_n is a strictly convex function of u_n, a linear increase in the number of packets requires an exponential increase in required transmit power.

The convex relation between required transmit power and rate is also evident in the nonasymptotic regime, either by considering error exponents [26] or finite-length codes. The error exponent for Gaussian channels varies logarithmically in power and linearly in rate. Thus, to guarantee the same error performance for all scheduler decisions u_n, the required power control will be similar to (6).

As an example of practical methods, consider a simple uncoded system using Q-ary QAM modulation. For large $(Q \geq 16)$ rectangular QAM constellations, the average power required to achieve a certain minimum distance between constellation points is only marginally greater than the average power required to achieve the same minimum distance using

the best known QAM [27]. The constellation points are given by two-tuples

$$\left((2q_1 - 1 - Q)d, (2q_2 - 1 - Q)d\right), \text{ for } q_1, q_2 = 1, 2, \ldots, Q.$$

The transmission rate is $2\log_2 Q$ bits per symbol and the average transmit power P is $\frac{2d^2(Q+1)(Q-1)}{3Q}$. The minimum distance between signal points is given by $2d$ for all Q, hence, these constellations achieve approximately the same bit-error rate. Thus, increasing the total information rate by a factor of ζ will require an average transmit power of $PQ^{\zeta-1}$. Equivalently, increasing the average power by a factor of ζ gains us an additional bit rate of $2\log_2\zeta$ bits per symbol. Thus, the uncoded systems also follow the exponential relationship between transmission rate and transmit power.

C. Zero-Outage Schedulers

In its general form, we define a *randomized* scheduler α as a memoryless probabilistic mapping from the buffer state x_n and channel fading state A_n to the number of packets u_n and transmission power P_n, i.e., $\alpha : (x_n, A_n) \mapsto (u_n, P_n)$. Thus, a scheduler is characterized by probabilities, $\mathrm{Prob}((u_n, P_n)|(x_n, A_n))$. Based on the discussion in Section II-B, the power P_n is a function of u_n which reduces the mapping α to a single-dimensional variable $\alpha : (x_n, A_n) \mapsto u_n$. For the special case of AWGN channel $(A_n \equiv 1)$, α is further reduced to $\alpha : x_n \mapsto u_n$. For the sake of clarity, all results in this section are derived for the case of AWGN channels and can be easily extended to finite-state fading channels.

With some abuse of notation, we use $\alpha_{j,i}$ to denote the probability of transmitting j packets given that the queue has i packets, i.e.,

$$\alpha_{j,i} = \mathrm{Prob}(u_n = j | x_n = i).$$

Thus, every scheduler α can be represented by an $(L+1) \times (L+1)$ upper triangular matrix (since any scheduler cannot transmit more packets than available in the buffer, thus, $\alpha_{j,i} = 0$ for $j > i$), with entries $\alpha_{j,i}$. Since $\alpha_{j,i}$ are probabilities

$$\sum_{j=0}^{i} \alpha_{j,i} = 1. \qquad (7)$$

The set of valid schedulers will be denoted by $\mathcal{S}$, where each scheduler $\alpha \in \mathcal{S}$ satisfies the above properties.

A scheduler is labeled *deterministic* if $\alpha_{j,i} \in \{0, 1\}$, i.e., the scheduler takes the same action for each queue state at all times. It follows immediately from the definition that the deterministic schedulers form a subset of randomized schedulers. Later in this section, we will show that the deterministic schedulers are a basis of the set of randomized schedulers.

Since the source is assumed to be i.i.d. and the schedulers are assumed to be memoryless, all schedulers lead to the stationarity of x_n. Due to the memoryless nature of the schedulers, the queue state forms a first-order Markov chain, which we assume to be aperiodic. Thus, we have a concise representation of the queue state process as follows. Denote by $\boldsymbol{C} = [C_{j,i}]$ the matrix of transition probabilities, where $C_{j,i}$ is the probability of transition from buffer state $x_n = i$ to buffer state $x_{n+1} = j$. Note that $\boldsymbol{C}$ is of size $(L+1) \times (L+1)$ and is not a function of time index

n due to stationarity of schedulers. The stationary probability of buffer state i is denoted by s_i where, $s_i = \mathrm{Prob}\,[x_n = i]$. Let $\boldsymbol{s} = [\,s_0\ s_1\ s_2\ \dots\ s_L\,]^T$ denote the vector of stationary probabilities. From the definition of $\boldsymbol{C}$ and $\boldsymbol{s}$, it follows that

$$\boldsymbol{C}\,\boldsymbol{s} = \boldsymbol{s}. \tag{8}$$

Note that $\boldsymbol{C}$ depends on the choice of α, which implies $\boldsymbol{s}$ is a function of α. The average packet delay of any scheduler can be expressed as

$$\begin{aligned} D_{\mathrm{avg}}(\alpha) &= \frac{1}{\lambda}\mathbb{E}[x_n] \\ &= \frac{1}{\lambda}\left(\sum_{i=0}^{L} i s_i\right). \end{aligned} \tag{9}$$

Similarly, the average transmission power P_{avg} is given by

$$\begin{aligned} P_{\mathrm{avg}}(\alpha) &= \mathbb{E}[P_n] \\ &= \sum_{i=0}^{L}\sum_{j=0}^{i} s_i \alpha_{j,i}\, P_n(j,1) \end{aligned} \tag{10}$$

where P_n is the power control defined in (6). For a given arrival process, more than one scheduler can have the same average delay and power; the following definition of equivalence class formalizes this.

Definition 2 (Delay–Power Equivalence Class): Two schedulers are said to belong to the equivalence class $\boldsymbol{Q}_{D_0,P_0}$ if they have the same average delay and average power. Formally

$$\boldsymbol{Q}_{D_0,P_0} = \{\alpha : \alpha \in \mathcal{S},\ D_{\mathrm{avg}}(\alpha) = D_0,\ P_{\mathrm{avg}}(\alpha) = P_0\}.$$

Two schedulers, $\alpha_1, \alpha_2 \in \boldsymbol{Q}_{D_0,P_0}$ are then said to be equivalent, and represented by $\alpha_1 \equiv \alpha_2$.

The importance of the preceding equivalence class definition will be become apparent in the next section. The following example illustrates that the equivalence classes of the set of all schedulers can have more than one element.

Example 1: The following two schedulers are equivalent for $M = 2$:

$$\alpha_a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \alpha_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In this case, the two schedulers only differ in buffer state 3 that occurs with zero probability.

In the rest of the paper, we will restrict our attention to a special class of schedulers which ensure a preset level of reliability, formally defined as follows.

Definition 3 (Zero-Outage Scheduler): A scheduler is said to be zero outage if packets are not dropped at the transmitter, there are no buffer overflows and power P_n is chosen to ensure reliable reception of the packets. The set of all zero-outage schedulers is denoted by Θ, i.e.,

$$\Theta = \{\alpha : \alpha \in \mathcal{S},\ \alpha \text{ is zero outage}\}. \tag{11}$$

The following proposition completely characterizes the set of zero-outage schedulers Θ.

Proposition 4: Consider a queue with finite buffer size L and an input process with no more than M packet arrivals in one slot. Assume that the buffer is initially empty. A stationary stochastic scheduler is zero outage if and only if for each state $x_n = i$, $i \in L - M, \dots, L$ one of the following two conditions are satisfied.

1) The stationary probability is zero for state i, i.e., $s_i = 0$.
2) If $z = \min\{j : \alpha_{j,i} \neq 0\}$ denotes the minimum number of packets transmitted in state i under the given scheduler, then $(i - z) \leq (L - M)$.

Proof: See Appendix A $\qquad\square$

Now, we note that the set of deterministic schedulers spans the set of all randomized schedulers, $\mathcal{S}$ and hence a basis for $\mathcal{S}$. The basis elements can be constructed column by column as follows. Consider the ith column of α. The $(i+1)$th element to $(L+1)$th element of the ith column are zero. Hence, i vectors of dimension $(L+1)$ form the basis for the ith column. The basis vectors are

$$[1\ 0\ \cdots\ 0],\ [0\ 1\ 0\ \cdots\ 0],\dots,[0\ \cdots\ 0\ 1\ 0\ \cdots\ 0]$$

where in the last basis, the 1 is in the ith position. Taking all possible combinations of the basis of all $(L+1)$ columns, we get a basis for the set of all schedulers. An important property of all schedulers in $\mathcal{S}$ which allows us to do this basis expansion is (7). Finally, note that the total number of deterministic policies is $(L+1)!$ and all of them are not zero outage.

The set of deterministic schedulers form a redundant basis for the set of all randomized schedulers, $\mathcal{S}$, and hence any scheduler can possibly have multiple basis expansions. Below, we give a constructive method for finding the coefficients of the basis expansion, in terms of deterministic schedulers. This constructive method is used in the proof of Theorem 8. Let $\alpha \in \mathcal{S}$ be a given randomized scheduler of size $(L+1) \times (L+1)$, and the following steps recursively generate the basis expansion of α.

1) Set $i = 1$ and $\Xi_i = \{(1, \alpha)\}$, where the jth member of of Ξ_i is a two-tuple (η_{ij}, ξ_{ij}) such that the first element of the tuple is the coefficient of the expansion and the second element of the tuple is the scheduler in the expansion. Thus, the expansion in the first step is $\alpha = \eta_{11}\xi_{11} = 1 \cdot \alpha$. The following construction updates the number and the elements in the set Ξ_i in each step, till all schedulers in the expansion are deterministic.

2) In step i, we consider each element of Ξ_i separately and generate $(i+1)$ elements for Ξ_{i+1}. Thus,

$$|\Xi_{i+1}| = (i+1)|\Xi_i|.$$

3) For each ξ_{ij}, write it as the convex combination of $(i+1)$ matrices that differ only in the $(i+1)$th column. The $(i+1)$th column of scheduler $\xi_{(i+1)j'}$ where

$$j' = (j-1)(i+1) + 1, \dots, j(i+1)$$

has value 1 in row $(j' - (j-1)(i+1))$ and zero in all other rows, i.e., the $i+1$ different columns form a basis for R^{i+1}. The schedulers $\xi_{(i+1)j'}$ are identical to ξ_{ij} for all columns other than column $(i+1)$.

The coefficients are given by

$$\eta_{(i+1)j'} = \eta_{ij}\xi_{ij,(j'-(j-1)(i+1))(i+1)}$$

where $\xi_{ij,kl}$ is the element in the kth row and lth column in ξ_{ij}. Since scheduler α satisfies (7), the weighted sum of the schedulers $\xi_{(i+1)j'}$ give the scheduler ξ_{ij} and all schedulers $\xi_{(i+1)j'}$ satisfies (7).

4) Set $i \leftarrow i + 1$ and repeat steps 2)–4) if $i < L + 1$.

5) The elements in Ξ_{L+1} give the basis expansion of

$$\alpha = \sum_{j=1}^{(L+1)!} \eta_{Lj} \xi_{Lj}.$$

The basis expansion technique is best illustrated in the following simple example.

Example 2:

$$\alpha = \begin{bmatrix} 1 & 0.2 & 0.1 \\ 0 & 0.8 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix}$$

$$= 0.2 \begin{bmatrix} 1 & 1 & 0.1 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix} + 0.8 \begin{bmatrix} 1 & 0 & 0.1 \\ 0 & 1 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix}$$

$$= 0.2 \left\{ 0.1 \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 0.3 \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right.$$

$$\left. + 0.6 \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\} + 0.8 \left\{ 0.1 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right.$$

$$\left. + 0.3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} + 0.6 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\}.$$

We end this section with two propositions, where we consider a queue with finite buffer size L and an input process with no more than M packet arrivals in one slot. Further define the set of deterministic schedulers

$$\mathbb{W} = \{\alpha : \alpha \in \mathcal{S}, \ \alpha_{j,i} \in \{0,1\} \ \forall i,j = 0,1,\ldots,L\}$$

and recall that the set of zero-outage schedulers is denoted by Θ. The following lemma shows that for every zero outage deterministic scheduler $\alpha \in \boldsymbol{Q}_{D_0,P_0}$ which satisfies Condition 1 of Proposition 4, there exists another deterministic scheduler $\hat{\alpha} \in \boldsymbol{Q}_{D_0,P_0}$ which satisfies Condition 2 of Proposition 4.

Proposition 5 (Equivalent Elements of $\boldsymbol{Q}_{D_0,P_0}$): Let $\alpha \in \boldsymbol{Q}_{D_0,P_0} \cap \mathbb{W}$ be a deterministic scheduler such that there exists a buffer state i such that $s_i = 0$. Then there exists a deterministic scheduler $\hat{\alpha} \in \boldsymbol{Q}_{D_0,P_0} \cap \mathbb{W}$ such that $(i-z) \le (L-M)$ where $z = \min\{j : \hat{\alpha}_{j,i} \ne 0\}$.

   Proof: See Appendix B. $\square$

In the next proposition, we show that in the basis expansion of zero-outage schedulers, only zero-outage deterministic scheduler are needed and the coefficients corresponding to the nonzero-outage schedulers are zero.

Proposition 6 (Basis Expansion of Elements in Θ): Let $\{\gamma_i\}_{i=1}^F$ represent the basis set of $\mathcal{S}$, where $\gamma_i \in \mathbb{W}$ for all i and $F = |\mathbb{W}|$ is the cardinality of the set $\mathbb{W}$. For any stationary

randomized scheduler $\phi \in \Theta$, let its expansion in terms of basis vectors be

$$\phi = \sum_{i=1}^F \eta_i \gamma_i, \qquad \text{where } 0 \le \eta_i \le 1 \ \text{ and } \sum_{i=1}^F \eta_i = 1.$$

Then, $\eta_i = 0$ for all $\gamma_i \in \mathbb{W} - \Theta$, where $\mathbb{W} - \Theta$ is the set difference.

   Proof: See Appendix C. $\square$

The following example illustrates the results of Propositions 5 and 6.

Example 3: Consider the following four schedulers (scheduler α_a and α_b are same as in Example 1, and reproduced here for the convenience of the reader)

$$\alpha_a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \alpha_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\alpha_c = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \alpha_d = \begin{bmatrix} 1 & 0.4 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is easy to see that $\alpha_d = 0.6\alpha_a + 0.4\alpha_c$. Now, if $M = 2$, then α_d is zero outage because it satisfies Condition 2 of Proposition 4. It is also easy to verify that α_a and α_c are zero outage schedulers. Again, when $M = 2$, state $i = 3$ has zero probability of occurance under scheduler α_b. However, scheduler α_a satisfies Condition 2 of Proposition 4 and has the same delay and power as scheduler α_b.

The converse to Proposition 6 is however not true, i.e., a convex combination of two zero-outage deterministic schedulers is not always a zero-outage randomized scheduler, as illustrated in Example 4.

Example 4: Let α_b and α_c be as in Example 3. Let $M = 2$ and

$$\alpha_e = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $s_i = 0$ for buffer state 3 under scheduler α_e, it is zero outage. A convex combination of α_e and α_c does not guarantee a scheduler with zero outage. The probability of being in buffer state 3 is nonzero under scheduler α_c, and if we then switch to scheduler α_e, we will remain in buffer state $x_n = 3$ indefinitely and continue to loose packets due to buffer overflow. Instead, a convex combination of schedulers α_b and α_c will result in schedulers that have no buffer overflows. Note again that when $M = 2$, α_e and α_b have the same average delay and power.

III. POWER-EFFICIENT SCHEDULERS

In this section, we provide a method to compute the minimal power scheduler in (4). Driven by the high computational complexity of solving for the optimal scheduler (using a dynamic program), we propose two suboptimal schedulers in Section III-B. In Section III-C, we derive a closed-form approximation for the average power as a function of delay.

Extension of the optimal schedulers to fading channel is briefly discussed in Section III-D. Finally, we provide numerical results to illustrate the savings in power with increasing delay, and state some empirically observed properties of optimal schedulers in AWGN and fading channels.

A. Optimal Packet Schedulers

We prove a lemma which is used in the proof of Theorem 8 on convexity of set of randomized schedulers.

Lemma 7 (Delay–Power of Two "Almost" Identical Schedulers): Let α_A and α_B be two zero-outage schedulers which are identical in all columns except the ith column. The delay and power of scheduler α which is given by a convex combination of α_A and α_B, equals a convex combination of the delays and powers of α_A and α_B, respectively. In other words, if $\alpha = \eta\alpha_A + (1-\eta)\alpha_B, \eta \in [0,1]$ then

$$D_{\text{avg}}(\alpha) = \eta' D_{\text{avg}}(\alpha_A) + (1-\eta')D_{\text{avg}}(\alpha_B)$$

and

$$P_{\text{avg}}(\alpha) = \eta' P_{\text{avg}}(\alpha_A) + (1-\eta')P_{\text{avg}}(\alpha_B)$$

where $0 \le \eta' \le 1$. Further

$$\eta' = \frac{\eta s_{\alpha_B,i}}{\eta s_{\alpha_B,i} + (1-\eta)s_{\alpha_A,i}}$$

where $s_{\alpha_A,i}$ and $s_{\alpha_B,i}$ are the stationary probabilities of being in state i under schedulers α_A and α_B, respectively.

Proof: See Appendix D. $\square$

An application of Lemma 7 is given in the following example.

Example 5: Let schedulers α_a, α_c, and α_d be as in Example 3. For $M = 2$ and a uniform arrival distribution, the delay and power under scheduler α_a equals $D_{\text{avg}}(\alpha_a) = 1$ time slot and $P_{\text{avg}}(\alpha_a) = 2.7$. Similarly, $D_{\text{avg}}(\alpha_c) = 1.33$ time slots and $P_{\text{avg}}(\alpha_c) = 4.25$. The stationary probabilities for the different states under schedulers α_a and α_c are

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \frac{2}{9} & \frac{1}{3} & \frac{1}{3} & \frac{1}{9} \end{bmatrix}.$$

Note again that $\alpha_d = 0.4\alpha_c + 0.6\alpha_a$. We find that

$$\begin{aligned} D_{\text{avg}}(\alpha_d) &= \eta' D_{\text{avg}}(\alpha_a) + (1-\eta')D_{\text{avg}}(\alpha_c) \\ &= 1.13 \text{ time slots} \end{aligned}$$

and

$$\begin{aligned} P_{\text{avg}}(\alpha_d) &= \eta' P_{\text{avg}}(\alpha_a) + (1-\eta')P_{\text{avg}}(\alpha_c) \\ &= 3.32 \end{aligned}$$

where $\eta' = \frac{0.6 \times \frac{1}{3}}{0.6 \times \frac{1}{3} + 0.4 \times \frac{1}{3}} = 0.6$.

By the repeated application of Lemma 7, it is easy to see that if we have more than two schedulers which differ in the same column, then the delay and power of the convex combination of such schedulers is given by the convex combination of the delays and powers of the individual schedulers.

The following theorem proves that the average power and delay for an arbitrary zero-outage randomized scheduler can be obtained as the convex combination of delays and power of zero-outage deterministic schedulers.

Theorem 8 (Characterization of Delay and Power): Consider a queue with finite buffer size L and an input process with

no more than M packet arrivals in one slot. For any randomized scheduler $\phi \in \Theta$, there exists $\eta'_i \in [0,1], i = 1, 2, \dots, F' = |\mathbb{W} \cap \Theta|$, with

$$\sum_{i=1}^{F'} \eta'_i = 1$$

such that

$$D_{\text{avg}}(\phi) = \sum_{i=1}^{F'} \eta'_i D_{\text{avg}}(\gamma_i) \tag{12}$$

$$P_{\text{avg}}(\phi) = \sum_{i=1}^{F'} \eta'_i P_{\text{avg}}(\gamma_i) \tag{13}$$

where γ_i is a scheduler and $\gamma_i \in \mathbb{W} \cap \Theta$ for all i.

Proof: See Appendix E. $\square$

Thus, the average packet delay achieved by any zero-outage randomized scheduler is given by a convex combination of the average packet delays achieved by all possible zero-outage deterministic schedulers. In addition, the same convex combination of the average powers of the zero-outage deterministic schedulers gives the average power of zero-outage randomized policy.

In Proposition 6, it was shown that every zero-outage scheduler in Θ admitted an expansion using only deterministic zero-outage schedulers. In the proof of Theorem 8, it is shown that among the several possible basis expansion of $\phi \in \Theta$ there is at least one which insures delay and power of ϕ is a convex combination of delay and power of some zero-outage deterministic schedulers. The following lemma shows a converse to this result, i.e., there exists schedulers which achieve delays and powers which are the convex combination of the delays and powers of any two arbitrary zero-outage deterministic schedulers.

Lemma 9 (Converse to Theorem 8): Let α_A and α_B be two zero-outage deterministic schedulers with average delays $D_{\text{avg}}(\alpha_A), D_{\text{avg}}(\alpha_B)$ and average powers $P_{\text{avg}}(\alpha_A)$, $P_{\text{avg}}(\alpha_B)$, respectively. Then, there exists a zero-outage scheduler which achieves average delay

$$\eta D_{\text{avg}}(\alpha_A) + (1-\eta)D_{\text{avg}}(\alpha_B)$$

and average power

$$\eta P_{\text{avg}}(\alpha_A) + (1-\eta)P_{\text{avg}}(\alpha_B)$$

for all $0 \le \eta \le 1$.

Proof: See Appendix F. $\square$

It should be emphasized that the convex span of $\mathbb{W} \cap \Theta$ is larger than Θ (some schedulers in convex span can be nonzero outage). Given that the randomized scheduler performance can be obtained by appropriate linear combination of deterministic schedulers, the delay–power region can be characterized as follows.

Corollary 10 (Characterization of Delay–Power Region): The boundary of the achievable region in the delay–power plane is piecewise linear with the vertices achieved by deterministic schedulers.
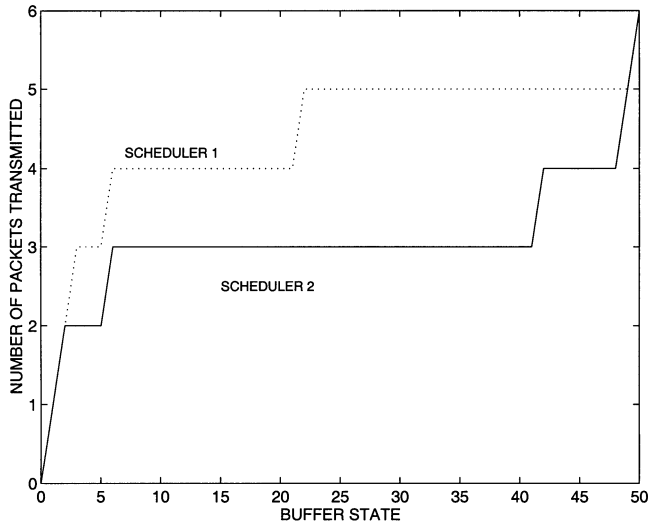
Fig. 2.   Schedulers for achieving different average delay bounds with minimal average power. (Buffer size $L = 50$, maximum arrival $M = 6$, uniform arrival distribution.)

Proof: The total number of deterministic zero-outage schedulers is finite, and the delay–power performance of all randomized zero-outage schedulers is given by a convex combination of delay and power of deterministic zero-outage schedulers. Hence, the corollary follows.   □

To find the achievable delay–power region, we use a dynamic programming technique commonly known as value iteration algorithm (VIA) [28]. The details of the VIA are given in Appendix G.

The delay bound D_o is reduced monotonically and the minimal power *deterministic* policy is found among $\mathbb{W} \cap \Theta$ for each D_o. Thus, instead of solving (4), we are required to solve the following optimization problem:

$$P^{\dagger}(D_o) = \min_{\substack{\Theta \cap \mathbb{W} \\ \mathbb{E}[x_n] \leq \lambda D_o}} \lim_{n \to \infty} \mathbb{E}\{P_n\}. \tag{14}$$

The convex hull of the resulting $(D_o, P^{\dagger}(D_o))$ pairs gives the delay–power region, using Theorem 8 and Corollary 10.

Two examples of the transmission policy obtained using VIA are shown in Fig. 2, which achieve two different points on the boundary of the achievable delay–power region. The figure illustrates the basic nature of the scheduler actions (u_n) as a function of queue state (x_n) as packet delay increases. Recall that the minimum possible average delay in our system is one, which is achieved by a scheduler which transmits all packets it receives as soon as possible. Thus, if schedulers start from zero buffer state, then schedulers achieving average delay one are such that $u_n = x_n$, at least for $x_n \leq M$. As the delay increases beyond one time slot, the scheduler chooses to transmit fewer than x_n packets for some states, and delays some of the packets; this can be seen in the behavior of Schedulers 1 and 2 in Fig. 2, when $M = 6$. Scheduler 1 achieves a smaller delay (≈ 1.8 time slots) compared to Scheduler 2 (≈ 7.5 time slots) with higher power

consumption (6.82 versus 5.08 units of power). Note that neither of the schedulers in Fig. 2 transmit x_n packets for $x_n \geq 3$, i.e., they do not flush the buffer like a unit delay scheduler. Furthermore, as the delay increases, the schedulers tend to transmit close to average arrival rate $\lambda = 3$ packets/time slot (notice the behavior for buffer states $6 \leq x_n \leq 40$ for Scheduler 2). For each of the schedulers, buffer overflows are avoided by transmitting close to M packets when x_n is close to L (see $x_n \geq 42$ for Scheduler 2).

The reduction in power requirements due to additional delay can be attributed to convexity of the relation between power and rate of transmission (6). Transmitting a large number of packets requires an exponentially increasing amount of power, thereby resulting in a large average transmission power for small delays. As the delay deadlines become less stringent, packets can be delayed so that the transmitter can minimize the transmission of a large number of packets, to reduce large-power spikes and hence reduce average power consumption. As the average delay increases unboundedly (assuming that the buffer size also increases to infinity), the scheduler delays every packet such that (approximately) only λ packets are always transmitted in every time slot. In Section III-E, the scheduler behavior is discussed further using spectral analysis.

Alternately, the power reduction with delay can be understood using the following source coding interpretation. Note that if we consider the *information* in the packet arrival times, in addition to the contents of the packets, then our scheduler action is analogous to that of a lossy source coder. The higher the delay allowed, the more is the compression of information. For example, if delay equals one then the arrival times of the packets are exactly known at the receiver; equivalent to lossless encoding of timing information. As the average packet delay increases, the scheduler delays certain packets more than others and the receiver will only have a *noisy* version of the packet arrival times; this is similar to lossy coding. Asymptotically, as the delay goes to infinity, the receiver will have no information about the actual arrival times of the packets; this corresponds to encoding timing at zero rate. The loss of information with increasing delays can be attributed to the fact that the scheduler mapping $\alpha : x \mapsto u$ is not invertible for $x_n \leq M$ except when delay equals 1 (assuming that buffer is initially empty).

Finally, we characterize extremal source statistics, resulting in the lowest power for any delay constraint D_{avg} and highest power source for $D_{\text{avg}} = 1$.

Proposition 11 (Extremal Power Source):   Let the arrival distribution be denoted by

$$\boldsymbol{m} = [m_0 \quad m_1 \quad m_2 \quad \cdots \quad m_M]^T$$

where m_i is the probability that i packets arrive in any time slot. Then the input distribution that results in the least transmit power in an AWGN channel, for any average delay is given by

$$m_i = \begin{cases} 1 - \lambda + \lfloor \lambda \rfloor, & i = \lfloor \lambda \rfloor \\ \lambda - \lfloor \lambda \rfloor, & i = \lfloor \lambda \rfloor + 1 \\ 0, & \text{else} \end{cases}$$

where $\lfloor \lambda \rfloor$ represents the greatest integer no larger than λ.

Also, the input distribution that results in the highest transmit power for a delay bound[3] of $D_o = 1$ is given by

$$m_i = \begin{cases} 1 - \frac{\lambda}{M}, & i = 0 \\ \frac{\lambda}{M}, & i = M \\ 0, & i = 1, 2, \ldots, M - 1. \end{cases}$$

Proof: See Appendix H. □

B. Suboptimal Schedulers

For large buffer sizes L, the number of possible states in the VIA increase exponentially and hence computing the optimal scheduler is computationally intensive. For the cases where the arrival distribution is measured in real time and the optimal scheduler is adapted over time, the implementation of VIA can lead to an intractable design. Motivated by the high complexity of computing the optimal scheduler, we present two suboptimal schedulers in this section. These suboptimal schedulers are deterministic. To obtain randomized suboptimal schedulers, we can again use convexity properties of randomized schedulers (Theorem 8 and Lemma 9).

A simple method for complexity reduction is to reduce the number of states in the VIA (see Appendix G). The following state reduction is most useful for moderate to large delay scenarios. As delay increases beyond 1, the scheduler tends to take the same action in several consecutive queue states. Thus, to reduce the state diagram size, multiple states can be morphed into one state. For instance, for all states $x_n = j, \ldots, j + K$, the scheduler can be constrained to take the same action u_n, thereby reducing the state space. It is clear that the constrained system will be suboptimal, but the loss due to additional constraints can be minimized by appropriate choice of the number and lengths of constraint intervals; some guidelines for the choice of the intervals can be derived from the empirical observations about the optimal schedulers made in Section III-E.

The second suboptimal scheduler is labeled the *log-linear* scheduler, described as follows. For a queuing delay of one time slot, the optimal scheduler[4] flushes the buffer at all time slots, i.e., $u_n = x_n$ and the corresponding power required in each time slot is proportional to $e^{u_n(x_n)}$. As the delay increases, we observe that the optimal scheduler tends to choose $u_n(x_n)$ so that the power in each time slot is linearly proportional to x_n, thereby "equalizing" the power penalty in large buffer states x_n. For equalizing the power, the scheduler picks packets $u_n(x_n) \approx \log(x_n)$. Combined with the natural constraint that we cannot transmit more packets than available ($u_n \leq x_n$), we propose the following log-linear scheduler:

$$u_n = \min\left(x_n, \lfloor \log(\kappa x_n) \rfloor\right). \tag{15}$$

The parameter κ of the log-linear scheduler is chosen to meet the delay bound. For buffer states greater than $L - M$, the log-linear scheduler transmits at least $x_n - (L - M)$ packets to prevent buffer overflows. The log-linear scheduler greatly simplifies scheduler design since it requires only one parameter κ. The delay and power using the log-linear scheduler is calculated using (9) and (10), respectively. The delay–power region of the log-linear scheduler is obtained by monotonically increasing κ. In all our simulations, the log-linear performance is close to the optimal scheduler performance; further results are given in Section III-E.

The value of κ to achieve a certain average delay depends on the arrival distribution of the source. We now propose a simple adaptive algorithm that computes the value of κ dynamically to achieve any given average delay constraint based only on the knowledge of the mean arrival rate. In the adaptive scheduler, κ_n is a function of time and is updated in every time slot as follows:

$$\kappa_n = \kappa_{n-1} + \Delta_\kappa \left(\hat{D}_n - D_0 \right) \tag{16}$$

where $\hat{D}_n$ is the sample average delay given by the ratio of the sample average buffer length and arrival rate. The performance of the adaptive scheduler is shown to numerically converge to the performance of a log-linear scheduler that has full knowledge of arrival statistics in Section III-E.

C. Approximate Delay–Power Relation

In this section, we derive an approximate closed-form relationship between the average transmit power P_{avg} and the average delay D_{avg} in an AWGN channel. The derivation is based on two simplifying assumptions: the packets are infinitely divisible (fluid model) and the output distribution of the scheduler is Gaussian. The fluid model allows us to divide packets arbitrarily, and thus avoid the discrete nature of scheduler decisions, which is a significant reason for intractability of closed-form analysis of the optimal scheduler.

As discussed in Section III-A, for large D_{avg}, the optimal scheduler transmits at rates close to average arrival rate λ if buffer overflows can be avoided. For large delays, the queue length x_n varies around the average queue length $\mathbb{E}\{x_n\}$. The scheduler action $u_n(x_n)$ can thus be approximated around $\mathbb{E}\{x_n\}$ by the following linear relation:

$$u_n \approx \mu x_n + \nu. \tag{17}$$

For a stable system, the average arrival rate should equal the average departure rate and hence $\mathbb{E}\{u_n\} = \lambda$. Taking expectations on (17) we find $\mu \mathbb{E}[x_n] = -\nu + \lambda$. The slope of the scheduler μ at $\mathbb{E}\{x_n\}$ is approximated by the slope of log-linear scheduler $\mu = 1/\mathbb{E}\{x_n\}$.

The variance of the output process u_n, denoted by σ_u, is computed as follows. From (2), (17)

$$\mathbb{E}[x_{n+1}^2] = \mathbb{E}[x_n^2] + \mathbb{E}[u_n^2] + \mathbb{E}[a_n^2] + 2\lambda(\mathbb{E}[x_n] - \lambda) \\ - 2\mathbb{E}[x_n(\mu x_n + \nu)].$$

From stationarity of x_n

$$\mathbb{E}[x_{n+1}^2] = \mathbb{E}[x_n^2]$$

and after algebraic manipulations, we can find σ_u^2 as

$$\sigma_u^2 = \frac{\sigma_a^2 \mu}{2 - \mu} = \frac{\sigma_a^2}{2\mathbb{E}[x_n] - 1} \tag{18}$$

---

[3]It is our conjecture that for any average delay ($D_{\text{avg}} \geq 1$), this is the input distribution requiring highest average transmit power. In other words, the *On–Off* arrival process requires the highest transmit power at any delay in an AWGN channel among all arrival processes with the same average and finite maximum arrival rate.

[4]Note that the optimal scheduler is not unique for $M < L$ and if the schedulers start from the zero buffer state.

where σ_a^2 is the variance of the arrival process $\{a_n\}$. Using the Gaussian approximation for the distribution of $u_n \sim \mathcal{N}(\lambda, \sigma_u^2)$, the average transmit power is

$$\mathbb{E}\{P_n\} = \frac{1}{\sqrt{2\pi\sigma_u^2}} \int_{-\infty}^{\infty} P_n(u_n, 1) e^{-\frac{(u-\lambda)^2}{2\sigma_u^2}} \, du_n.$$

$$= \sigma^2 \left(e^{\lambda + \frac{\sigma_u^2}{2}} - 1 \right)$$

$$= \sigma^2 \left(e^{\lambda + \frac{\sigma_a^2}{4\lambda D_{\mathrm{avg}} - 2}} - 1 \right) \qquad \text{(from (18))}$$

$$\approx \sigma^2 \left(e^{\lambda + \frac{\sigma_a^2}{4\lambda D_{\mathrm{avg}}}} - 1 \right) \qquad \text{(for large } \lambda D_{\mathrm{avg}}\text{)}.$$

Thus,

$$\log\left(1 + \frac{P_{\mathrm{avg}}}{\sigma^2}\right) = \lambda + \frac{\sigma_a^2}{4\lambda D_{\mathrm{avg}}}. \tag{19}$$

The effect of random source arrivals is clearly highlighted in (19) by the factor $\frac{\sigma_a^2}{\lambda}$. Sources which exhibit large variations in their arrivals compared to their mean rate and are "more" bursty, have large $\frac{\sigma_a^2}{\lambda}$. For more bursty sources, even small increase in D_{avg} allows considerable power savings; practical examples of such sources include web and e-mail traffic. Asymptotically, as $D_{\mathrm{avg}} \to \infty$, the high delay approximation gives

$$\log\left(1 + \frac{P_{\mathrm{avg}}}{\sigma^2}\right) = \lambda$$

which is the same as the Shannon limit. As $D_{\mathrm{avg}} \to \infty$, the random arrivals of the source are completely smoothened and hence well-known information-theoretic analysis applies.

D. Scheduling in Fading Channels

In the preceding sections, we showed that time scheduling packets from a source that produces variable number of packets every time-slot results in significant power reduction, even in AWGN channels. An analogous problem to scheduling due to source burstiness is the one where scheduling is performed due to time-varying channel conditions. To further emphasize the fact that channel time variations lead to a nontrivial scheduling problem under delay constraints, we consider transmission of a constant rate source over a fading channel.[5]

Note that scheduling is equivalent to power control due to our emphasis on reliable communication, and their relation is defined by (6). Power control in fading channels has been an area of active interest (see [29] for a review). We illustrate, by an example, that the proposed method of dynamically changing the transmission rate and power, can lead to power savings in finite-state fading channels with increasing delay. For a delay bound, $D_{\mathrm{avg}} = 1$ time slot, packet scheduling reduces to conventional power control. As the delay bound goes to infinity, the delay-bounded scheduling leads to maximal mutual information power control [1].

For simplicity, we consider a two-state fading channel in the remainder of this subsection although our results are directly applicable to any channel with finite number of states. Consider a source that produces a constant number of packets every time slot, being transmitted over a two-state channel, with $A_n \in \{1, 10\}$ and the transition probabilities T between the two channel states given by

$$\boldsymbol{T} = \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix} \tag{20}$$

where $T_{i,j}$ is the probability of transitioning from channel state $A_n = i$ to $A_{n+1} = j$. The corresponding stationary probabilities of being in the two channel states are $0.25 \ (A_n = 1)$ and $0.75 \ (A_n = 10)$.

At a delay of one time slot, the channel transmits 25% of the packets when $A_n = 1$ and the remaining packets when $A_n = 10$. As the delay increases to 4.7 time slots, only 2% of the packets are transmitted when $A_n = 1$, a savings of 9 dB in power over case when delay equals one time slot. As the delay tends to infinity, the average number of packets transmitted in each state is given by the time water-filling solution, and is characterized in the following proposition.

Proposition 12 (Time Water-Filling): For a two-state fading channel, power minimizing transmission rates $\{r_1, r_2\}$ for the two states, with no delay constraint, are given by

$$r_1 = \left[\lambda + 2p_2 \log\left(\frac{A_1}{A_2}\right) \right]^*$$

$$r_2 = \left[\lambda + 2p_1 \log\left(\frac{A_2}{A_1}\right) \right]^*.$$

The stationary probability of being in channel state i is denoted by p_i and

$$[y_i]^* = \begin{cases} \frac{\lambda}{p_i}, & \text{if } y_i > \frac{\lambda}{p_i} \\ y_i, & \text{if } 0 \le y_i \le \frac{\lambda}{p_i} \\ 0, & \text{if } y_i < 0. \end{cases}$$

Proof: Follows from (6) and standard Lagrangian techniques.[6]   □

Applying Proposition 12 to the channel given above suggests that at infinite delay $r_1 = 0$ and $r_2 = 4$. We see that these values of r_i are reflected in the small fraction of packets that are transmitted in state $A_n = 1$ at a delay of 4.7 time slots.

In practical systems, both the source arrival rate and channel state are time varying; hence, scheduling becomes *doubly* important. Our formulation is applicable, in a straightforward manner, to the case where both source and channel are time varying. The power savings consists of two components—one due to source burstiness and the other due to channel variations. The total gain depends on the following factors: the extent of source burstiness σ_a^2/λ, the delay bound D_0, the amount of channel variations $\sigma_c^2/\mathbb{E}\{|A_n|^2\}$, and the stationary probabilities of different channel states. The larger the σ_a^2/λ, the larger is the savings in power with increasing delay because the departure process u_n is well smoothened out. Analogously, the larger the $\sigma_c^2/\mathbb{E}\{|A_n|^2\}$, the larger difference between the signal-to-noise ratios (SNRs) of the two states leads to decreased average power with increasing delays.

---

[5]This parallels our previous discussion on transmission of a bursty source over an AWGN channel. Preliminary results on bursty source-fading channel duality can be found in [22].

[6]These values of r_i are derived assuming that fractionally sized packets can be transmitted; packet integrity constraint must be imposed on top of these rate values to find integral transmission rates.
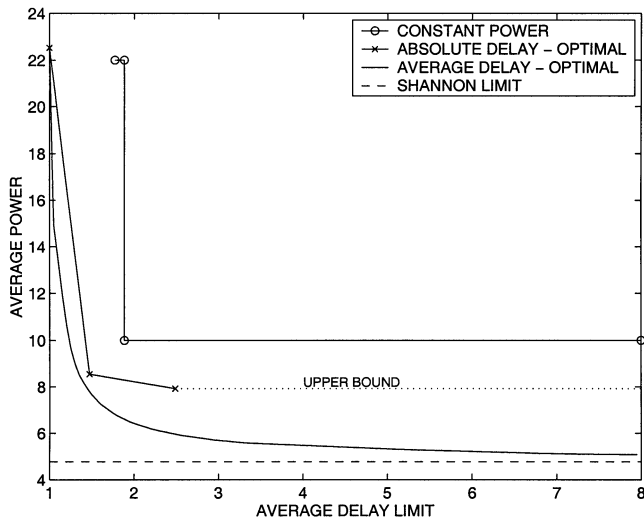
Fig. 3. Achievable delay–power region for AWGN channel. The points marked by x represent the lowest average power achievable under absolute delay bounds of one, two, and three time slots. (For all plots buffer size $L = 50$, maximum arrival $M = 6$, uniform arrival distribution.)

E. Numerical Results

In this subsection, we present numerical results to study the performance of the proposed schedulers. We first compare the performance of the optimal scheduler with the proposed suboptimal log-linear scheduler, and the closed-form approximation derived in Section III-C. Spectral analysis is also used to analyze the behavior of the optimal and log-linear schedulers. An example for four-state fading channel is also presented. Finally, we make some empirical observations about the optimal schedulers based on our numerical simulations.

The plot of the minimal transmission power versus average delay bound is shown in Fig. 3. For all delay–power pairs above the solid curve, schedulers exists which achieve that delay and power. Also, no scheduler can guarantee delays and powers in the region below the solid line. Note that the required power decreases quickly as we relax the delay constraint from one to two time slots. The substantial reduction in power for small increases in delay beyond one time slot was observed in all our simulations.

To emphasize the importance of proposed scheduling, Fig. 3 also shows the performance of the following *constant-power* scheduler:

$$u_n = \begin{cases} w, & \text{if } x_n \geq w \\ 0, & \text{if } x_n < w \end{cases} \tag{21}$$

where the value of w determines the power and delay achieved. The constant power scheduler either sends a fixed number of packets or nothing. Note that only certain values of w can guarantee zero outage, and hence the constant power curve in Fig. 3 has a step-like behavior. The difference between the optimal curve and constant power curve gives an indication of the gains in the proposed variation of the transmit power based on the queue state.

Also shown in Fig. 3 is the minimum power required to transmit packets with no delay constraints, indicated as

Shannon limit on the plot. For AWGN, the Shannon limit is given by

$$P_{\text{Shannon}} = \sigma^2 \left[(1 - \lambda + \lfloor \lambda \rfloor)(e^{\lfloor \lambda \rfloor} - 1) \right.$$
$$\left. + (\lambda - \lfloor \lambda \rfloor)(e^{\lfloor \lambda \rfloor + 1} - 1) \right].$$

Thus, as delay increases unboundedly, the Shannon limit is achieved by the policy which transmits either $\lfloor \lambda \rfloor$ or $\lfloor \lambda \rfloor + 1$ packets in each transmission, with probabilities $(1 - \lambda + \lfloor \lambda \rfloor)$ and $(\lambda - \lfloor \lambda \rfloor)$, respectively.

The excess power required by delay-bounded schedulers over the Shannon limit is an indication of the loss due to finite delay requirements. As expected, the power required by the delay-bounded scheduling approaches the Shannon limit as the delay increases. From Fig. 3, which is indicative of the typical performance we obtained, we observe the encouraging result that the rate of convergence to the asymptotic limit is fairly fast. We briefly note the fact that to achieve arbitrarily large delays, the buffer size should be increased accordingly, which directly follows from (9). However, large amount of buffering may not be required in practical systems, to achieve significant reduction in power consumption by traffic smoothening, because the delay only needs to increase to around two time slots. The fourth curve in Fig. 3 shows the average power versus average delay performance of the scheduler with absolute delay guarantees, derived and discussed in Section IV.

Our scheduler formulation which bounds average delays, does not depend on the order in which packets in the buffer are serviced. So, FCFS and LCFS have similar power requirements for the same average-delay bounds. However, the distribution of actual delays experienced by the packets shows completely different characteristics in FCFS- and LCFS-based schedulers. For example, to achieve an average delay of two time slots using FCFS policy, the distribution of packet delays was almost uniform and had a small range. In contrast, with LCFS, the distribution of delay was highly skewed; a large number of packets experienced a delay of one time slot while a few packets experienced delays greater than 900 time slots.

The performance of the log-linear and adaptive log-linear schedulers are compared with the optimal scheduler in Fig. 4. The proposed log-linear scheduler performed close to the optimal scheduler in all our simulations, thereby verifying the "equalization" intuition behind the log-linear scheduler. For the adaptive log-linear performance shown in Fig. 4, we choose the parameters $\Delta_\kappa = 0.001$ and initial value for κ_n as $\kappa_0 = \frac{10}{D_0}$. The power–delay performance is plotted after 10 000 time slots. From Fig. 4, we find that the proposed adaptive log-linear scheduler achieves near-optimal performance by using only knowledge of average arrival rate. In most cases we found that the adaptive κ reaches within 5% of the asymptotic value in 500–2000 samples depending on the average delay bound D_0.

The traffic smoothening effect due to increasing delay can be seen in Fig. 5. In Fig. 5, the power spectral density of the arrival ($\{a_n\}$) and departure ($\{u_n\}$) processes is shown. Since th arrival process is assumed to be i.i.d., its power spectrum is white. The smoothening effect due to buffering is evident from the low-pass nature of the power spectrum of the output process.
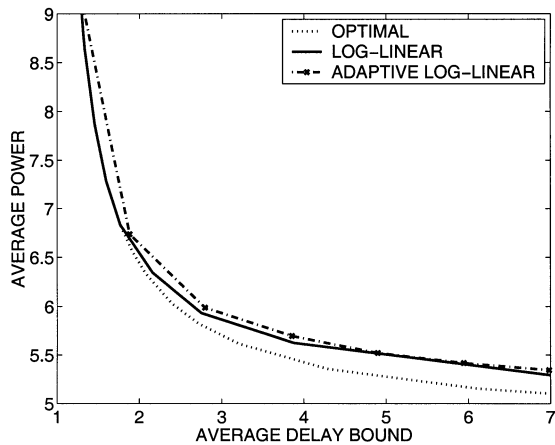
Fig. 4. Comparison of log-linear and adaptive log-linear schedulers with corresponding optimal scheduler. (Buffer length $L = 50$, maximum arrival $M = 6$, uniform arrival distribution.)
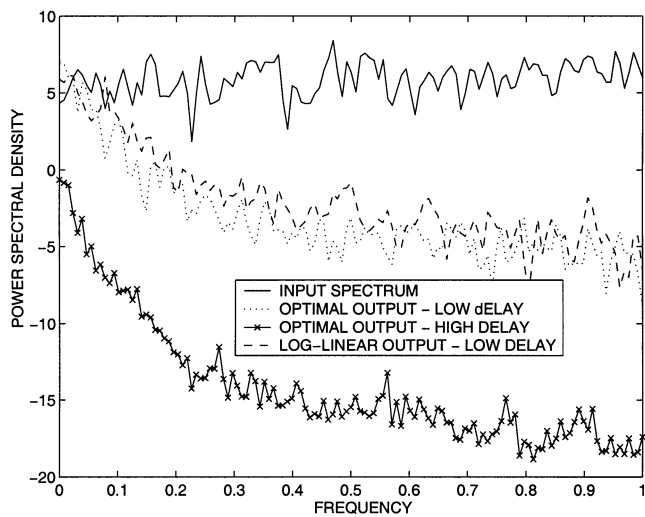

Fig. 5. Power spectral density of input and output process for different delays. (Buffer length $L = 5$, maximum arrivals $M = 6$, uniformly distributed arrivals.) Delays of output process are 2.2 time slots (low delay) and 7.9 time slots (high delay).

We observed that, as the delay increases, the bandwidth of the departure process decreases. In essence, the optimal scheduler acts like a low-pass filter. In contrast, in a fading channel, when the arrival process is constant rate (power spectrum is a delta function), the output process from the queue is variable and has higher frequency components [22].

Finally, in Fig. 6, we compare the large delay approximation (19) with the performance of the optimal scheduler. The axes in Fig. 6 are $\log(1 + \mathbb{E}\{P_n\})$ and $1/\mathbb{E}\{D_n\}$. The arrival distribution was assumed to be uniformly distributed between $[0, 6]$ with mean three packets per time slot. The high-delay approximation was derived with an assumption that the departure traffic is Gaussian, an assumption which is least applicable for low delays. For small delays, the transmitted power is dependent on the input distribution, and hence Gaussian distribution-based high-delay approximation has a large error in this delay regime. But for large average delays (beyond two time slots), the optimal scheduler follows the linear trend, as predicted by the high-delay
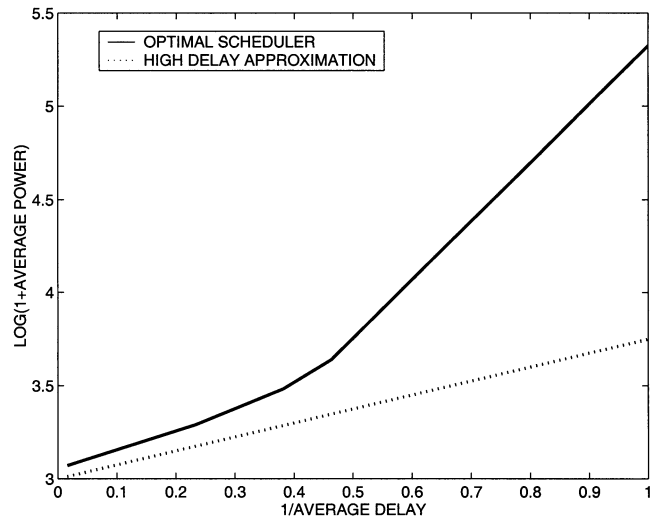

Fig. 6. Plot of logarithm of average power versus inverse of average delay. (Buffer length $L = 250$, maximum arrivals $M = 6$.) The actual powers and delays are compared to high delay approximation (19).
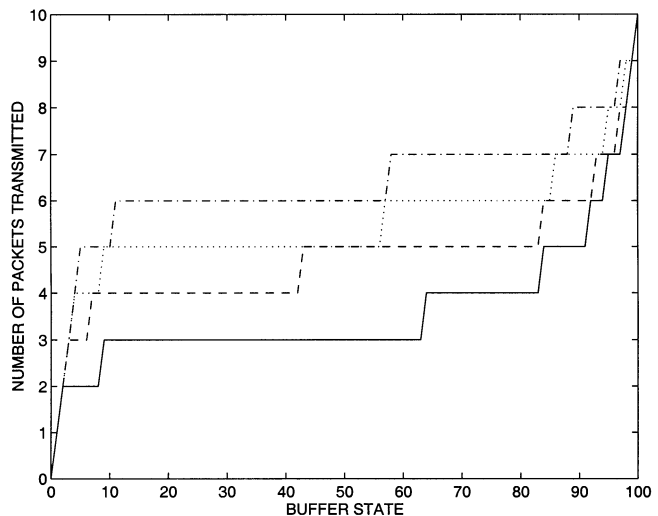

Fig. 7. Schedulers for fading channel. (Buffer size $L = 100$, maximum arrival $M = 10$, uniform arrival distribution, four-state fading $A = [2\,4\,6\,10]$, transition probability between states given in (22).).

approximation. In fact, due to smoothening effect, the effect of input distribution is minimal for moderate to large delays.

An example of the scheduler action for a four-state fading channel with constant rate traffic is shown in Fig. 7. The channel amplitudes in different states are $A_n \in \{2, 4, 6, 10\}$. The transition probabilities between the different states are given by

$$\boldsymbol{T} = \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.4 & 0.1 & 0.4 \\ 0.15 & 0.15 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{bmatrix}. \tag{22}$$

The lowest curve in Fig. 7 corresponds to $A_n = 2$ and the highest to $A_n = 10$. The time water-filling nature of the scheduler, for finite delays, follows from the Fig. 7. The scheduler always transmits more packets in good channel states, and less in poor channel conditions.

Based on our numerical study, we state some empirically observed properties of the optimal deterministic schedulers.

1) In an AWGN channel, u_n is a nondecreasing function of x_n, i.e., more packets are transmitted when there are more packets in the buffer. In fading channels, conditioned on each state A_n, u_n is a nondecreasing function of x_n.

2) In an AWGN channel, the optimal scheduler is work conserving, i.e., $u_n = 0 \Leftrightarrow x_n = 0$. In a fading channel, we found situations in which the scheduler was not work conserving. For large-delay bounds, if A_n in the *bad* channel state is much worse than in *good* state, u_n was zero when x_n was nonzero and the channel was in the bad state. This scheduler action in fading channels is in agreement with the time water-filling solution obtained by maximizing mutual information [1] to achieve highest possible long-term throughput.

3) In an AWGN channel, among all deterministic schedulers, the increase in u_n is less than or equal to increase in x_n. In other words, for all $a > 0$ and x_n

$$|u_n(x_n + a) - u_n(x_n)| \le a.$$

This property is reflected in the fact that u_n only increases in increments of 1 for each unit increment in buffer size.

4) In an AWGN channel, the maximum number of packets transmitted in any time slot equals the maximum number of arrivals in one time slot, i.e., $u_n \le M$.

5) In a finite-state fading channel, for a fixed buffer state more packets are transmitted in channel with higher A_n, i.e., for a fixed x_n, u_n is a nondecreasing function of A_n.

6) In an AWGN channel, we observed that the entropy of departure process $\{u_n\}$ was higher, lower or equal to entropy of arrival process $\{a_n\}$ depending on delay, power, and arrival distribution.

   - When the delay bound equals 1, $u_n = a_{n-1}$ and so arrival and departure process have the same entropy.
   - When the delay bound asymptotically goes to infinity, $u_n \Rightarrow \lambda$ and so entropy of the departure process is asymptotically zero which is lower than the entropy of a bursty arrival process.
   - Consider an *On–Off* arrival process with probability of being in the *On* state equal to v, so its entropy equals $-v \log(v) - (1-v) \log(1-v)$. We observed that for this arrival process, and delays slightly higher than 1, the departure process has entropy higher than the arrival process. The entropy increasing property of certain queuing systems is investigated in [30].

The properties in AWGN channels are supported by Fig. 2. The properties for fading channels are empirically clear from Fig. 7 which shows the scheduler action u_n as a function of x_n and A_n.

Recall that in our system we have assumed all packets to be of fixed size and indivisible, i.e., whole packets have to be transmitted completely within one time slot. Note that if fractional packets can be transmitted in any time slot, then the power required can be further lowered compared to the proposed limits. The potential power reduction in transmitting fractional packets is easily verified for large delays when the average arrival rate λ is not an integer, say, 3.5 packets/time slot. If packets are not divisible, the Shannon limit is $0.5\sigma^2 \left[\left(e^3 - 1 \right) + \left(e^4 - 1 \right) \right]$. In contrast, when fractional packets cane be transmitted, the Shannon limit is given by $\sigma^2 \left(e^{3.5} - 1 \right)$, which is lower than the one with no fractionalization, again due to the convexity of the exponential function. Preliminary results on scheduling with fractionalization are given in [22].

IV. SCHEDULERS GUARANTEEING ABSOLUTE DELAY BOUNDS

For the systems using schedulers which bound the average packet delay, the actual delay suffered by a packet depends on the service order, like FCFS or LCFS, and can be potentially unbounded. Large packets delays are highly probable in fading channels, where the scheduler is not guaranteed to be work conserving. For delay-sensitive multimedia applications like streaming video, bounded packet delays are essential to maintain appropriate quality of service. In this section, we derive power minimizing zero-outage schedulers with a bounded absolute packet delay.

A. Problem Formulation

Schedulers designed according to the average delay constraint (4) are not constrained to be work conserving. However (as stated in Section III-E), in all our experiments, the optimal scheduler was work conserving for Gaussian channels. Thus, with FCFS service discipline, no packet will suffer arbitrarily large delays in the average delay optimal scheduler, which implies that the buffer size L is a trivial bound on the absolute packet delay. Thus, limiting the class of schedulers to the set of FCFS work-conserving schedulers could potentially be used to obtain schedulers with absolute delay guarantees. We study an alternate approach, in which the system knows the arrival time stamp of each packet and uses that information to make its scheduling decisions.

With the knowledge of packet time stamps, the state of the queue not only includes queue length but also complete timing information of the queued packets. We state, without proof, that an average power-minimizing scheduler, with bounded absolute delay D_{abs}, does not need information about packet arrivals $D_{\text{abs}} + 1$ or more slots prior to the current slot.[7] Thus, defining the state of the queue B_n as a D_{abs}-dimensional vector suffices

$$B_n = \{(b_{1,n}, b_{2,n}, \dots, b_{D_{\text{abs}},n}) : 0 \le b_{i,n} \le M\} \quad (23)$$

where $b_{i,n}$ represents the number of packets that arrived during the $(n-i)$th slot that have not been transmitted before time slot n. We assume an FCFS service discipline and find the number of packets to be transmitted u_n to be a function of B_n. Note that the total number of *queue states* in the system is given by $(M+1)^{D_{\text{abs}}}$. Similar to our treatment of average delay bounded schedulers, all queue states might not have finite probability of occurring depending on the scheduler action.

---

[7]Intuitively, if a scheduler guarantees all packets leave within D_{abs} time slots of their arrivals, during any time slot n the number of buffered packets which arrived earlier than time slot $n - D_{\text{abs}} - 1$ equals 0.

If the buffer length L is greater than $(M+1)D_{\mathrm{abs}}$ then any scheduler which guarantees absolute delay bound D_{abs} will have no buffer overflow. If $L < (M+1)D_{\mathrm{abs}}$, necessary and sufficient conditions for zero buffer overflow similar to Section II-C exist and hence are not stated here. Since the class of zero outage schedulers is smaller for the case $L < (M+1)D_{\mathrm{abs}}$ than for the case $L \geq (M+1)D_{\mathrm{abs}}$, the average power required by a scheduler with the same D_{abs} will be greater in the first case as compared to the latter. We will assume $L \geq (M+1)D_{\mathrm{abs}}$ throughout the remainder of this section.

The set of all schedulers satisfying the absolute delay bound D_{abs} is denoted by $\Omega(D_{\mathrm{abs}})$. The optimization problem analogous to (4) can be expressed as

$$\min_{\Omega(D_{\mathrm{abs}})} \lim_{n\to\infty} \mathbb{E}[P_n] \tag{24}$$

where $d_{\max}$ represents the maximum delay experienced among all packets.

B. Optimal Scheduler and its Properties

As in Section III, u_n can be a deterministic or a randomized stationary function of B_n. In Section III-A, we proved that the vertices of the delay power region are achieved by deterministic schedulers. Since packets cannot be divided and transmitted, all possible values of D_{abs} have to be positive integers, i.e., $D_{\mathrm{abs}} \in \mathbb{Z}^+$. We show that for the case of absolute delay constraints, we need only deterministic schedulers since deterministic schedulers achieve the lowest average power for *all* values of D_{abs} using a corollary to the following theorem. We first show that the power required using a randomized scheduler satisfying the absolute delay bound D_{abs} equals the convex combination of powers of deterministic schedulers satisfying the same delay bound D_{abs}.

Theorem 13 (Convexity of Randomized Schedulers): Let $\hat{Q}_{D_{\mathrm{abs}}}$ be the set of all zero-outage deterministic schedulers that guarantee an absolute delay D_{abs}. If a zero-outage randomized scheduler ζ also guarantees absolute delay D_{abs}, then the average power is

$$P_{\mathrm{avg}}(\zeta) = \sum_{i=1}^{G} \eta_i P_{\mathrm{avg}}(v_i) \tag{25}$$

where $v_i \in \hat{Q}_{D_{\mathrm{abs}}}$ and $G < |\hat{Q}_{D_{\mathrm{abs}}}|$. Also, for all i, $\eta_i \in [0,1]$ and $\sum_{i=1}^{G} \eta_i = 1$.

Proof: Follows readily from time-sharing arguments and is summarized in Appendix I.   □

The following corollary shows that the solution of (24) is always a deterministic scheduler.

Corollary 14 (Optimal Scheduler: Absolute Delay Bound): Among all schedulers which guarantee an absolute delay bound D_{abs}, the one requiring least power is always deterministic.

Proof: Follows readily from Theorem 13 and realizing that all $\eta_i \geq 0$.   □

As a result of Corollary 14, the constraint set $\Omega(D_{\mathrm{abs}})$ in the optimization problem (24) can be further reduced to include only the set of all deterministic schedulers satisfying absolute delay bound D_{abs}. Now, to solve the optimization problem (24), we again use the VIA. Notice that with the increasing value of D_{abs}, the state complexity of VIA grows exponentially. Thus, for illustration, we compute the optimal policies for $D_{\mathrm{abs}} = 1, 2, 3$ and show the minimal achieved power as a function of their average (not absolute) delays in Fig. 3. The three points marked by x correspond to absolute delay guarantees of one, two, and three time slots. Beyond the third point corresponding to $D_{\mathrm{abs}} = 3$, the computational complexity of VIA becomes prohibitive. Thus, only an upper bound on required power is shown, which is obtained by using the scheduler designed for $D_{\mathrm{abs}} = 3$. As can be seen from the figure, a policy guaranteeing an absolute delay requires more transmit power to achieve the same average delay as the one which does not give any absolute delay guarantees.

Note that although the queue state represented using B_n carries more information than x_n, its use in Section III was avoided for two reasons. First, the number of states increases exponentially with D_{abs} and the VIA quickly becomes intractable. Second, the Little's theorem [24] gives a simple relationship between x_n and D_{avg} whereas there is no such simple relation between x_n and D_{abs}.

Having an average delay bound only fixes the mean of the distribution of actual packet delays. In contrast, having an absolute delay bound controls the support of the delay distribution. Both mean and the support of the delay distribution can be controlled by the following optimization:

$$\min_{\Omega(D_{\mathrm{abs}})} \mathbb{E}[P_n + \delta x_n]. \tag{26}$$

Different values of δ yield different D_{avg} under the same D_{abs}. For $\delta = 0$, (26) reduces to (24), which transmits least power for a given absolute delay bound. As δ increases, the required power also increases while the average delay decreases for a fixed absolute delay bound. Eventually, as $\delta \to \infty$, (26) would yield a scheduler which flushes all the packets in the buffer at all instants, i.e., the average (and absolute) delays would equal one. Note that we require all packets to have a delay of less than D_{abs}. In practice, this maybe too stringent and delay violations with some probability can be tolerated [31].

V. ISSUES IN IMPLEMENTING SCHEDULERS IN PRACTICAL SYSTEMS

The proposed scheduler structure requires a physical layer implementation which can adaptively change the transmission rate and power from one time slot to another, a feature which can be supported in systems [32] like EDGE (TDMA) and 1Xtreme (CDMA). In this section, we show that the power gains predicted using information-theoretic power control (6) can be obtained by using a simple multirate transmission scheme. Our aim in this section is to show, by an example design, the tradeoff between packet delay and subsequent minimum required transmission power; no particular optimality is claimed in the example design.

TABLE I
THE POWER REQUIRED TO ACHIEVE AN AVERAGE PACKET ERROR OF 0.05
USING A CONVOLUTIONAL CODE WITH VARIABLE QAM.

# Packets/time-slot	# Bits/modulation-symbol	Average power (dB)
1	2	5.5
2	4	12
3	6	22.5

In our system formulation, the receiver is unaware of the chosen number of packets u_n and hence the transmitter needs to send this information which is referred to as *protocol information* in [33] to the receiver. In Section V-B, we show how this protocol information can be obtained from the received signal Y_n, thus obviating the need for side information from the transmitter to the receiver.

A. Simple Multirate Transmission Scheme

The information-theoretic notion of reliability, used in the previous sections, requires coding over large time intervals to achieve arbitrarily small probability of error. For finite-length time slots, only a finite packet error is achievable. Thus, an arbitrarily small packet error will be replaced by a target finite packet error rate.

We assume the size of each packet to be 25 bits and the channel bandwidth and transmit pulse shape are such that 25 symbols can be transmitted in each time slot. The transmitter can choose to transmit zero to three packets. For every choice of number of packets, the data bits of the all packets are jointly encoded as a one long packet, using a convolutional encoder of rate $1/2$ with constraint length 3 and generator matrix $[4\ 7]$ [27]. The output of the convolutional encoder is modulated using a variable-rate QAM depending on u_n according to Table I. For example, to transmit two packets per time slot, the scheme needs to transmit 100 coded bits (2 packets $\times$ 25 bits/packet $\times$ 2 coded bits/information bit) using 25 symbols, which implies 4 bits/symbol; hence, we choose a simple rectangular 16-QAM constellation in this case. For $\sigma^2 = 1$, the power required to achieve a packet error rate of 0.05 in an AWGN channel is given in Table I.[8]

For simulations, we use an *On–Off* arrival process which produces either zero or three packets in any time slot with equal probability, thus has an average packet arrival rate $\lambda = 1.5$. We consider a buffer of length $L = 25$ packets. To transmit all packets at error rate of 0.05, with an average delay of one time slot requires an average transmit power of 19.5 dB. We use the optimal scheduling technique introduced in Section III to find u_n as a function of the queue state x_n for different delays. Using the optimal scheduler to achieve an average delay bound of two time slots, the average power required is 10.02 dB, giving a substantial saving of almost 9.5 dB over the case of the delay of one time slot. Again, the savings are possible due to the strictly convex relation between the power and the rate

of transmission. A sample histogram of the number of packets transmitted in each time slot for an average delay of two time slots was found by calculating the number of packets transmitted in each time slot over 50 000 time slots. The histogram reveals that 0 packets/time slot are transmitted with probability 0.0549, 1 packet/time slot is transmitted with probability 0.3825, and 2 packets/time slot are transmitted with probability 0.5625. Interestingly, 3 packets/time slot are never transmitted in our simulations which translates into significant power savings when compared to the delay of one time slot.

B. Power Based Detection of Transmission Rate

In a multirate system, where the rate of transmission is controlled by the transmitter, the receiver should have the knowledge of instantaneous transmission rate to achieve target error rates. Following the terminology in [33], the information regarding instantaneous rate is labeled as *protocol information*.[9] In our formulation, the instantaneous rate information can be extracted from the average received signal power. Since the transmission power is chosen proportional to the transmission rate, in an AWGN channel the received signal power is directly related to the transmission rate. Below, we derive a rate detector for the multirate system described in Section V-A.

Let hypothesis H_i represent the transmission of i packets per time slot. For clarity of presentation, the subscript n representing the time index is suppressed. For an AWGN channel, the received signal Y_{ij} is given by $Y_{ij} = X_{ij} + \varepsilon_{ij}$ with $i = 1, 2$ and $j = 1, 2, \ldots, N$, where X_{1j} and X_{2j} represents the jth symbol transmitted in the I and Q channels, respectively. Recall that ε_{ij} is assumed to be zero-mean circularly symmetric i.i.d. Gaussian noise with variance σ^2. The number of symbols in each time slot equals N. Denote the matrix of received signal by $Y = [Y_{ij}]$, the likelihoods [34] of the different hypothesis are given by

$$p(Y|H_0) = \prod_{i=1}^{2} \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{Y_{ij}^2}{2\sigma^2}}$$

$$p(Y|H_1) = \prod_{i=1}^{2} \prod_{j=1}^{N} \frac{1}{2\sqrt{2\pi\sigma^2}} \left[e^{-\frac{(Y_{ij}-d_1)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}+d_1)^2}{2\sigma^2}} \right]$$

$$p(Y|H_2) = \prod_{i=1}^{2} \prod_{j=1}^{N} \frac{1}{4\sqrt{2\pi\sigma^2}} \left[e^{-\frac{(Y_{ij}-d_2)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}+d_2)^2}{2\sigma^2}} \right. $$
$$\left. + e^{-\frac{(Y_{ij}+3d_2)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}-3d_2)^2}{2\sigma^2}} \right]$$

$$p(Y|H_3) = \prod_{i=1}^{2} \prod_{j=1}^{N} \frac{1}{8\sqrt{2\pi\sigma^2}} \left[e^{-\frac{(Y_{ij}-d_3)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}+d_3)^2}{2\sigma^2}} \right.$$
$$+ e^{-\frac{(Y_{ij}+3d_3)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}-3d_3)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}+5d_3)^2}{2\sigma^2}}$$
$$\left. + e^{-\frac{(Y_{ij}-5d_3)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}+7d_3)^2}{2\sigma^2}} + e^{-\frac{(Y_{ij}-7d_3)^2}{2\sigma^2}} \right].$$

The minimum distance between constellation points equals $2d_1, 2d_2,$ and $2d_3$ under hypothesis $H_1, H_2,$ and H_3, respec-

---

[8]An alternative is to change the coding rate assuming the modulation (number of constellation points) is fixed, say 4-QAM. To transmit one, two, or three packets per time slot, coding rates of $1/6, 1/3,$ or $1/2$ could be used, respectively. Alternately, different puncturing of the same code can be used. Depending on the coding rate, the minimum distance between signal points is changed to maintain the same packet error rate.

[9]A lower bound on the protocol information is given by the entropy of the departure process $\{u_n\}$ and an upper bound is given by $\lceil \log_2(M+1) \rceil$.

tively. The optimum maximum-likelihood (ML) detector for the instantaneous rate is then given by

$$\widehat{u}_n = \arg\max_i p_{Y|H_i}(Y|H_i). \qquad (27)$$

The probability of bit error under hypothesis H_i depends largely on d_i and to maintain same symbol error rate under each hypothesis, we need to make $d_1 = d_2 = d_3$. However, to maintain the same packet error rate, we need to choose different d_i for a given noise variance. We tested the performance of the ML detector (27) with the d_i chosen to guarantee packet error rates of 0.1 in each hypothesis. We transmitted variable number of packets in each time slot for $100\,000$ time slots and the ML detector (27) accurately detects the instantaneous rates in each time slot. Hence, the error is lower than 10^{-5}. The ML detector had error rates lower than 10^{-5} whenever d_i (and, correspondingly, transmit powers) were chosen to have packet error rates lower than 0.1.

Thus, the complete receiver for the proposed multirate system will first detect the number of transmitted packets, and then use the appropriate decoder to recover the transmitted packets. An alternate method is to encode the number of packets in the header of the packet (using a fixed code), decode the header first, and then appropriately choose the right decoder for the rest of the data.[10] The alternative is better suited where u_n can range over a large set, and hence makes (27) computationally hard.

VI. DISCUSSION

A. Timing Capacity

The idea of sending information in the arrivals times of the packets in addition to the packet contents was introduced and formalized in [35]. In our system, we can use the variable number of packets in each time slot to convey extra information in addition to packet contents. If we design a scheduler to maximize this extra information, we would need to maximize the entropy of $\{u_n\}$ while maintaining delay bounds and buffer overflow requirements. Obviously, sending such covert information comes at the cost of increasing the average power of transmission.

We present a simple scheduler that can be used to send covert information. Let $\{v_i\}$ represent in binary form the covert information to be transmitted. Choose u_n as follows:

$$u_n = \begin{cases} 0, & \text{if } x_n < 2 \\ 1, & \text{if } 2 \le x_n \le L - M \text{ and } v_i = 0 \\ 2, & \text{if } 2 \le x_n \le L - M \text{ and } v_i = 1 \\ x_n, & \text{if } x_n > L - M. \end{cases} \qquad (28)$$

The power-minimizing scheduler transmits at rates equal to average rates (if delay and buffer requirements are satisfied) and hence minimizes the covert information that could be transmitted.

B. Packet Reordering

If multiple users are competing for the wireless resource, scheduling the users becomes critical in maximizing system performance. Service disciplines like FCFS, last come first serve

(LCFS), EDF, and WFQ address the problem of resource allocation to different users.

Our scheduler formulation which bounds average delays, does not depend on the order in which packets in the buffer are serviced. So, FCFS and LCFS have similar power requirements for the same average delay bounds. However, the distribution of actual delays experienced by the packets shows completely different characteristics in FCFS- and LCFS-based schedulers. Using FCFS policy, the distribution of delays was almost uniform and had a *small* range. In contrast, with LCFS, the distribution of delay was highly skewed; a large number of packets experienced a delay of one time slot while a few packets experienced delays greater than 900 time slots. Consequently, for certain types of quality of service (QoS) requirements of the form (29), LCFS is better than FCFS.

Most schemes provide QoS guarantees of the following form:

$$Pr\{\text{delay} > D\} \le \delta. \qquad (29)$$

For $\delta = 0$, this is the same as the absolute delay guarantees that our scheme provides. For $\delta > 0$, this QoS guarantee is similar but not identical to having a weighted combination of average and absolute delay in our optimization (26).

Scheduling has been used to refer to the order in which multiple users/flows are serviced at any node [14]. Such disciplines solve the problem of ordering the different flows, based on certain metrics, before transmission. On the other hand, our schedulers solve the problem of finding the number of packets in a given flow to be transmitted during any time slot. In an actual wireless network, we should potentially combine the known service disciplines like WFQ with rate- and power-adaptive schedulers we introduced.

C. Source Coding Interpretation

If we consider our *information* to lie in the packet arrival times, then our scheduler action is analogous to that of a lossy source coder. The higher the delay allowed, the more is the compression of information. For example, if delay limit is one then the arrival timing of the packets are exactly known at the receiver; equivalent to lossless coding. As the delay limit increases, the scheduler delays certain packets more than others and the receiver will only have a *noisy* version of the packet arrival times; this is similar to lossy coding. Asymptotically, as the delay goes to infinity, the receiver will only have information on the average arrival rate of the packets and no information about their arrival times; this case corresponds to zero-rate coding. This loss of information with increasing delays can be attributed to the fact that the scheduler mapping $\alpha : x \mapsto u$ is not invertible except when delay equals 1.

VII. CONCLUSION

In this paper, we presented power optimal schedulers for transmission of bursty traffic on AWGN and fading channels under delay constraints. A computationally simple near-optimal scheduler, labeled log-linear scheduler, was also proposed. An approximate closed-form relation between average power and average delay was derived; it was shown that at large values of delay, the logarithm of the power varies inversely with the

---

[10]Note that the header-based alternative is similar to using a preamble to train the receiver, and the ML detector (27) is equivalent to blind channel estimation.

delay. Power-efficient schedulers which use packet arrival time stamps to guarantee bounds on absolute delay were also proposed. The power savings were demonstrated in a practical system using a simple multirate transmission scheme, which employs a convolutional codes and multirate QAM.

The power savings can be attributed to the strictly convex relation between power and rate. Fundamentally, if the power versus rate relationship is strictly convex, scheduling of packets can be used to trade delay with power, if either the source or channel is time varying.

Our main objective in the paper was to demonstrate the utility of packet scheduling in wireless systems. For the same reason, we had assumed simplistic source and channel models. For instance, i.i.d. source models rarely represent real sources, which typically do have strong time correlation [36]. Thus, a challenge in implementing the proposed schedulers in a practical system is faced when there are errors in the model or if the model parameters are not known accurately. Other imperfections include the delay and error in transmitter knowledge of the channel, which can possibly be accounted for by using results from [37] and [38].

APPENDIX A
PROOF OF PROPOSITION 4: CONDITIONS FOR ZERO-OUTAGE SCHEDULERS

If $x_n \leq L - M$ then from (2), $x_{n+1} \leq L$ and, as a result, there will be no buffer overflow. So a buffer overflow is possible only for $x_n > L - M$, thus we will concentrate on the case when $x_n = i, i = L - M + 1, \ldots, L$.

Direct (If): We will first prove that if either Condition 1 or 2 is satisfied, then the scheduler will be zero outage. If $x_n \geq L - M$, and Condition 2 is satisfied, then using (2) we get

$$x_{n+1} \leq i - z + M$$
$$\leq L - M + M = L.$$

Therefore, there will be no buffer overflow. On the other hand, if Condition 1 is satisfied for state $x_n \geq L - M$, then there is zero probability for the buffer to be in that state. Thus, no buffer overflow is possible. Note that we have assumed that the buffer starts from the zero state $x_0 = 0$ (if the buffer was initialized from a state $x_n = i$ such that $s_i = 0$ and $(i - z) > L - M + 1$ then there will be a buffer overflow with finite probability).

Converse (Only If): We will prove the converse by contradiction, that if there exists state $x_n = i \geq L - M$ that does not satisfy both Conditions 1 and 2, then the corresponding scheduler will not be zero outage. If state i of the buffer does not satisfy Condition 1, with finite nonzero probability the buffer will be in state i. Since state i also does not satisfy Condition 2, using (2) with finite probability

$$x_{n+1} = i - z + M$$
$$> L - M + M = L.$$

Consequently, a buffer overflow occurs. □

APPENDIX B
PROOF OF PROPOSITION 5: EQUIVALENT ELEMENTS OF $\boldsymbol{Q}_{D_0,P_0}$

If state i for which $s_i = 0$ is such that $i \leq (L - M)$, then the proposition is trivially proved. So, assume $s_i = 0$ for some $i > (L - M)$. Since α is deterministic, all its entries are zero or one. Using (7) implies that each column of α and $\hat{\alpha}$ contains exactly one nonzero element. Without loss of generality, let $\alpha_{j,i} = 1$ for some i, j. Now, if $j = i$, then $\hat{\alpha} = \alpha$ and $z = i \Rightarrow i - z \leq L - M$. If $j \neq i$, choose $\hat{\alpha}_{j,k} = \alpha_{j,k}, k = 0, 1, \ldots, L; k \neq i, j = 0, 1, \ldots, L$, and $\hat{\alpha}_{i,i} = 1$. Clearly, under scheduler $\hat{\alpha}$ for state i, $(i - z) \leq (L - M)$. Since $\hat{\alpha}_{j,k} = \alpha_{j,k}, k = 0, 1, \ldots, L; k \neq i, j = 0, 1, \ldots, L$ and $s_i = 0$, α, and $\hat{\alpha}$ give the same average delay and power (which follows from (9) and (10)). That is, $\alpha, \hat{\alpha} \in \boldsymbol{Q}_{D_0,P_0}$. □

APPENDIX C
PROOF OF PROPOSITION 6: BASIS EXPANSION OF ELEMENTS IN Θ

Since $\phi \in \Theta$, it must satisfy either Condition 1 or Condition 2 of Proposition 4. If state i of ϕ satisfies Condition 1, $s_i = 0$, of Proposition 4, then from (8), we note that

$$\boldsymbol{c}_{i,\phi}^T \boldsymbol{s} = 0 \tag{30}$$

where $\boldsymbol{c}_{i,\phi}$ is the ith row of $\boldsymbol{C}_\phi$; note that for clarity, we explicitly show the dependence of the matrix $\boldsymbol{C}$ on ϕ. From the definition of $\boldsymbol{C}_\phi$ it follows that if $\phi = \sum_{j=1}^{F} \eta_j \gamma_j$ then $\boldsymbol{C}_\phi = \sum_{j=1}^{F} \eta_j \boldsymbol{C}_{\gamma_j}$. Hence,

$$\boldsymbol{c}_{i,\phi} = \sum_{j=1}^{F} \eta_j \boldsymbol{c}_{i,\gamma_j}. \tag{31}$$

Now, from (30) and (31) we get

$$\left(\sum_{j=1}^{F} \eta_j \boldsymbol{c}_{i,\gamma_j}^T \right) \boldsymbol{s} = 0$$

which implies that for all $j = 1, 2, \ldots, F$, either $\eta_j = 0$ or $\boldsymbol{c}_{i,\gamma_j}^T \boldsymbol{s} = 0$. If $\boldsymbol{c}_{i,\gamma_j}^T \boldsymbol{s} = 0$ then γ_j is a zero-outage scheduler, i.e., $\gamma_j \in \Theta$. So if γ_j is not a zero-outage scheduler, we must have $\eta_j = 0$. In other words, if a certain buffer state has zero probability under a randomized scheduler, then all deterministic policies which cause that buffer state to have nonzero probability, have zero weight in the basis expansion of that randomized scheduler.

On the other hand, if state i of ϕ satisfies Condition 2 of Proposition 4, then $z = \min\{j : \phi_{j,k} \neq 0\}$, satisfies $k - z \leq L - M$ for all $k = L - M + 1, \ldots, L$. Thus, in the expansion in terms of its basis, $z = \min z_i$ such that $\eta_i \neq 0$ where

$$z_i = \min\left\{j : (\gamma_i)_{j,k} \neq 0\right\}.$$

Consequently, $\eta_i = 0$ if z_i does not satisfy $k - z_i \leq L - M$ for all $k = L - M, \ldots, L$. In other words, if the number of packets transmitted in each buffer state under a given randomized scheduler is large enough to maintain zero buffer overflow, then all deterministic policies which do not transmit enough packets to

maintain zero buffer overflow have zero weight in the basis expansion of the randomized scheduler in terms of deterministic policies.

Hence, $\eta_i = 0$ if γ_i does not satisfy Conditions 1 or 2 of Proposition 4, i.e., if γ_i is not zero outage. It follows that $\eta_i = 0$ if $\gamma_i \in \mathbb{W} - \Theta$. $\qquad\square$

<div align="center">APPENDIX D</div>
<div align="center">PROOF OF LEMMA 7: DELAY-POWER OF TWO "ALMOST"
IDENTICAL SCHEDULERS</div>

Let $\boldsymbol{C}_{\alpha_A}$ and $\boldsymbol{s}_{\alpha_A}$ be the transition probability matrix and stationary probability vector under scheduler α_A. Define $\boldsymbol{C}_{\alpha_B}$ and $\boldsymbol{s}_{\alpha_B}$ in a similar manner. Also define

$$\vec{L} = [0 \; 1 \; \ldots \; L]^T \quad \text{and} \quad \vec{P} = [P_n(0) \; P_n(1) \; \ldots \; P_n(L)]^T.$$

Using (9), $D_{\mathrm{avg}}(\alpha_A) = \boldsymbol{s}_{\alpha_A}^T \vec{L}$ and $D_{\mathrm{avg}}(\alpha_B) = \boldsymbol{s}_{\alpha_B}^T \vec{L}$. From the definition of $\boldsymbol{C}$'s it follows that $\boldsymbol{C}_{\alpha_A}$ and $\boldsymbol{C}_{\alpha_B}$ differ only in the ith column. Further, $\boldsymbol{C}_\alpha = \eta \boldsymbol{C}_{\alpha_A} + (1 - \eta)\boldsymbol{C}_{\alpha_B}$.

Let us assume $\boldsymbol{s}_\alpha = \eta' \boldsymbol{s}_{\alpha_A} + (1 - \eta')\boldsymbol{s}_{\alpha_B}$ and verify there exists an η' to satisfy (8), i.e.,

$$[\eta \boldsymbol{C}_{\alpha_A} + (1 - \eta)\boldsymbol{C}_{\alpha_B}][\eta' \boldsymbol{s}_{\alpha_A} + (1 - \eta')\boldsymbol{s}_{\alpha_B}]$$
$$= \eta' \boldsymbol{s}_{\alpha_A} + (1 - \eta')\boldsymbol{s}_{\alpha_B}$$

Using simple algebraic manipulation of the matrices, we can reduce the condition for finding η' to the following:

$$[\boldsymbol{C}_{\alpha_A} - \boldsymbol{C}_{\alpha_B}][\eta(1 - \eta')\boldsymbol{s}_{\alpha_B} - \eta'(1 - \eta)\boldsymbol{s}_{\alpha_A}] = 0.$$

Since $\boldsymbol{C}_{\alpha_A}$ and $\boldsymbol{C}_{\alpha_A}$ differ only in the ith column, $\boldsymbol{C}_{\alpha_A} - \boldsymbol{C}_{\alpha_B}$ is a matrix with 0 elements in all columns other than the ith column. Therefore, we need to choose η' to satisfy

$$\eta(1 - \eta')s_{\alpha_A, i} - \eta'(1 - \eta)s_{\alpha_B, i} = 0. \qquad (32)$$

Hence,

$$\eta' = \frac{\eta s_{\alpha_B, i}}{\eta s_{\alpha_B, i} + (1 - \eta)s_{\alpha_A, i}}.$$

Now, we only need to show that

$$P_{\mathrm{avg}}(\alpha) = \eta' P_{\mathrm{avg}}(\alpha_A) + (1 - \eta')P_{\mathrm{avg}}(\alpha_B).$$

Using (10), $P_{\mathrm{avg}}(\alpha_A) = \boldsymbol{s}_{\alpha_A}^T \alpha_A^T \vec{P}$

$$\begin{aligned}
P_{\mathrm{avg}}(\alpha) &= \boldsymbol{s}_\alpha^T \alpha^T \vec{P} \\
&= (\eta' \boldsymbol{s}_{\alpha_A}^T + (1 - \eta')\boldsymbol{s}_{\alpha_B}^T)(\eta \alpha_A^T + (1 - \eta)\alpha_B^T)\vec{P} \\
&= \eta\eta' P_{\mathrm{avg}}(\alpha_A) + (1 - \eta)(1 - \eta')P_{\mathrm{avg}}(\alpha_B) \\
&\quad + \eta(1 - \eta')\boldsymbol{s}_{\alpha_B}^T \alpha_A^T \vec{P} + \eta'(1 - \eta)\boldsymbol{s}_{\alpha_A}^T \alpha_B^T \vec{P} \\
&= \eta\eta' P_{\mathrm{avg}}(\alpha_A) + (1 - \eta)(1 - \eta')P_{\mathrm{avg}}(\alpha_B) \\
&\quad + \eta(1 - \eta')\boldsymbol{s}_{\alpha_B}^T(\alpha_A^T - \alpha_B^T)\vec{P} \\
&\quad + \eta(1 - \eta')\boldsymbol{s}_{\alpha_B}^T \alpha_B^T \vec{P} + \eta'(1 - \eta)\boldsymbol{s}_{\alpha_A}^T \alpha_A^T \vec{P} \\
&\quad + \eta'(1 - \eta)\boldsymbol{s}_{\alpha_A}^T(\alpha_B^T - \alpha_A^T)\vec{P} \\
&= \eta' P_{\mathrm{avg}}(\alpha_A) + (1 - \eta')P_{\mathrm{avg}}(\alpha_B) \\
&\quad + [\eta(1 - \eta')\boldsymbol{s}_{\alpha_B}^T - \eta'(1 - \eta)\boldsymbol{s}_{\alpha_A}^T](\alpha_A^T - \alpha_B^T)\vec{P}.
\end{aligned}$$

Now, $(\alpha_A^T - \alpha_B^T)$ equals a matrix with 0 in all rows except the ith row and hence $(\alpha_A^T - \alpha_B^T)\vec{P}$ is a vector with 0 in all elements except the ith row. Now using (32), we find that

$$P_{\mathrm{avg}}(\alpha) = \eta' P_{\mathrm{avg}}(\alpha_A) + (1 - \eta')P_{\mathrm{avg}}(\alpha_B). \qquad\square$$

<div align="center">APPENDIX E</div>
<div align="center">PROOF OF THEOREM 8: CHARACTERIZATION
OF DELAY AND POWER</div>

Using Proposition 6, $\hat{\eta}_i = 0$ if $\gamma_i \in \mathbb{W} - \Theta$. Hence,

$$\phi = \sum_{i=1}^{F'} \hat{\eta}_i \gamma_i$$

where $\gamma_i \in \mathbb{W} \cap \Theta$, $0 \leq \hat{\eta}_i \leq 1$, $\sum_{i=1}^{F'} \hat{\eta}_i = 1$ with $F' = |\mathbb{W} \cap \Theta|$.

In our construction of the basis expansion of scheduler ϕ in the first step, we have a convex combination of two schedulers which differ only in one column. Hence, using Lemma 7, the delay of scheduler ϕ is given by a convex combination of the delays and powers of two schedulers (which need not be deterministic). Each of these matrices is further written in terms of matrices which differ in only one column and hence its power and delay are convex combinations of delay and power of some matrices. Repeating this process till the submatrices are deterministic, the theorem is proved. $\qquad\square$

<div align="center">APPENDIX F</div>
<div align="center">PROOF OF LEMMA 9: CONVERSE TO THEOREM 8</div>

Since scheduler α_A achieves average delay $D_{\mathrm{avg}}(\alpha_A)$ and average power $P_{\mathrm{avg}}(\alpha_A)$, it implies that for any $\epsilon > 0$, there exists an $N > N_{\alpha_A}$ such that using scheduler α_A for N time slots gives average delay $\hat{D}_{\alpha_A}$ and average power $\hat{P}_{\alpha_A}$ satisfying $|\hat{D}_{\mathrm{avg}}(\alpha_A) - D_{\mathrm{avg}}(\alpha_A)| < \epsilon$ and $|\hat{P}_{\mathrm{avg}}(\alpha_A) - P_{\mathrm{avg}}(\alpha_A)| < \epsilon$. Similarly, N_{α_B} exists for α_B.

Now, we time-share between schedulers α_A and α_B in the ratio $\eta : (1 - \eta)$ to get scheduler α. Given any $\epsilon > 0$ find

$$N = \max\left(\frac{N_{\alpha_A}}{\eta}, \frac{N_{\alpha_B}}{1 - \eta}\right).$$

While switching from scheduler α_A to α_B, we additionally flush the buffer in time slot ηN. Now

$$\begin{aligned}
|\hat{D}_{\mathrm{avg}}(\alpha) - D_{\mathrm{avg}}(\alpha)| &\leq \eta|\hat{D}_{\mathrm{avg}}(\alpha_A) - D_{\mathrm{avg}}(\alpha_A)| \\
&\quad + (1 - \eta)|\hat{D}_{\mathrm{avg}}(\alpha_B) - D_{\mathrm{avg}}(\alpha_B)| + \frac{L}{N\lambda} \\
&\leq \eta\epsilon + (1 - \eta)\epsilon + \frac{L}{N\lambda} \\
&= \epsilon + \frac{L}{N\lambda}.
\end{aligned}$$

In the limit as $N \to \infty$, $\frac{L}{N\lambda} \to 0$. Hence,

$$\hat{D}_{\mathrm{avg}}(\alpha) \to \eta D_{\mathrm{avg}}(\alpha_A) + (1 - \eta)D_{\mathrm{avg}}(\alpha_B).$$

Similarly

$$
\begin{aligned}
|\hat{P}_{\text{avg}}(\alpha) - P_{\text{avg}}(\alpha)| \leq{} & \eta|\hat{P}_{\text{avg}}(\alpha_A) - P_{\text{avg}}(\alpha_A)| \\
& + (1-\eta)|\hat{P}_{\text{avg}}(\alpha_B) - P_{\text{avg}}(\alpha_B)| + \frac{1}{N} P_n(L, A_n, \sigma^2) \\
& \leq \eta\epsilon + (1-\eta)\epsilon + \frac{1}{N} P_n(L, A_n, \sigma^2) \\
& = \epsilon + \frac{1}{N} P_n(L, A_n, \sigma^2).
\end{aligned}
$$

In the limit as $N \to \infty$, $\frac{1}{N} P_n(L, A_n, \sigma^2) \to 0$. Again, if we consider K consecutive blocks of N time slots, the total average delay and power is given by the average of the delays and powers in each block of N time slots. Hence, by time-sharing between schedulers α_A and α_B we can achieve all delays and powers given by the convex combination of the delays and powers of α_A and α_B. $\qquad\square$

APPENDIX G
VALUE ITERATION ALGORITHM (VIA)

Let $C(i,a)$ represent the cost incurred in doing action a in state i. Let $P_{ij}(a)$ be the probability of transitioning from state i to state j under action a. Also, let x denote a predetermined state and ϵ be a small positive number that determines the stopping criterion. The general steps in the VIA [28] are as follows.

1) Initialize $v_0 \equiv 0$, $\delta = 1$ and $k = 0$.
2) Evaluate $w_k(i) = \min_a \left\{ C(i,a) + \sum_j P_{ij}(a) v_k(j) \right\}$.
3) Set $\delta = |w_k(x) - w_{k-1}(x)|$ and $v_{k+1}(i) = w_k(i) - w_k(x)$.
4) Repeat steps 2) and 3) above till $\delta < \epsilon$.
5) The optimal actions for each state i are obtained as

$$
a^*(i) = \arg\min_a \left\{ C(i,a) + \sum_j P_{ij}(a) v_k(j) \right\}.
$$

In the above, k represents the kth iteration of the VIA. In the case of schedulers which bound average delay, the different VIA symbols have the following correspondence: $i \equiv x_n, a \equiv u_n$. Also, $P_{i,j}(a)$ depends on the arrival distribution $p(a_n)$. For Gaussian channels, $C(i,a) = P_n(u_n, 1, \sigma^2) + \varepsilon x_n$ where ε is the Lagrangian used in (14).

APPENDIX H
PROOF OF PROPOSITION 11: EXTREMAL POWER SOURCE

The proof follows from the convexity of the power control function $P_n(u_n)$; for simplicity, the dependence on $A_n = 1$ and σ^2 is suppressed. Let the output distribution be $\boldsymbol{n} = [n_0, n_1, \dots n_L]^T$. Since the system is stable, $\mathbb{E}_{\boldsymbol{m}}[a_n] = \mathbb{E}_{\boldsymbol{n}}[u_n]$. Since $P_n(u_n)$ is a convex function, it follows from Jensen's inequality [10] that

$$
P_n(\mathbb{E}_{\boldsymbol{n}}[u_n]) \leq \mathbb{E}_{\boldsymbol{n}}[P_n(u_n)] \tag{33}
$$

where the expectation is over the output distribution $\boldsymbol{n}$. The lower bound in (33) is equal to $P_n(\lambda)$ and is independent of
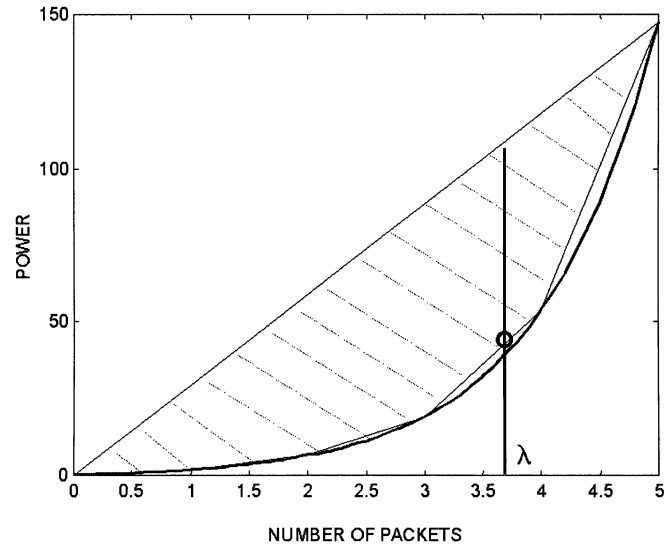

Fig. 8. Power regions achievable for a given output distribution. Hypothetical case of fluid packet models is also shown. This figure is used in the proof of Proposition 11.

output distribution $\boldsymbol{n}$. The output distribution that achieves the lower bound is

$$
n_i = \begin{cases} 1, & i = \lambda \\ 0, & i \neq \lambda. \end{cases}
$$

However, the equality in (33) is achieved only if $\lambda \in \mathbb{Z}^+$. Given the constraint on integer packet sizes (not a fluid source), for any given output distribution $\boldsymbol{n}$ and λ, the average achievable power lies inside or on the boundary of the convex polygon with vertices $(i, P_n(i)), i = 0, 1, \dots, L$, shown graphically as the shaded region in Fig. 8. Now, given a constraint on the mean of $\boldsymbol{n}$, the achievable powers are given by the points of the polygon with abscissa λ. Consequently, the minimum achievable power (shown by $\circ$ in Fig. 8) is given by a convex combination of powers achieved by transmitting $\lfloor\lambda\rfloor$ or $\lfloor\lambda\rfloor + 1$ packets. Since the time-sharing between transmitting $\lfloor\lambda\rfloor$ and $\lfloor\lambda\rfloor + 1$ packets has an average values of λ, it follows that the optimal output distribution is given by

$$
n_i = \begin{cases} 1 - \lambda + \lfloor\lambda\rfloor, & i = \lfloor\lambda\rfloor \\ \lambda - \lfloor\lambda\rfloor, & i = \lfloor\lambda\rfloor + 1 \\ 0, & \text{else.} \end{cases} \tag{34}
$$

It is not clear if there exists a scheduler that shapes the given input distribution to give this distribution within a delay of D. However, if the input has distribution (34), then a simple scheduler that flushes the queue during every time slot achieves the lowest possible power for all delay bounds. Hence, $\boldsymbol{m} = \boldsymbol{n}$ is the input distribution that results in least transmit power in an AWGN channel.

The proof for the distribution with the highest transmit power for $D_o = 1$ is similar, and is omitted. $\qquad\square$

APPENDIX I
PROOF OF THEOREM 13: CONVEXITY OF RANDOMIZED SCHEDULERS WITH ABSOLUTE DELAY BOUND

Time sharing between schedulers v_i in the ratios η_i gives the total average power $P_{\text{avg}}(\zeta) = \sum_{i=1}^{G} \eta_i P_{\text{avg}}(v_i)$. However, we

need to verify that by time sharing of schedulers υ_i, we still have a scheduler which guarantees absolute delay D_{abs}. The situation is similar to our treatment of average-delay bounded schedulers having zero outage. In this case, while switching from scheduler υ_i to υ_j, we should not be in some state B_n which has zero probability of occurring while operating under policy υ_j alone. If we are in such a state B_n, we can find a policy $\hat{\upsilon}_j \in \hat{Q}_{D_{\mathrm{abs}}}$ such that the number of packets transmitted in state B_n is sufficient to meet the absolute delay bound (proof similar to Lemma 5). Further, both υ_j and $\hat{\upsilon}_j$ will have the same average transmit power. $\qquad\square$

REFERENCES

[1] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1986–1992, Nov. 1997.

[2] D. J. Goodman, J. Borras, N. B. Mandayam, and R. Yates, "INFOSTA-TIONS: A new system model for data and messaging services," in *Proc. Vehicular Technology Conf.*, May 1997, pp. 969–973.

[3] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, pp. 379–423, July 1948.

[4] B. Prabhakar, E. U. Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proc. IN-FOCOM*, Anchorage, AK, Apr. 2001.

[5] B. E. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," in *Proc. Allerton Int. Conf. Communication, Control and Computing*, Monticello, IL, 1999, pp. 709–717.

[6] H. Wang and N. B. Mandayam, "Dynamic power control under energy and delay constraints," *IEEE Trans. Inform. Theory*, submitted for publication.

[7] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1135–1149, May 2002.

[8] R. Negi, "Power adaptation strategies for delay constrained channels," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2000.

[9] G. Caire, G. Taricco, and E. Biglieri, "Optimum power control over fading channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1468–1489, July 1999.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[11] R. G. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 124–142, Mar. 1985.

[12] A. Ephremides and B. Hajek, "Information theory and communication networks: An unconsummated union," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2416–2434, Oct. 1998.

[13] İ. E. Telatar and R. G. Gallager, "Combining queueing theory with information theory for multiaccess," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 963–969, Aug. 1995.

[14] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1396, Oct. 1995.

[15] V. Bharghavan, S. Lu, and T. Nandagopal, "Fair scheduling in wireless packet networks: Issues and approaches," *IEEE Personal Commun. Mag.*, vol. 6, pp. 44–55, Feb. 1999.

[16] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless utilization," in *Proc. INFOCOM*, Anchorage, AK, Apr. 2001.

[17] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "CDMA Data QoS Scheduling on the forward link with variable channel conditions," Bell Labs., Lucent Technologies, Tech. Rep., 2000.

[18] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram, "Downlink scheduling in CDMA data networks," in *Proc. MOBICOM*, 2000, pp. 179–190.

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, pp. 150–154, Feb. 2001.

[20] D. Rajan, A. Sabharwal, and B. Aazhang, "Delay differentiated power efficient downlink scheduling," *IEEE Trans. Commun.*, submitted for publication.

[21] ——, "Transmission policies for bursty sources through time varying channels," in *Proc. 35th Annu. Conf. Information Sciences and Systems*, Baltimore, MD, Mar. 2001.

[22] A. Sabharwal, D. Rajan, and B. Aazhang, "Dual problems in power control," in *Proc. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 2001.

[23] D. Rajan, "Power efficient transmission policies for multimedia traffic over wireless channels," Ph.D. dissertation, Rice Univ., Houston, TX, 2002.

[24] D. P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[25] L. H. Ozarow, S. Shamai (Shitz), and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 359–378, May 1994.

[26] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[27] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[28] L. I. Sennott, *Stochastic Dynamic Programming and the Control of Queueing Systems*. New York: Wiley, 1999.

[29] E. Biglieri, J. G. Proakis, and S. Shamai (Shitz), "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2619–2692, Oct. 1998.

[30] B. Prabhakar and R. G. Gallager, "Entropy and the timing capacity of discrete queues," *IEEE Trans. Inform. Theory*, vol. 49, pp. 357–370, Feb. 2003.

[31] E. Knightly, "Enforceable quality of service guarantees for bursty traffic streams," in *Proc. INFOCOM*, San Francisco, CA, Mar. 1998.

[32] S. Nanda, K. Balachandran, and S. Kumar, "Adaptation techniques in wireless packet data services," *IEEE Commun. Mag.*, vol. 38, pp. 54–64, Jan. 2000.

[33] R. G. Gallager, "Basic limits on protocol information in data communication networks," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 385–399, July 1976.

[34] H. L. VanTrees, *Detection, Estimation and Modulation Theory Part I*. New York: Wiley, 1968.

[35] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Trans. Inform. Theory*, vol. 42, pp. 4–18, Jan. 1996.

[36] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *Proc. ACM SIGCOMM*, San Francisco, CA, Sept. 1993.

[37] H. Viswanathan, "Capacity of Markov channels with receiver CSI and delayed feedback," *IEEE Trans. Inform. Theory*, vol. 45, pp. 761–771, Mar. 1999.

[38] G. Caire and S. Shamai (Shitz), "On the capacity of some channels with channel state information," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2007–2019, Sept. 1999.