Intelligent Packet Dropping for Optimal Energy-Delay Tradeoffs in Wireless Networks

Michael J. Neely
University of Southern California
http://www-rcf.usc.edu/∼mjneely

***Abstract*— We explore the advantages of intelligently dropping a small fraction of packets that arrive for transmission over a time varying wireless downlink. Without packet dropping, the optimal energy-delay tradeoff conforms to a square root tradeoff law, as shown by Berry and Gallager (2002). We show that intelligently dropping any non-zero fraction of the input rate dramatically changes this relation from a square root tradeoff law to a logarithmic tradeoff law. Further, we demonstrate an innovative algorithm for achieving this logarithmic tradeoff without requiring a-priori knowledge of arrival rates or channel probabilities. The algorithm can be implemented in real time and easily extends to yield similar performance for multi-user systems.**

I. Introduction

Wireless systems must offer high throughput and low delay while operating with very little power. In order to maximize performance, it is desirable for systems to react to current channel conditions using rate adaptive and power adaptive transmission technology. In this paper, we develop a scheduling algorithm that uses channel information to yield an average power expenditure that can be pushed arbitrarily close to the minimum average power required for system stability, with a corresponding optimal tradeoff in average delay.

In [5] it was shown that when all packets must be transmitted, the optimal energy-delay tradeoff is given by a square root tradeoff law, known as the Berry-Gallager bound. In this paper, we consider optimal energy-delay tradeoffs under the assumption that a small fraction of packets can be dropped. We show that intelligently dropping packets can dramatically change the energy-delay relation from a square root tradeoff law to a logarithmic tradeoff law. This result holds for any non-zero bound on the packet drop rate. Further, we demonstrate an innovative algorithm for joint power allocation and packet dropping that achieves the optimal logarithmic tradeoff without requiring a-priori knowledge of the input rate or the channel state probabilities. The algorithm can be implemented in real time and easily extends to offer provably optimal energy-delay tradeoffs for multi-user systems. This demonstrates that orders of magnitude improvements in average delay are possible if a non-zero packet drop rate can be tolerated.

Related work in [6] [7] [8] [9] considers energy and delay issues in a single wireless downlink with a static channel, and work in [5] [10] [11] considers downlinks with fading channels. The fundamental square root tradeoff for single-user systems is developed by Berry and Gallager in [5], and this tradeoff is extended to multi-user systems in [12]. The problem of fairness and utility optimal flow control is investigated in [13], where it is shown that the fundamental utility-delay tradeoff law is quite different and has a logarithmic structure. The dynamic control algorithms of [12] [13] combine the concepts of *buffer partitioning* developed in [5] and *performance optimal Lyapunov scheduling* developed in [2] [3] [4]. Specifically, the work in [2] [3] [4] develops a simple Lyapunov technique for achieving stability and performance optimization simultaneously (extending the Lyapunov results developed for queueing stability in works such as [14]-[21]).

This paper uses similar techniques to address the problem of intelligent packet dropping for energy efficiency. However, the optimal control strategies in this context have a different structure from those of [5] [12] [13]. Specifically, the algorithms of [5] [12] [13] partition the buffer of an infinite queue into two halves, where different drift modes are designed for each partition. Here, we design a strategy that emulates a *finite buffer queue* with strictly positive drift. We show that the strategy yields a logarithmic delay tradeoff that cannot be achieved in systems that do not allow packet dropping.

An outline of this paper is as follows: In the next section we present the system model and problem formulation. In Section III the basic positive drift algorithm is developed, under the assumption that all channel state probabilities are a-priori known. A more practical dynamic strategy that does not require such a-priori knowledge is developed in Section IV. The strategy uses a novel form of Lyapunov theory to make on-line decisions that are tradeoff optimal. Necessity of the logarithmic tradeoff is also discussed.

II. System Model

Consider a single wireless transmitter that operates in slotted time with slots normalized to integer units $t \in \{0, 1, 2, \ldots\}$. The transmission rate offered by the transmitter on slot t depends on a controllable power variable $P(t)$ and an uncontrollable channel state $S(t)$ according to a general *rate-power function* $C(P(t), S(t))$, taking units of bits/slot. We assume that power allocations are limited by a peak power constraint P_{max}, so that $0 \leq P(t) \leq P_{max}$ for all t. Channel states $S(t)$ are assumed to be contained within some finite but arbitrarily large state space $\mathcal{S}$. An example rate-power function is the logarithmic capacity curve for an Additive White Gaussian

This material is based on work supported in part by the National Science Foundation grant OCE 0520324. A more detailed version of this work will be presented in WiOpt 2006 [1].

Noise channel:

$$C(P, S) = \log(1 + P\alpha_S) \tag{1}$$

where α_S is an attenuation/noise coefficient associated with channel state S. We assume that rate-power functions are bounded by some finite maximum transmission rate μ_{max}.

Channel states $S(t)$ are assumed to be independent and identically distributed (i.i.d.) every slot, with state probabilities $\pi_S = Pr[S(t) = S]$ for all $S \in \mathcal{S}$. Let $A(t)$ represent the amount of new data that enters the system at time t (in units of bits). This arrival process $A(t)$ is assumed to be i.i.d. with rate λ, so that $\mathbb{E}\{A(t)\} = \lambda$ for all t. Further, we assume the second moment of arrivals is bounded by a constant $\hat{A}_{max}$, so that:

$$\mathbb{E}\left\{A(t)^2\right\} \leq \hat{A}_{max}^2 \text{ for all } t$$

Newly arriving data is either admitted to the system, or dropped. Let $\tilde{A}(t)$ be a control variable representing the amount of new arrivals admitted on slot t, where $0 \leq \tilde{A}(t) \leq A(t)$. All admitted data is stored in a queue to await transmission, and we let $U(t)$ represent the queue backlog or *unfinished work* in the system at time t. Every timeslot, a downlink controller observes the current channel state $S(t)$ and the current queue backlog $U(t)$ and chooses a power allocation $P(t)$ subject to the constraint $0 \leq P(t) \leq P_{max}$. This yields an offered transmission rate of $\mu(t) = C(P(t), S(t))$. The queueing dynamics thus proceed as follows:

$$U(t+1) = \max[U(t) - \mu(t), 0] + \tilde{A}(t) \tag{2}$$

Note that the actual bits transmitted can be different from $\mu(t)$ if there are not enough bits in the queue to transmit at the full offered transmission rate. Let $\tilde{\mu}(t)$ represent the actual amount of bits transmitted during slot t. Note that $\tilde{\mu}(t) \leq \mu(t)$, and strict inequality can only occur if $U(t) < \mu_{max}$. Let $\rho < 1$ represent a required *acceptance ratio*. The goal is to achieve an optimal energy-delay tradeoff while maintaining an acceptance rate greater than or equal to $\rho\lambda$. That is, we require the following guarantee on long term throughput:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\tilde{\mu}(\tau)\right\} \geq \rho\lambda$$

A. The Berry-Gallager Bound

Let μ_c represent the *downlink capacity*, so that the system can stably support any arrival rate λ such that $0 \leq \lambda < \mu_c$. Define $\Phi(\lambda)$ as the minimum energy required to stabilize the queue if the input rate is λ (assuming that $0 \leq \lambda < \mu_c$). It can be shown that $\Phi(\lambda)$ indeed depends only on λ (and not on higher order arrival statistics), and that it is convex over the interval $0 \leq \lambda < \mu_c$. In [5], it is shown that a sequence of stabilizing power allocation algorithms, indexed by increasing positive numbers V, can be designed that push average power expenditure arbitrarily close to $\Phi(\lambda)$. Further, it was shown that, subject to some concavity assumptions on the $C(P, S)$ function and some admissibility assumptions on the input process, any stabilizing power allocation algorithm that yields average power within $O(1/V)$ of the minimum power $\Phi(\lambda)$ must also have average delay greater than or equal to $\Omega(\sqrt{V})$.[1] We refer to this square root tradeoff law as the *Berry-Gallager bound*. We note that the admissibility assumptions required for this bound include the assumption that arrivals and channel states are i.i.d. over timeslots. Further, the bound is derived under the assumption that no data is allowed to be dropped.

As an example, consider a downlink that satisfies all assumptions required for the Berry-Gallager bound. Assume the arrival process is i.i.d. with rate λ. However, suppose that we only need to admit a fraction $\rho < 1$ of all incoming data, so that a drop rate of up to $(1-\rho)\lambda$ bits/slot can be tolerated. The minimum power required to stabilize such a system is thus equal to $\Phi^* \triangleq \Phi(\rho\lambda)$. Hence, the new goal is to push average power expenditure arbitrarily close to Φ^*. Consider now the naive dropping policy that makes random and independent admission decisions every timeslot, where all incoming data $A(t)$ is admitted with probability ρ, and else it is dropped. The resulting admitted rate is exactly equal to $\rho\lambda$. However, the admitted input stream is still i.i.d. from slot to slot, and hence the Berry-Gallager bound still governs the energy-delay performance associated with scheduling this admitted data. Therefore, this naive approach to packet dropping cannot overcome the square root tradeoff relation.

However, instead of *randomly* dropping packets, we consider schemes that *intelligently* drop packets. Remarkably, we find that for any arbitrarily small but positive dropping ratio (i.e., any $(1-\rho) > 0$), it is possible to design an intelligent packet dropping scheme (together with a power allocation scheme) that yields an average power expenditure that differs from Φ^* by at most $O(1/V)$, while yielding average delay that grows only logarithmically in the control parameter V. Hence, the ability to drop packets dramatically improves the energy-delay tradeoff law. This result shows that the square root curvature of the Berry-Gallager bound is due only to a very small fraction of packets that arrive at in-opportune times. Average delay can be dramatically reduced by identifying these packets and dropping them.

III. A Dropping Scheme for Known Statistics

In this section we demonstrate existence of a scheme that uses intelligent packet dropping to overcome the Berry-Gallager bound. The policy developed in this section is not intended as a practical means of control, as it can only be constructed via off-line computations based on full knowledge of the arrival rate λ and the channel state probabilities π_S (for each $S \in \mathcal{S}$). In Section IV we construct an on-line strategy that achieves the same performance without requiring knowledge of these parameters. We first present the following Lemma from [3]:

Lemma 1: If channel states $S(t)$ are i.i.d. and if the rate-power function $C(P, S)$ satisfies the structural properties of the previous section, then for any $\lambda < \mu_c$ a stationary power allocation policy can be designed that makes randomized power allocation decisions $P^*(t)$ based only on observations

[1] Where the notation $f(V) = \Omega(\sqrt{V})$ denotes a function that increases at least as fast as a square root function.

of the current channel state $S(t)$, yielding:

$$\mathbb{E}\{P^*(t)\} = \Phi(\lambda) \text{ for all } t$$
$$\mathbb{E}\{\mu^*(t)\} = \lambda \text{ for all } t$$

where $\mu^*(t) = C(P^*(t), S(t))$ is the associated transmission rate of the randomized scheme.

A. The Positive Drift Algorithm

The first step of our intelligent packet dropping algorithm is to emulate a *finite buffer queueing system* with buffer size Q, where the constant Q is to be determined later. That is, we modify the queueing update equation as follows:

$$U(t+1) = \min[Q, \max[U(t) - \mu(t), 0] + A(t)] \quad (3)$$

This is the same queue update equation as (2), with the exception that any data exceeding the buffer size Q is necessarily dropped. The following policy is defined in terms of a given required acceptance ratio $\rho < 1$.

Positive Drift Algorithm for Known Statistics:

1) Emulate the finite buffer system (3) using a constant buffer size Q (to be chosen later).
2) Let $P(t) = P^*(t)$, where $P^*(t)$ is the stationary policy that observes $S(t)$ and then randomly allocates power to yield $\mathbb{E}\{P^*(t)\} = \Phi((\rho+\epsilon)\lambda)$, $\mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda$ for all t (as in Lemma 1), for some small value ϵ such that $0 < \epsilon < (1-\rho)$, to be determined later.

For suitable choices of Q and ϵ, the above policy yields a logarithmic energy-delay tradeoff relation. It is perhaps surprising that the policy is designed to have a *positive drift in the direction of the finite buffer theshold* Q. Intuitively, one might expect an optimal queueing control algorithm to have negative drift towards the empty state $U(t) = 0$. However, this is precisely what the algorithm is designed to avoid, as fundamental inefficiencies arise from the edge effects associated with a queue becoming empty. The algorithm is similar in spirit to the *buffer partitioning algorithm* of [5], which uses a positive drift whenever queue backlog is below a given threshold and a negative drift when backlog is larger than this threshold. However, in our algorithm above, the "threshold" is given by the finite buffer size Q. Any data that violates this threshold is simply dropped.

B. Analysis of the Positive Drift Algorithm

To analyze performance of the algorithm, note that time average power expenditure satisfies:

$$\begin{aligned}\overline{P} \triangleq \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{P^*(t)\} &= \Phi(\rho\lambda + \epsilon\lambda) \\ &\leq \Phi(\rho\lambda) + \Phi'(\lambda)\epsilon\lambda \quad (4)\end{aligned}$$

where $\Phi'(\lambda)$ denotes the right derivative of the $\Phi(\cdot)$ function evaluated at λ (note that finite right derivatives exist for any convex function over an open interval). Thus, average power expenditure satisfies $\overline{P} \leq \Phi^* + \Phi'(\lambda)\epsilon\lambda$. For any fixed control parameter $V \geq 1$, the idea is to choose $\epsilon = (1-\rho)/(2V)$. With this choice, it follows from (4) that average power expenditure is within $O(1/V)$ of the minimum average power Φ^*.

Next, note that the time average transmission rate is given by:

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda$$

However, this time average transmission rate can be larger than the time average throughput, due to the fact that the actual data transmitted may be less than $\mu^*(t)$ if $U(t) < \mu_{max}$. To ensure that the throughput is greater than or equal to $\rho\lambda$, we present the following lemma concerning edge effects in any queueing system with a transmission rate $\mu(t)$. Recall that $\tilde{\mu}(t)$ is defined as the actual data transmitted during slot t.

Lemma 2: (Edge Effects) If $\mu(t) \leq \mu_{max}$ for all t, then any stochastic queueing system that transmits at the full rate $\mu(t)$ whenever $U(t) \geq \mu_{max}$ must satisfy:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{\tilde{\mu}(\tau)\} \geq \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{\mu(\tau)\} - \alpha\mu_{max}$$

where:

$$\alpha \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} Pr[U(t) < \mu_{max}] \quad (5)$$

Proof: Omitted for brevity. □

Intuitively, the above lemma indicates that the actual throughput of the queueing system differs from the time average transmission rate by an amount that is at most $\alpha\mu_{max}$, where α represents time average probability that the queue backlog drops below μ_{max}. We call α the "edge probability."

Applying the above lemma to the positive drift algorithm above (where $\mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda$ for all t) yields the following guarantee on time average throughput:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{\tilde{\mu}^*(\tau)\} \geq \rho\lambda + \epsilon\lambda - \alpha\mu_{max} \quad (6)$$

To ensure that the throughput is greater than or equal to $\rho\lambda$, from (6) we find it suffices to ensure that the edge probability α is small enough to satisfy $\alpha\mu_{max} \leq \epsilon\lambda$. However, note that on every timeslot t, the expected difference between the arrival rate and the transmission rate satisfies:

$$\mathbb{E}\{A(t) - \mu^*(t)\} = \lambda - (\rho+\epsilon)\lambda = \lambda(1-\rho-\epsilon) \quad (7)$$

The above expectation is defined as the *drift* of the algorithm. Using the fact that $\epsilon = (1-\rho)/(2V)$, it follows that the drift is greater than or equal to $\lambda(1-\rho)/2$ whenever $V \geq 1$. This positive drift tends to increase queue backlog, pushing $U(t)$ away from the edge region $U(t) < \mu_{max}$. Further, it can be shown that the resulting edge probability α decays exponentially in the buffer size Q. Therefore, the edge probability α can be made as small as desired, satisfying $\alpha\mu_{max} \leq \epsilon\lambda$, *while maintaining a buffer size Q that is logarithmic in $1/\epsilon$, and hence logarithmic in V*. Because $U(t) \leq Q$ for all t, it follows that average queue backlog is $O(\log(V))$, as is the average delay of admitted data (via Little's Theorem). This demonstrates feasibility of a logarithmic energy-delay tradeoff.

While the positive drift algorithm is conceptually very simple, it cannot be implemented without full a-priori knowledge of the arrival rate λ and the channel probabilities π_S (for each $S \in \mathcal{S}$). Even if all of these parameters are estimated, the intrinsic estimation error might preclude realization of the desired performance, and could lead to significant mismatch problems if input rates or channel probabilities change over time. Further, the algorithm does not easily extend to multi-user, multi-channel systems, because the total number of channel states in such systems grows geometrically with the number of channels. Therefore, it is essential to construct a more practical algorithm to achieve a logarithmic energy-delay tradeoff.

IV. An On-Line Algorithm for Unknown Statistics

To construct an on-line algorithm for achieving a logarithmic energy-delay tradeoff, we use the theory of *performance optimal Lyapunov scheduling* [2] [3] [4]. To this end, suppose the system emulates a finite buffer system with buffer size Q (to be chosen later), so that queue backlog $U(t)$ evolves according to (3). Define the following *Lyapunov function* $L(U)$:

$$L(U)\triangleq e^{\omega(Q-U)}$$

where $\omega > 0$ is a parameter to be determined later. Because $U(t) \leq Q$ for all t, the Lyapunov function $L(U(t))$ reaches its minimum value when $U(t) = Q$, and increases exponentially when queue backlog deviates from the buffer threshold Q. We show that scheduling to minimize the drift of this Lyapunov function from one slot to the next ensures that the edge probability α decays exponentially in Q.

To maintain high throughput, it is desirable to ensure that the time average transmission rate $\mu(t)$ is greater than or equal to $(\rho+\epsilon)\lambda$, for some value ϵ such that $0 < \epsilon < (1-\rho)$, to be determined later. To this end, we use the *virtual queue* concept developed in [3]. Let $X(t)$ represent a virtual queue that is implemented purely in software, where $X(0) = 0$ and where $X(t)$ follows the following update equation every slot:

$$X(t+1) = \max[X(t) - \mu(t), 0] + (\rho+\epsilon)A(t) \qquad (8)$$

where $A(t)$ is the amount of new arrivals during slot t (some of which may not be admitted to the actual queue $U(t)$), and where $\mu(t)$ is the transmission rate chosen by the downlink control algorithm. Note that $X(t)$ can be viewed as the backlog in a queue with input $(\rho+\epsilon)A(t)$ and time varying server rate $\mu(t)$ (see Fig. 1). We say that a queue $X(t)$ is *strongly stable* if $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{X(\tau)\right\} < \infty$.

Lemma 3: If the $X(t)$ queue is strongly stable and the $A(t)$ process is i.i.d. with rate λ, then:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\mu(\tau)\right\} \geq \rho\lambda + \epsilon\lambda$$

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\tilde{\mu}(\tau)\right\} \geq \rho\lambda + \epsilon\lambda - \alpha\mu_{max}$$

Proof: The first inequality follows from basic principles of queueing stability, and the second inequality follows from the first together with Lemma 2. □

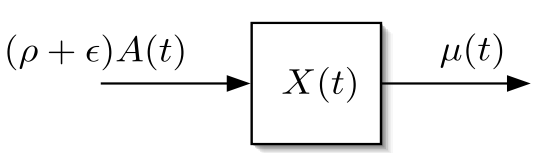

Fig. 1. An illustration of the virtual queueing system associated with the $X(t)$ update equation (8).

A. Performance Optimal Lyapunov Scheduling

Our technique of stochastic queue optimization is based on the theory of *performance optimal Lyapunov scheduling*, which allows stability and performance optimization to be achieved via a single drift argument [2] [4] [3]. This extends the Lyapunov stability results of [14]-[21], and is closely related to *stochastic gradient optimization* (see, for example, [22] for an application to data networks). To demonstrate the technique, consider a system with a vector process $\boldsymbol{Z}(t)$ representing a set of queue states that evolve according to some probability law. Let $P(t)$ represent a non-negative control process that affects system dynamics, and let P^* represent a target upper bound desired for the time average of $P(t)$. Let $\Psi(\boldsymbol{Z})$ represent any non-negative function of $\boldsymbol{Z}$ (representing a Lyapunov function), and let $\Delta(\boldsymbol{Z}(t))$ represent the *conditional Lyapunov drift*, defined as follows:[2]

$$\Delta(\boldsymbol{Z}(t))\triangleq\mathbb{E}\left\{\Psi(\boldsymbol{Z}(t+1)) - \Psi(\boldsymbol{Z}(t)) \mid \boldsymbol{Z}(t)\right\} \qquad (9)$$

We have the following important lemma, which is a modified version of similar results developed in [2] [3] [4].

Lemma 4: (Lyapunov Optimization [2][3][4]) If there are constants $B > 0$, $\epsilon > 0$, and $V \geq 0$, together with a non-negative function $f(\boldsymbol{Z})$, such that the queueing system satisfies the following drift inequality for all t and all $\boldsymbol{Z}(t)$:

$$\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\left\{P(t) \mid \boldsymbol{Z}(t)\right\} \leq B - \epsilon f(\boldsymbol{Z}(t)) + VP^*$$

then:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{P(\tau)\right\} \leq P^* + B/V$$

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{f(\boldsymbol{Z}(\tau))\right\} \leq \frac{B + V(P^* - \overline{P}_{inf})}{\epsilon}$$

where

$$\overline{P}_{inf}\triangleq\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{P(\tau)\right\}$$

If V is a control parameter of the system, the above lemma indicates that the time average of the $P(t)$ process can be bounded by a value that is arbitrarily close to the target value P^*, with a corresponding tradeoff in the time average value of $f(\boldsymbol{Z}(t))$ that is at most linear in V.

To apply Lemma 4 to our queueing problem, let $\boldsymbol{Z}(t) = (U(t), X(t))$ represent the vector queue state of both the actual

[2] Strictly speaking, correct notation for the conditional Lyapunov drift in (9) is $\Delta(\boldsymbol{Z}(t), t)$, as the drift may also depend on the timeslot t. However, we use the simpler notation $\Delta(\boldsymbol{Z}(t))$ as a formal and more concise representation of the same thing.

and virtual queues, and define the following *mixed Lyapunov function*:

$$\Psi(\boldsymbol{Z}) \triangleq L(U) + \frac{1}{2}X^2 = e^{\omega(Q-U)} + \frac{1}{2}X^2$$

The conditional drift $\Delta(\boldsymbol{Z}(t))$ for the above Lyapunov function is defined in (9). Motivated by Lemma 4, the goal of our dynamic control strategy is to choose $P(t)$ to minimize a bound on the following drift metric every timeslot t:

$$\text{Drift Metric:} \quad \Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\left\{P(t) \mid \boldsymbol{Z}(t)\right\} \tag{10}$$

where $V \geq 1$ is a control parameter that effects the energy-delay performance of the algorithm.

B. Algorithm Construction

To compute a bound on the drift metric of the previous subsection, it is useful to define σ^2 to be any positive constant that satisfies the following inequality for all t:

$$\sigma^2 \geq \mathbb{E}\left\{(\mu(t) - A(t))^2 \mid \boldsymbol{Z}(t)\right\} \tag{11}$$

Note that because $\mu(t) \leq \mu_{max}$ and $\mathbb{E}\left\{A(t)^2\right\} \leq \hat{A}_{max}^2$, choosing $\sigma^2 \triangleq \mu_{max}^2 + \hat{A}_{max}^2$ ensures that (11) is satisfied for all t. Further assume that the parameter ω is chosen to satisfy:

$$\omega e^{\omega \mu_{max}} \leq \lambda(1 - \rho - \epsilon)/\sigma^2 \tag{12}$$

Dynamic Packet Dropping Policy: Every timeslot, observe the current channel state $S(t)$ and the current queue backlogs $U(t)$ and $X(t)$. Then:

1) Allocate power $P(t) = P$, where P solves:

   Maximize: $C(P, S(t))(X(t) - \omega e^{\omega(Q-U(t))}) - VP$

   Subject to: $0 \leq P \leq P_{max}$

2) Iterate the virtual queue $X(t)$ according to (8), using $\mu(t) = C(P(t), S(t))$.
3) Emulate the finite buffer queue $U(t)$ according to (3).

Theorem 1: (Dynamic Packet Dropping Performance) For a given value $\rho < 1$ and a fixed control parameter $V \geq 1$, if parameters ϵ, ω, and Q are chosen so that $\epsilon = (1-\rho)/(2V)$, ω is positive and satisfies (12), and $Q = \log(xV)/\omega$, where x is any value that satisfies:

$$x \geq \frac{4\mu_{max}e^{\omega\mu_{max}}B}{\lambda^2\omega(1-\rho-\epsilon)(1-\rho)} \tag{13}$$

then:

(a) $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{P(\tau)\right\} \leq \Phi^* + O(1/V)$
(b) $U(t) \leq Q$ for all t, where $Q = O(\log(V))$
(c) $\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\tilde{\mu}(\tau)\right\} \geq \rho\lambda$

We omit the proof for brevity (see [1]). Note that because $U(t) \leq O(\log(V))$ for all t, average delay is also $O(\log(V))$ (by Little's Theorem). Hence, the algorithm satisfies the required acceptance rate and yields a logarithmic energy-delay tradeoff. Note that the constants can be chosen to satisfy the necessary inequalities (12) and (13) just by knowing a lower bound λ_0 on the input rate λ, so that the exact input rate λ is not required. Likewise, the channel state probabilities π_s are not required for implementation. We note that the value of Q was chosen only to ensure a sufficiently small analytical bound on the edge probability α. Our analysis was conservative, and experimentally we find that (constant factor) improvements in delay can be achieved by appropriately reducing the value of Q, without affecting throughput or average energy expenditure. Example systems can be constructed for which no algorithm can yield a sub-logarithmic energy-delay tradeoff, and hence our $O(\log(V))$ result is fundamental. Our technique can also be applied to multi-user downlinks and multi-hop networks, where a drift metric similar to (10) is minimized [1].

REFERENCES

[1] M. J. Neely. Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks. *Proceedings of WiOpt*, Boston: April 2006.

[2] M. J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, LIDS, 2003.

[3] M. J. Neely. Energy optimal control for time varying wireless networks. *Proceedings of IEEE INFOCOM*, March 2005.

[4] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proceedings of IEEE INFOCOM*, March 2005.

[5] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135-1149, May 2002.

[6] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. Energy-efficient packet transmission over a wireless link. *IEEE/ACM Trans. Networking*, vol. 10, pp. 487-499, Aug. 2002.

[7] M. Zafer and E. Modiano. A calculus approach to minimum energy transmission policies with quality of service guarantees. *IEEE Proceedings of INFOCOM*, March 2005.

[8] M. A. Khojastepour and A. Sabharwal. Delay-constrained scheduling: Power efficiency, filter design, and bounds. *IEEE Proceedings of INFOCOM*, March 2004.

[9] W. Chen and U. Mitra. Delay-constrained energy-efficient packet transmissions. *Proc. of IEEE INFOCOM*, April 2006.

[10] A. Fu, E. Modiano, and J. Tsitsiklis. Optimal energy allocation for delay-constrained data transmission over a time-varying channel. *IEEE Proceedings of INFOCOM*, 2003.

[11] M. Goyal, A. Kumar, and V. Sharma. Power constrained and delay optimal policies for scheduling transmission over a fading channel. *IEEE Proceedings of INFOCOM*, April 2003.

[12] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *Proceedings of IEEE INFOCOM*, April 2006.

[13] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *Proceedings of IEEE INFOCOM*, April 2006.

[14] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, Vol. 37, no. 12, Dec. 1992.

[15] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. on Inform. Theory*, vol. 39, pp. 466-478, March 1993.

[16] P.R. Kumar and S.P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control*, Feb. 1995.

[17] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *Proc. INFOCOM*, 1996.

[18] N. Kahale and P. E. Wright. Dynamic global packet routing in wireless networks. *IEEE Proceedings of INFOCOM*, 1997.

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, 2001.

[20] E. Leonardi, M. Melia, F. Neri, and M. Ajmone Marson. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. *Proc. INFOCOM*, 2001.

[21] M. J. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, January 2005.

[22] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *to appear in IEEE Trans. on Wireless Systems*, 2005.