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

Michael J. Neely, *Senior Member, IEEE*

***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.**

***Index Terms*—Berry-Gallager bound, opportunistic scheduling.**

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 [2] 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 significant improvements in average delay are possible if a non-zero packet drop rate can be tolerated.

Related work in [3]–[6] considers energy and delay issues in a single wireless downlink with a static channel, and work in [2], [7], [8] considers downlinks with fading channels. The fundamental square root tradeoff for single-user systems is developed by Berry and Gallager in [2], and this tradeoff is extended to multi-user systems in [9]. The problem of fairness and utility optimal flow control is investigated in [10], where it is shown that the fundamental utility-delay tradeoff law is quite different and has a logarithmic structure. The dynamic control algorithms of [9], [10] combine the concepts of *buffer partitioning* developed in [2] and *performance optimal Lyapunov networking* from [11]–[14]. Specifically, the work in [11]–[14] presents simple Lyapunov techniques for achieving stability and performance optimization simultaneously (extending the Lyapunov results developed for queueing stability in works such as [15]–[22]).

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 [2], [9]. Specifically, the algorithms of [2], [9] 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 proven in Section VI for the special case of systems with no channel state variation. Extensions to multi-user systems are briefly considered in Section VII. A modified adaptive threshold algorithm is presented in Section VIII to yield a further constant-factor delay improvement while maintaining tradeoff optimality. This adaptive threshold policy extends the conference version of this paper [1] and can be viewed as a new technique for network learning. Simulations are provided in Section IX.

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

Manuscript received February 01, 2008; revised May 14, 2008. Current version published March 11, 2009. This work was supported in part by the DARPA IT-MANET Program Grant W911NF-07-0028, the NSF Grant OCE 0520324, and the NSF Career Grant CCF-0747525. This work was presented in part at the WiOpt Conference, Boston, MA, April 2006. Recommended by Associate Editor C. Abdallah.

The author is with the Electrical Engineering Department, University of Southern California, Los Angeles, CA 90089-2565 USA (e-mail: mjneely@usc.edu).

Digital Object Identifier 10.1109/TAC.2009.2013652

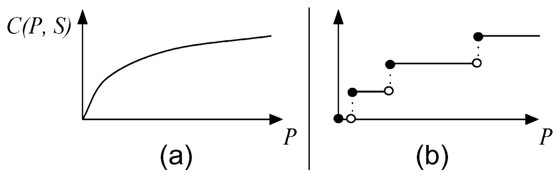


Fig. 1. Example $C(P,S)$ functions: (a) A function that is concave and increasing in P. (b) A piecewise constant function with the upper semi-continuity property.

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 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. Other examples include link budget functions corresponding to a finite number of modulation and coding schemes designed to achieve a sufficiently low probability of error, in which case the $C(P,S)$ function can be a discontinuous step function in P for each channel state S (see Fig. 1). Such discontinuous steps might also arise in the special case when all packets have fixed bit lengths and when $C(P,S)$ takes only a finite set of rates associated with transmitting an integral number of packets. Our analysis holds for all such $C(P,S)$ functions, and in particular we assume only that $C(P,S)$ satisfies the following *structural properties*:[1]

1) *Boundedness*: There is a maximum transmission rate μ_{max} such that $0 \leq C(P,S) \leq \mu_{max}$ for all $P \leq P_{max}$ and all $S \in \mathcal{S}$.
2) *Zero Rate for Zero Power*: $C(0,S) = 0$ for all channel states $S \in \mathcal{S}$.
3) *Upper Semi-Continuity*: For each channel state $S \in \mathcal{S}$, the $C(P,S)$ function is *upper semi-continuous* in the P variable. Specifically, for any power level P such that $0 \leq P \leq P_{max}$ and for any infinite sequence $\{P_n\}$ such that $\lim_{n\to\infty} P_n = P$, we have:

$$C(P,S) \geq \limsup_{n\to\infty} C(P_n, S)$$

We note that the class of upper semi-continuous functions includes all continuous functions, and also includes all piecewise continuous functions such that the function value at any point of discontinuity is equal to the largest of its limiting values at that point (see Fig. 1). This is a natural property for any practical rate-power curve, as a point of discontinuity usually represents a threshold point at which it is possible to support a larger transmission rate.

[1]The first structural property (boundedness) is the only one essential to our analysis. The other two properties can be removed without affecting the main results of this paper. They are used only to simplify exposition of Lemma 1 in Section III.

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 in terms of a finite 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:

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

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 $0 \leq \lambda < \mu_c$, and μ_c is the largest number with this property. 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 [2], 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 $\Omega(\sqrt{V})$.[2] 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

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

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. Dropping Scheme for Known System 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 [13]:

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 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.

Note that the expectations of the above lemma are taken with respect to the random channel state $S(t)$ and the potentially random power allocation that depends on $S(t)$. Such a policy could in principle be constructed with a-priori knowledge of λ and π_S for all $S \in \mathcal{S}$. It can be shown that if the structural properties 2 and 3 for the $C(P,S)$ function are removed, then the above lemma can be modified to state that there exists an infinite sequence of randomized power allocation policies $P_n^*(t)$ such that $\lim_{n\to\infty} \mathbb{E}\{P_n^*(t)\} = \Phi(\lambda)$ and $\lim_{n\to\infty} \mathbb{E}\{\mu_n^*(t)\} \geq \lambda$. This modified statement can also be used to prove our main results, although it is more convenient to use the simpler statement given in Lemma 1.

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:

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

This is the same queue update equation as (2), with the exception that any data exceeding the buffer size Q is necessarily dropped. Specifically, the amount of arrivals $\tilde{A}(t)$ admitted every slot is decided purely in terms of this finite buffer threshold, so that $\tilde{A}(t) = A(t)$ whenever adding all new $A(t)$ arrivals does not make total backlog exceed the threshold, and else $\tilde{A}(t)$ is equal to only that portion of the new arrivals that take backlog up to the Q threshold. 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. Indeed, every slot the expected new arrivals is given by $\mathbb{E}\{A(t)\} = \lambda$, which exceeds the expected transmission rate $\mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda$ (recall that ϵ is chosen so that $\rho + \epsilon < 1$). 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 [2], 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:

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

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, because $\Phi^* \triangleq \Phi(\rho\lambda)$, average power expenditure satisfies:

$$\overline{P} \leq \Phi^* + \Phi'(\lambda)\epsilon\lambda \tag{4}$$

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:

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

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:

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

where:

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

Proof: Note that we have $\tilde{\mu}(t) \leq \mu(t)$ for all t, with equality whenever $U(t) \geq \mu_{max}$. Hence:

$$\tilde{\mu}(t) \geq \mu(t) - \mu_{max}1_{[U(t)<\mu_{max}]} \tag{6}$$

where 1_X is an indicator function equal to 1 if event X is satisfied, and zero else. Inequality (6) can be verified as follows: If $U(t) < \mu_{max}$, then the right hand side of (6) is equal to $\mu(t) - \mu_{max}$, which is non-positive. Hence the inequality trivially holds in this case. Otherwise, $U(t) \geq \mu_{max}$, and the inequality (6) holds with equality. Taking expectations of (6) yields for all t:

$$\mathbb{E}\left\{\tilde{\mu}(t)\right\} \geq \mathbb{E}\left\{\mu(t)\right\} - \mu_{max}Pr\left[U(t) < \mu_{max}\right]$$

Summing over $\tau \in \{0,\ldots,t-1\}$, dividing by t, and taking the $\liminf$ as $t \to \infty$ yields the result. □

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:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\tilde{\mu}^*(\tau)\right\} \geq \rho\lambda + \epsilon\lambda - \alpha\mu_{max} \tag{7}$$

To ensure that the throughput is greater than or equal to $\rho\lambda$, from (7) 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:

$$\mathbb{E}\left\{A(t) - \mu^*(t)\right\} = \lambda - (\rho+\epsilon)\lambda = \lambda(1-\rho-\epsilon) \tag{8}$$

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. Formally, the fact that α decays exponentially in Q is shown by the following lemma.

Lemma 3: Given a queueing system with a finite buffer size Q and a positive drift that satisfies (8), there exists a positive constant θ^* such that the edge probability α satisfies:

$$\alpha \leq e^{-\theta^*(Q-\mu_{max})}$$

The above lemma follows from the Kingman bound [23], which also specifies the constant θ^*. The proof, together with a simple lower bound on θ^*, are given in Appendix B. Hence, if Q is chosen as follows:

$$Q \triangleq \mu_{max} + \frac{1}{\theta^*}\log\left(\frac{\mu_{max}}{\epsilon\lambda}\right) \tag{9}$$

then the lemma implies $\alpha \leq \epsilon\lambda/\mu_{max}$, ensuring from (7) that throughput is greater than or equal to $\rho\lambda$. Further, because $1/\epsilon = O(V)$ and $U(t) \leq Q$ for all t, it follows from (9) 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 mis-match 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.

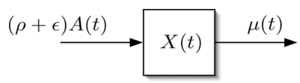


Fig. 2. An illustration of the virtual queueing system associated with the $X(t)$ update equation (10).

IV. On-Line Algorithm for Intelligent Packet Dropping

To construct an on-line algorithm for achieving a logarithmic energy-delay tradeoff, we use the theory of *performance optimal Lyapunov networking* [11]–[14]. 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 [13]. 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:

$$X(t+1) = \max[X(t) - \mu(t), 0] + (\rho + \epsilon)A(t) \tag{10}$$

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. 2).

Definition 1: A queueing system with unfinished work $X(t)$ is *strongly stable* if:

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{X(\tau)\} < \infty$$

It is not difficult to show that a strongly stable queue with an upper bounded transmission rate has the property that the $\liminf$ of the difference between the time average server rate and the time average arrival rate is non-negative [13]. Because the time average arrival rate to the $X(t)$ queue is given by $(\rho + \epsilon)\lambda$, we have the following lemma:

Lemma 4: If the $X(t)$ queue is strongly stable and the $A(t)$ process is i.i.d. with rate λ, then:

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\mu(\tau)\} \geq \rho\lambda + \epsilon\lambda \tag{11}$$

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\tilde{\mu}(\tau)\} \geq \rho\lambda + \epsilon\lambda - \alpha\mu_{max} \tag{12}$$

Proof: Because the $X(t)$ queue is strongly stable with an upper bounded transmission rate μ_{max}, the $\liminf$ difference between the time average server rate and arrival rate satisfies:

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

The inequality (11) follows from the above inequality together with the fact that $\mathbb{E}\{A(t)\} = \lambda$ for all t. The inequality (12) follows from (11) together with Lemma 2. □

A. Performance Optimal Lyapunov Networking

Our technique of stochastic queue optimization is based on the theory of *performance optimal Lyapunov networking*, which allows stability and performance optimization to be achieved via a single drift argument [11], [13], [14]. This extends the Lyapunov stability results of [15]–[22], and is closely related to *stochastic gradient optimization* (see, for example, [24] 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:[3]

$$\Delta(\boldsymbol{Z}(t)) \triangleq \mathbb{E}\{\Psi(\boldsymbol{Z}(t+1)) - \Psi(\boldsymbol{Z}(t)) | \boldsymbol{Z}(t)\} \tag{13}$$

We have the following important lemma, which is a modified version of similar results developed in [11], [13], [14].

Lemma 5: (Lyapunov Optimization [11], [13], [14]): If there is a process $B(t)$ and constants $\epsilon > 0$, $V \geq 0$, together with a non-negative function $f(\boldsymbol{Z})$, such that the queueing system satisfies the following drift inequality for all t and all $\boldsymbol{Z}(t)$:

$$\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} \leq B(t) - \epsilon f(\boldsymbol{Z}(t)) + VP^* \tag{14}$$

then:

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

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

where

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

[3]Strictly speaking, correct notation for the conditional Lyapunov drift in (13) 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 right hand side of (13).

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 5 to our queueing problem, let $\boldsymbol{Z}(t) = (U(t), X(t))$ represent the vector queue state of both the actual 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 (13). Motivated by Lemma 5, the goal of our dynamic control strategy is to choose $P(t)$ to minimize a bound on the following drift metric every timeslot t:

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

where $V \geq 1$ is a control parameter that effects the energy-delay performance of the algorithm. Note that this drift metric is simply the left hand side of (14).

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 and all $\boldsymbol{Z}(t)$:

$$\sigma^2 \geq \mathbb{E}\left\{\left(\mu(t) - A(t)\right)^2 |\boldsymbol{Z}(t)\right\} \tag{15}$$

Note that because $\mu(t) \leq \mu_{max}$ and $\mathbb{E}\{A(t)^2\} \leq \hat{A}^2_{max}$, choosing $\sigma^2 \triangleq \mu^2_{max} + \hat{A}^2_{max}$ ensures that (15) is satisfied for all t. A tighter bound is given by $\sigma^2 \triangleq \hat{A}^2_{max} + \max[0, \mu^2_{max} - 2\lambda\mu_{max}]$, which is useful in cases when the input rate λ is known. Likewise, if there exists a deterministic arrival bound A_{max} such that $A(t) \leq A_{max}$ for all t, then choosing $\sigma^2 \triangleq \max[\mu^2_{max}, A^2_{max}]$ also ensures that (15) is satisfied.

Lemma 6: If a positive constant ω is chosen to satisfy the inequality:

$$\omega e^{\omega\mu_{max}} \leq \lambda(1 - \rho - \epsilon)/\sigma^2 \tag{16}$$

then for all t we have the following bound on the drift metric:

$$\begin{aligned}\Delta\left(\boldsymbol{Z}(t)\right) &+ V\mathbb{E}\left\{P(t)|\boldsymbol{Z}(t)\right\} \\ &\leq B + V\mathbb{E}\left\{P(t)|\boldsymbol{Z}(t)\right\} - \omega e^{\omega(Q-U(t))} \\ &\quad \times \left[\lambda - \mathbb{E}\left\{\mu(t)|\boldsymbol{Z}(t)\right\} - \lambda(1-\rho-\epsilon)/2\right] \\ &\quad - X(t)\left[\mathbb{E}\left\{\mu(t)|\boldsymbol{Z}(t)\right\} - (\rho+\epsilon)\lambda\right]\end{aligned} \tag{17}$$

where $\mu(t) = C(P(t), S(t))$, and where:

$$B \triangleq \frac{\mu^2_{max} + (\rho+\epsilon)^2\hat{A}^2_{max}}{2} + 1 \tag{18}$$

Proof: The proof involves summing the Lyapunov drift expressions associated with the actual queue $U(t)$ and the virtual queue $X(t)$. These expressions are computed using the dynamic queue update equations (10) and (3). The detailed proof is given in Appendix A. □

It is not difficult to show that if ω is chosen as follows:

$$\omega \triangleq \frac{\lambda(1-\rho-\epsilon)}{\sigma^2}e^{-\lambda\mu_{max}(1-\rho-\epsilon)/\sigma^2} \tag{19}$$

then the inequality constraint (16) is satisfied, and hence the result of Lemma 6 holds. This follows directly from the fact that for any positive value c, the inequality $xe^x \leq c$ is always satisfied by the variable $x = ce^{-c}$. If λ is unknown, then any positive lower bound λ_0 (such that $0 < \lambda_0 \leq \lambda$) can be used in replacement for λ in (19) while still ensuring that (16) is satisfied.

We design the dynamic control policy to minimize the drift bound given in the right hand side of (17) every timeslot, considering all possible power allocation options $P(t)$ such that $0 \leq P(t) \leq P_{max}$. Isolating the terms on the right hand side of (17) that depend on the control variable $P(t)$ (noting that $\mu(t) = C(P(t), S(t))$), it is clear that minimizing the bound in (17) is equivalent to choosing $P(t)$ in reaction to the current channel state and the current queue backlogs in order to maximize the following expression every timeslot:

$$\mathbb{E}\left\{\left(X(t) - \omega e^{\omega(Q-U(t))}\right)C\left(P(t), S(t)\right) - VP(t)|\boldsymbol{Z}(t)\right\}$$

Maximizing the above conditional expectation is accomplished by deterministically maximizing the resulting expression corresponding to the particular channel state realization $S(t)$ and the particular queue state $\boldsymbol{Z}(t) = (U(t), X(t))$ observed on the current timeslot t. This leads to the following dynamic policy, which uses a control parameter $V \geq 1$ and uses fixed parameters Q, ϵ, ω to be determined later in terms of V and ρ.

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:

$$\begin{aligned}\text{Maximize} : &\quad C\left(P, S(t)\right)\left(X(t) - \omega e^{\omega(Q-U(t))}\right) - VP \\ \text{Subject to} : &\quad 0 \leq P \leq P_{max}\end{aligned}$$

2) Iterate the virtual queue $X(t)$ according to (10), using $\mu(t) = C(P(t), S(t))$.

3) Emulate the finite buffer queue $U(t)$ according to (3).

Note that the power allocation step in the above control policy involves a simple optimization of a function of one variable, and can easily be solved in real time for most practical $C(P, S)$ functions. For example, if $C(P, S)$ is concave and differentiable in P for all channel states S (as in (1)), then the optimal $P(t)$ value can be solved simply by taking a derivative and setting $P(t)$ to the local maximum found on the interval $0 \leq P \leq P_{max}$ (possibly achieved at the endpoints $P = 0$ or $P = P_{max}$). If $C(P, S)$ is piecewise constant with a fixed number of transmission rate options (and hence a fixed number of power options), then the solution is found simply by comparing each option.

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 (16), and $Q = \log(xV)/\omega$, where x is any value that satisfies:

$$x \geq \frac{4\mu_{max}e^{\omega\mu_{max}}B}{\lambda^2\omega(1-\rho-\epsilon)(1-\rho)} \tag{20}$$

then:

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

We prove the above theorem in the next section. 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 (16) and (20) 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. Hence, the algorithm can easily adapt to the situation where the channel state probabilities change between periods of network operation. 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. This is discussed further in Sections VIII and IX, where simulation results are presented and a modified adaptive threshold policy is introduced.

V. Performance Analysis

Here we prove Theorem 1. Note that the dynamic policy is designed to minimize the right hand side of the drift bound (17) over all possible power allocation policies. In particular, the resulting bound is less than or equal to the corresponding expression associated with any alternative power allocation policy $P^*(t)$. Thus, for any policy where $P^*(t)$ is randomly chosen every slot in reaction to the current channel state $S(t)$ but independently of the current queue state $Z(t) = (U(t), X(t))$, we have:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\leq B + V\mathbb{E}\{P^*(t)\}\\ &- \omega e^{\omega(Q-U(t))} \times [\lambda - \mathbb{E}\{\mu^*(t)\} - \lambda(1-\rho-\epsilon)/2]\\ &- X(t)[\mathbb{E}\{\mu^*(t)\} - (\rho+\epsilon)\lambda]\end{aligned} \tag{21}$$

where $\mu^*(t) = C(P^*(t), S(t))$. We emphasize that the $P(t), U(t), X(t)$ variables used in the above inequality correspond to the dynamic control policy under investigation (and are the same as those in (17)), while the new variables $P^*(t)$ and $\mu^*(t)$ are used as replacements in the right hand side of (17) and correspond to an alternative power assignment.

Consider now the particular randomized policy $P^*(t)$ that allocates power in reaction to the current channel state $S(t)$ (but independently of queue backlog) to yield the following for all t:

$$\mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda \tag{22}$$

$$\mathbb{E}\{P^*(t)\} = \Phi((\rho+\epsilon)\lambda) \tag{23}$$

Such a policy is guaranteed to exist by Lemma 1. Using (22) and (23) in (21) yields:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\leq B + V\Phi((\rho+\epsilon)\lambda)\\ &-\omega e^{\omega(Q-U(t))}\lambda(1-\rho-\epsilon)/2\end{aligned} \tag{24}$$

The above drift inequality is in the exact form as given in the Lyapunov Optimization Lemma (Lemma 5). Directly applying the lemma yields:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{P(\tau)\} \leq \Phi((\rho+\epsilon)\lambda) + B/V \tag{25}$$

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{e^{\omega(Q-U(\tau))}\right\} \leq \frac{B + V\left[\Phi((\rho+\epsilon)\lambda) - \overline{P}_{inf}\right]}{\omega\lambda(1-\rho-\epsilon)/2} \tag{26}$$

Proof of Part (a) of Theorem 1: Recall that $\Phi^* \triangleq \Phi(\rho\lambda)$. Thus, from (25) we have:

$$\begin{aligned}\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{P(\tau)\} &\leq \Phi(\rho\lambda+\epsilon\lambda) + B/V\\ &\leq \Phi^* + \Phi'(\lambda)\epsilon\lambda + B/V\\ &\leq \Phi^* + O(1/V)\end{aligned}$$

where the final inequality follows because $\epsilon = (1-\rho)/(2V)$. □

Proof of Part (b) of Theorem 1: Note that the finite buffer queueing update equation in (3) ensures that $U(t) \leq Q$ for all t. The result follows by noting that $Q = O(\log(V))$. □

Proof of Part (c) of Theorem 1: To prove part (c), we first make the following claims:

Claim 1: The $X(t)$ queue is strongly stable.

Claim 2: The edge probability α (defined in (5)) satisfies $\alpha \leq \lambda\epsilon/\mu_{max}$.

The claims are proven at the end of this section. Because the $X(t)$ queue is stable, from Lemma 4 of the previous section we have that the time average system throughput satisfies:

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

where the final inequality follows from Claim 2. □

It remains only to prove Claims 1 and 2.

Proof: (Claim 1): Consider again the drift bound in (21). However, instead of considering a power allocation policy $P^*(t)$ that satisfies (22) and (23), we consider an alternative policy $P^*(t)$ that also makes randomized decisions based only on the current channel state $S(t)$, but which yields:

$$\mathbb{E}\{\mu^*(t)\} = \lambda(1+\rho+\epsilon)/2$$
$$\mathbb{E}\{P^*(t)\} = \Phi(\lambda(1+\rho+\epsilon)/2)$$

Again, such a policy is guaranteed to exist by Lemma 1. Using the above expressions in (21) yields:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\leq B + VP_{max}\\ &-X(t)\lambda(1-\rho-\epsilon)/2\end{aligned}$$

Because $(1-\rho-\epsilon)>0$, the above drift expression is in the exact form for application of Lemma 5. We thus have:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{X(\tau)\right\}\leq\frac{B+VP_{max}}{\lambda(1-\rho-\epsilon)/2}$$

proving that $X(t)$ is strongly stable. □

Proof: (Claim 2): For any timeslot τ and for any distribution on the random variable $U(\tau)$, we have:

$$\begin{aligned}\mathbb{E}\left\{e^{\omega(Q-U(\tau))}\right\}\geq&\,\mathbb{E}\left\{e^{\omega(Q-U(\tau))}|U(\tau)<\mu_{max}\right\}\\&\times Pr\left[U(\tau)<\mu_{max}\right]\\\geq&\,e^{\omega(Q-\mu_{max})}Pr\left[U(\tau)<\mu_{max}\right]\end{aligned}$$

Summing the above inequality over $\tau\in\{0,\ldots,t-1\}$ yields:

$$e^{\omega(Q-\mu_{max})}\sum_{\tau=0}^{t-1}Pr\left[U(\tau)<\mu_{max}\right]\leq\sum_{\tau=0}^{t-1}\mathbb{E}\left\{e^{\omega(Q-U(\tau))}\right\}$$

Dividing both sides by t, taking the $\limsup$, and using the inequality (26) yields:

$$e^{\omega(Q-\mu_{max})}\alpha\leq\frac{B+V\left[\Phi\left((\rho+\epsilon)\lambda\right)-\overline{P}_{inf}\right]}{\omega\lambda(1-\rho-\epsilon)/2}\tag{27}$$

where α is defined in (5).

However, from Claim 1 we know that the $X(t)$ queue is stable, and hence the $\liminf$ of the time average energy used to stabilize the queue must be greater than or equal to the minimum average energy required for stability [13]. Because the transmission rate of the $X(t)$ queue is given by $\mu(t)=C(P(t),S(t))$ and the average input rate is given by $(\rho+\epsilon)\lambda$ (see (10) and Fig. 2), the $\liminf$ of the expended energy is given by $\overline{P}_{inf}$, and the minimum energy for stability is given by $\Phi((\rho+\epsilon)\lambda)$. Therefore, we have $\overline{P}_{inf}\geq\Phi((\rho+\epsilon)\lambda)$. Using this fact in (27) yields:

$$e^{\omega(Q-\mu_{max})}\alpha\leq\frac{B}{\omega\lambda(1-\rho-\epsilon)/2}$$

Therefore:

$$\alpha\leq\left[\frac{e^{\omega\mu_{max}}B}{\omega\lambda(1-\rho-\epsilon)/2}\right]e^{-\omega Q}$$

Using the fact that $Q\triangleq\log(xV)/\omega$, we have:

$$\alpha\leq\left[\frac{e^{\omega\mu_{max}}B}{\omega\lambda(1-\rho-\epsilon)/2}\right]\frac{1}{Vx}$$

Using (20) to replace the x variable in the right hand side of the above inequality yields:

$$\alpha\leq\frac{\lambda(1-\rho)}{2\mu_{max}V}$$

Using the fact that $\epsilon\triangleq(1-\rho)/(2V)$ in the above inequality yields $\alpha\leq\lambda\epsilon/\mu_{max}$, proving the claim. □

VI. Necessity of the Logarithmic Tradeoff

The logarithmic energy-delay tradeoff may seem to be an artifact of the exponential Lyapunov function $L(U)$ that was used, so that another Lyapunov function (perhaps doubly exponential) could perhaps offer sub-logarithmic performance. However, this is not the case. Here we present a class of systems for which the optimal energy-delay tradeoff is necessarily logarithmic, and hence this tradeoff is fundamental.

We consider the special case of a system with *no channel variation*, so that the rate-power function is given by $C(P)$. Further, we assume the system has the following properties:

1) Arrivals $A(t)$ are i.i.d. over timeslots, and there exists a probability $q>0$ such that $Pr[A(t)=0]=q$.
2) All admission/rejection decisions are made immediately upon arrival, so that admitted data is necessarily served.
3) The minimum average power function $\Phi(x)$ is non-linear over the interval $\rho\lambda/2\leq x\leq\rho\lambda$, and hence (by convexity):
$$\Phi'(\rho\lambda)>\frac{\Phi(\rho\lambda)-\Phi(\rho\lambda/2)}{\rho\lambda/2}\tag{28}$$
where $\Phi'(\rho\lambda)$ is the right derivative of the minimum energy function at the point $\rho\lambda$.

We further restrict attention to the class of ergodic scheduling policies with well defined time averages.

Theorem 2: If a control policy of the type described above yields a throughput of at least $\rho\lambda$ and has an average energy expenditure $\overline{P}$ such that:

$$\overline{P}-\Phi(\rho\lambda)\leq 1/V\tag{29}$$

then average congestion (and hence average delay) must be greater than or equal to $\Omega(\log(V))$.

Proof: Assume that $\rho\lambda$ satisfies (28), and define the constant β as follows:

$$\beta\triangleq\Phi'(\rho\lambda)-\frac{\Phi(\rho\lambda)-\Phi(\rho\lambda/2)}{\rho\lambda/2}$$

Note that $\beta>0$. Consider a control policy as described in the statement of the theorem. Let $\overline{U}$ and $\overline{P}$ represent the time average queue backlog and the time average power expenditure, respectively. Assume that (29) holds. Further, define δ as the time average fraction of time that $\tilde{\mu}(t)<\rho\lambda/2$, where $\tilde{\mu}(t)$ is the actual amount of data transmitted during slot t. That is:

$$\delta\triangleq\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}1_{[\tilde{\mu}(\tau)<\rho\lambda/2]}$$

where 1_X is an indicator function that is 1 whenever condition X is true, and is zero otherwise. We make the following claims (proven below):

Claim 1: $\delta\leq 2/(V\beta\rho\lambda)$

Claim 2: There exist positive constants C and c (that do not depend on V or $\overline{U}$) such that: $\delta\geq Cq^{c\overline{U}}$

Combining Claims 1 and 2 yields:

$$Cq^{c\overline{U}}\leq 2/(V\beta\rho\lambda)$$

Taking the logarithm of both sides and shifting terms yields:

$$\overline{U}\geq\frac{\log(V\beta\rho\lambda C/2)}{c\log(1/q)}$$

establishing the result. □

To complete the proof, below we prove Claims 1 and 2.

Proof: (Claim 1): Define $\overline{\mu}_1$ as the conditional time average rate of $\tilde{\mu}(t)$ given that $\tilde{\mu}(t) \geq \rho\lambda/2$, and define $\overline{P}_1$ as the conditional time average power expenditure associated with such transmissions. Similarly, define $\overline{\mu}_2$ and $\overline{P}_2$ as the conditional time averages given that $\tilde{\mu}(t) < \rho\lambda/2$.

Note by definition that $0 \leq \overline{\mu}_2 \leq \rho\lambda/2$. Because total throughput is greater than or equal to $\rho\lambda$, we have:

$$\rho\lambda \leq (1-\delta)\overline{\mu}_1 + \delta\overline{\mu}_2$$

Rearranging terms in the above inequality yields:

$$\overline{\mu}_1 \geq \rho\lambda + \frac{\delta(\rho\lambda - \overline{\mu}_2)}{(1-\delta)} \tag{30}$$

Likewise, we have:

$$\begin{aligned}\overline{P} &= (1-\delta)\overline{P}_1 + \delta\overline{P}_2 \\ &\geq (1-\delta)\Phi(\overline{\mu}_1) + \delta\Phi(\overline{\mu}_2)\end{aligned} \tag{31}$$

where (31) follows because $\Phi(x)$ is defined as the minimum average energy required to support an average transmission rate of x, and hence is less than or equal to the average power of any particular strategy that achieves a transmission rate of at least x. Specifically, recall here that the channel is static, so that the power allocation strategy used to achieve the conditional transmission rate $\overline{\mu}_1$ over the special slots in which $\tilde{\mu}(t) \geq \rho\lambda/2$ can be used over *any slot* to achieve an unconditional time average transmission rate of at least $\overline{\mu}_1$ with a time average power equal to $\overline{P}_1$. Using (30) in (31) and noting that $\Phi(x)$ is non-decreasing and convex, we have:

$$\begin{aligned}\overline{P} &\geq (1-\delta)\Phi\left(\rho\lambda + \frac{\delta(\rho\lambda - \overline{\mu}_2)}{(1-\delta)}\right) + \delta\Phi(\overline{\mu}_2) \\ &\geq (1-\delta)\left[\Phi(\rho\lambda) + \Phi'(\rho\lambda)\frac{\delta(\rho\lambda - \overline{\mu}_2)}{(1-\delta)}\right] + \delta\Phi(\overline{\mu}_2) \\ &= \Phi(\rho\lambda) + \delta(\rho\lambda - \overline{\mu}_2)\left[\Phi'(\rho\lambda) - \frac{\Phi(\rho\lambda) - \Phi(\overline{\mu}_2)}{\rho\lambda - \overline{\mu}_2}\right] \\ &\geq \Phi(\rho\lambda) + \delta(\rho\lambda - \overline{\mu}_2)\left[\Phi'(\rho\lambda) - \frac{\Phi(\rho\lambda) - \Phi(\rho\lambda/2)}{\rho\lambda/2}\right] \\ &= \Phi(\rho\lambda) + \delta(\rho\lambda - \overline{\mu}_2)\beta\end{aligned}$$

where the inequalities follow by convexity of $\Phi(x)$ together with the fact that $0 \leq \overline{\mu}_2 \leq \rho\lambda/2$. Therefore, $\overline{P} \geq \Phi(\rho\lambda) + \delta\beta\rho\lambda/2$, and hence by (29) we have $1/V \geq \delta\beta\rho\lambda/2$, proving the claim. □

Proof: (Claim 2): Let t be a time at which the system is in steady state, so that $\mathbb{E}\{U(t)\} = \overline{U}$. By the Markov inequality, we have $Pr[U(t) \leq 2\overline{U}] \geq 1/2$. The probability that $U(t) \leq 2\overline{U}$ and that we then have k consecutive slots over which there are no arrivals is thus at least $(1/2)q^k$. Let $k = \lceil 2\overline{U}/(\rho\lambda/2) + 1 \rceil$. If there are no arrivals over this set of k slots, we know there must be at least one of the k slots in which fewer than $\rho\lambda/2$ units of data were served. Indeed, if all k slots served at least $\rho\lambda/2$ units of data, then the total amount of transmitted data over these slots would be at least $k\rho\lambda/2$, which is greater than $2\overline{U}$ and hence a contradiction.

Without loss of generality, assume the system is in steady state at time 0, and divide the timeline into successive frames of size k slots. The time average rate δ at which we serve fewer than $\rho\lambda/2$ bits is greater than or equal to $(1/k)$ times the probability that a particular frame experiences such an event, and so $\delta \geq (1/2)q^k/k$. Thus:

$$\begin{aligned}\delta &\geq (1/2)q^k/k \\ &\geq (1/2)q^{\lceil 4\overline{U}/(\rho\lambda)+2\rceil} / \lceil 4\overline{U}/(\rho\lambda) + 2 \rceil \\ &= \frac{\tilde{C}q^{\tilde{c}\overline{U}}}{\overline{U} + \theta}\end{aligned}$$

where $\tilde{C} = \rho\lambda q^2/8$, $\tilde{c} = 4/(\rho\lambda)$, and $\theta = \rho\lambda/2$. Let $d = 1/\log(1/q)$. It is not difficult to show that $q^{d(\overline{U}+\theta)} \leq 1/(\overline{U}+\theta)$ (using the fact that $e^{-x} \leq 1/x$ for all $x > 0$), and hence

$$\delta \geq \tilde{C}q^{\tilde{c}\overline{U}}q^{d(\overline{U}+\theta)} = Cq^{c\overline{U}}$$

where we defined $c = \tilde{c} + d$ and $C = \tilde{C}q^{d\theta}$. This proves Claim 2. □

While Theorem 2 is presented for the case of static channels, we conjecture that the analysis can be extended to prove that the logarithmic delay tradeoff is also necessary in the case of dynamic channels.

VII. Multi-User Systems

The algorithm can easily be extended to multi-user systems with L links with a vector link state process $\boldsymbol{S}(t) = (S_1(t), \ldots, S_L(t))$, a vector valued rate-power function $\boldsymbol{C}(\boldsymbol{P}(t), \boldsymbol{S}(t))$, and a vector arrival process $\boldsymbol{A}(t) = (A_1(t), \ldots, A_L(t))$. In this case, we have actual queues $\boldsymbol{U}(t) = (U_1(t), \ldots, U_L(t))$ and virtual queues $\boldsymbol{X}(t) = (X_1(t), \ldots, X_L(t))$, each of which is updated according to queueing equations similar to (3) and (10). Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_L)$ represent the input rate vector, and for simplicity assume that $\lambda_i > 0$ for all i. The minimum energy function $\Phi(\boldsymbol{\lambda})$ for this multi-user problem specifies the minimum average sum power expended by the system, minimized over all possible policies that support the input rates $\boldsymbol{\lambda}$. For a given value $\rho < 1$, the goal is to design a strategy that ensures a throughput vector of at least $\rho\boldsymbol{\lambda}$, while pushing average power expenditure arbitrarily close to the target value $\Phi^* = \Phi(\rho\boldsymbol{\lambda})$.

Suppose that the power vector $\boldsymbol{P}(t)$ is contained within a compact set Π every timeslot, where Π represents the set of acceptable power allocation vectors. The multi-user policy observes the current channel state $\boldsymbol{S}(t)$ and the current queue backlogs $\boldsymbol{X}(t)$ and $\boldsymbol{U}(t)$, and chooses a power vector $\boldsymbol{P}(t) = (P_1, \ldots, P_L)$ every slot to optimize:

$$\text{Maximize}: \sum_{i=1}^{L}\left[C_i(\boldsymbol{P}, \boldsymbol{S}(t))\left(X_i(t) - \omega_i e^{\omega_i(Q_i - U_i(t))}\right) - VP_i\right]$$

subject to $\boldsymbol{P}(t) \in \Pi$. The virtual queues $X_i(t)$ are then updated as follows:

$$X_i(t+1) = \max[X_i(t) - \mu_i(t), 0] + (\rho + \epsilon)A_i(t)$$

where $\mu_i(t) = C_i(\boldsymbol{P}(t), \boldsymbol{S}(t))$. Each $U_i(t)$ queue operates according to the finite buffer queueing equation (3) with a buffer size Q_i and with an input process $A_i(t)$ and server rate process $\mu_i(t)$. This multi-channel, multi-user algorithm yields a logarithmic energy-delay tradeoff for appropriately chosen $\{Q_i\}, \{\omega_i\}, \epsilon$ values. Specifically, for each $i \in \{1, \ldots, L\}$, let

σ_i^2 be any constant such that the following holds for all $\boldsymbol{Z}(t)$ (where $\boldsymbol{Z}(t) \triangleq [\boldsymbol{U}(t); \boldsymbol{X}(t)]$):

$$\sigma_i^2 \geq \mathbb{E}\left\{(\mu_i(t) - A_i(t))^2 \,|\boldsymbol{Z}(t)\right\}$$

Further suppose that for all i, the the constants ω_i are chosen to be positive and to satisfy:

$$\omega_i e^{\omega_i \mu_{max}} \leq \lambda_i(1-\rho-\epsilon)/\sigma_i^2 \tag{32}$$

Theorem: (Multi-User Packet Dropping): For a given value $\rho < 1$ and a fixed control parameter $V \geq 1$, if parameters $\epsilon, \{\omega_i\}$, and $\{Q_i\}$ are chosen so that $\epsilon = (1-\rho)/(2V)$, each ω_i is positive and satisfies (32), and $Q_i = \log(x_i V)/\omega_i$, where x_i is any value that satisfies:

$$x_i \geq \frac{4\mu_{max} L B e^{\omega_i \mu_{max}}}{\lambda_i^2 \omega_i (1-\rho-\epsilon)(1-\rho)}$$

where B is defined in (18), then:

(a) $\limsup_{t\rightarrow\infty}(1/t)\sum_{\tau=0}^{t-1}\sum_{i=1}^{L}\mathbb{E}\{P_i(\tau)\} \leq \Phi^* + O(L/V)$

(b) $U_i(t) \leq Q_i$ for all t, where $Q_i = O(\log(V))$ for all i

(c) $\liminf_{t\rightarrow\infty}(1/t)\sum_{\tau=0}^{t-1}\mathbb{E}\{\tilde{\mu}_i(\tau)\} \geq \rho\lambda_i$ for all i

The proof of the above theorem is similar to the proof for the single-user case, and is omitted for brevity.

VIII. Adaptive-Threshold Policy for Constant-Factor Delay Improvement

Our dynamic packet-dropping policy was proven to achieve an optimal $[O(1/V), O(\log(V))]$ energy-delay tradeoff. Specifically, the policy ensures that no more than a fraction $1-\rho$ of packets are dropped, that time average power expenditure is within $O(1/V)$ from optimality, and that queue backlog $U(t)$ satisfies $U(t) \leq Q$ for all time t, where $Q = O(\log(V))$ (see Theorem 1). However, while the Q parameter is logarithmic in V, it was chosen to have a constant coefficient that is large enough to analytically ensure that the edge probability α satisfies $\alpha \leq \lambda\epsilon/\mu_{max}$. Our analysis was conservative, and in simulations it was observed that edge events were very rare. Indeed, it was found that the condition $\alpha \leq \lambda\epsilon/\mu_{max}$ was typically satisfied under the much less conservative threshold $\tilde{Q} = Q/15$ (see simulations in Section IX). This demonstrates that, while the policy achieves the optimal $[O(1/V), O(\log(V))]$ asymptotic performance, average delay can be further reduced by a significant constant factor.

In this section we design a modified policy that preserves the same analytical delay guarantees, but that adaptively adjusts the Q parameter to yield significant constant factor improvements in delay, as observed in the simulations of Section IX. The policy uses a novel type of virtual queue that accumulates the excess amount by which the edge probability has violated its time average constraint. The Q parameter is adjusted in response to this virtual queue. For simplicity, in this section we consider only the single-queue problem, although the technique readily extends to the multi-user scenario treated in Section VII.

A. Threshold-Adaptive Packet Dropping Policy

Fix the acceptance ratio $\rho < 1$. Again let V be a positive control parameter that affects the energy-delay tradeoff, and define constants ϵ, ω, and x as in Theorem 1, where $\epsilon = (1-\rho)/(2V)$, ω is positive and satisfies (16), and x is any positive value that satisfies (20). The modified algorithm uses a time varying maximum buffer size $Q(t)$, with queueing update equation as follows:

$$U(t+1) = \min\left[Q(t), \max\left[U(t)-\mu(t), 0\right] + A(t)\right] \tag{33}$$

The value of $Q(t)$ is chosen so that $Q_{min} \leq Q(t) \leq Q_{max}$ for all t, where $Q_{max} = \log(xV)/\omega$. That is, Q_{max} is the same as the constant value Q used in the original algorithm of Theorem 1. We set the minimum Q value to be $Q_{min} = \max[Q_{max}/f, 10\mu_{max}]$, where f is a suitably defined constant reduction factor. In our simulations we choose the reduction factor f to be between 15 and 40. The $Q(t)$ value is modified over time in response to a new virtual queue, as follows: Let $Y(t)$ be a new virtual queueing process with $Y(0) = 0$, and with update equation:

$$Y(t+1) = \max[Y(t) - \epsilon\lambda/\mu_{max}, 0] + 1_\alpha(t) \tag{34}$$

where $1_\alpha(t)$ is an indicator function that is 1 if and only if an edge event occurs on slot t:

$$1_\alpha(t) = \begin{cases} 1 & \text{if } U(t) < \mu_{max} \\ 0 & \text{if } U(t) \geq \mu_{max} \end{cases}$$

It is clear that if the virtual queue $Y(t)$ is stable, then the $\limsup$ time average edge probability α is less than or equal to $\epsilon\lambda/\mu_{max}$, as desired. Let θ be a positive value that defines an acceptable number of timeslots by which the time average edge probability can exceed its desired upper bound $\epsilon\lambda/\mu_{max}$ (we use $\theta = 5$ throughout). The threshold-adaptive algorithm is designed so that $Y(t)$ consistently returns to a value below θ. This is accomplished by ensuring that $Q(t)$ increases and remains at Q_{max} whenever $Y(t)$ is above the θ threshold for a long duration of time.

Specifically, let $s = (Q_{max} - Q_{min})/100$ be the additive increment by which $Q(t)$ can be increased (or decreased), so that:

$$Q(t) \in \{Q_{min}, Q_{min}+s, Q_{min}+2s, \ldots, Q_{max}\}$$

Initialize the adaptive Q threshold to $Q(0) = Q_{min}$. For each slot $t \in \{0, 1, 2, \ldots\}$, let $Q_{cap}(t)$ be the smallest value of the adaptive Q threshold under which no edge event has occurred (so that there exists a time $\tau \leq t$ such that $U(\tau) < \mu_{max}$ and $Q(\tau) = Q_{cap}(t) - s$). Initialize $Q_{cap}(0) = Q_{min} + s$. Every slot t, after the queueing equation (33) is updated, $Q(t)$ is updated to $Q(t+1)$ as follows:

- $Q(t+1) = \min[Q(t)+s, Q_{max}]$ if $Y(t) > \theta$, if $U(t+1) = Q(t)$, and $Q(t) < Q_{cap}(t)$.
- $Q(t+1) = \max[Q(t)-s, Q_{min}]$ if $Y(t) \leq \theta$, if $U(t+1) \leq Q(t) - s$, and no edge event has occurred within the past 1000 slots.
- $Q(t+1) = Q(t)$ if neither of the above events are satisfied.

Intuitively, the above policy decreases $Q(t)$ when $Y(t)$ is small, as a small $Y(t)$ indicates a low time average number of edge events and suggests the current Q value is unnecessarily large. Note that $Q(t)$ can only decrease if no edge events have occurred within the past 1000 slots (a heuristic that enables a quicker response to combat edge events), and is not decreased unless $U(t+1) \leq Q(t) - s$ (so that decreasing the Q value would not force any data to be dropped on slot $t+1$). Alternatively, $Q(t)$ is increased when $Y(t)$ is large in order to reduce the future rate of edge events. The $Q_{cap}(t)$ parameter is added so that $Q(t)$ only increases when an edge event has actually occurred under a Q value that was greater than or equal to its current value. This is useful because edge events typically disappear altogether once Q is large enough, so that there is no value in making Q larger unless more edge events are experienced. The requirement that $Q(t)$ cannot be increased unless $U(t+1) = Q(t)$ ensures that the threshold is only adjusted when queue backlog is at the level of the old threshold, a requirement that plays a crucial role in our Lyapunov analysis of the above strategy (provided in the next sub-section).

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

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

$$\text{Maximize}: \; C(P, S(t))\left(X(t) - \omega e^{\omega(Q(t)-U(t))}\right) - VP$$
$$\text{Subject to}: \; 0 \le P \le P_{max}$$

2) Iterate the virtual queue $X(t)$ according to (10), using $\mu(t) = C(P(t), S(t))$.
3) Iterate the virtual queue $Y(t)$ according to (34).
4) Emulate the finite buffer queue $U(t)$ according to (33).
5) Update the $Q(t)$ value as specified above.

Theorem 4: (Threshold-Adaptive Algorithm Performance): For any parameter $V > 0$, for values ϵ, ω, x chosen as above, and for any positive θ value, the threshold-adaptive packet dropping policy yields the same guarantees specified in parts (a), (b), and (c) of Theorem 1 for the original packet dropping algorithm.

Thus, the threshold-adaptive policy preserves the same analytical delay guarantees as the original algorithm, but experimentally achieves much better delay performance (see simulations in Section IX). Choosing a threshold value $\theta = 5$ allows the time average edge rate to be 5 slots over its desired rate before $Q(t)$ is increased. A larger θ value would make $Q(t)$ less likely to increase, at the expense of expanding the amount of time required for the time average edge probability to begin "averaging out" to a value less than $\epsilon\lambda/\mu_{max}$. The adaptive $Q(t)$ threshold creates a non-trivial modification of the system dynamics of the original algorithm, and below we present a proof of Theorem 4.

B. Performance Analysis of the Threshold-Adaptive Policy

Define the following *time-dependent* Lyapunov function: $L(U(t), t) = e^{\omega(Q(t)-U(t))}$. Again define the vector of queue backlogs $\boldsymbol{Z}(t) \triangleq (U(t), X(t))$, and define:

$$\Psi(\boldsymbol{Z}(t), t) \triangleq L(U(t), t) + \frac{1}{2}(X(t))^2$$

Using the same drift analysis as Lemma 6 (in Appendix A), we have:

$$\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} = \Delta_L(\boldsymbol{Z}(t)) + \Delta_J(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\}$$

where $\Delta_J(\boldsymbol{Z}(t))$ satisfies the same bound as in part (a) of Lemma 7. The computation to bound $\Delta_L(\boldsymbol{Z}(t))$ is similar to that given in Appendix A: Because $U(t+1) \ge \min[Q(t), U(t) - \mu(t) + A(t)]$, the inequality (37) from Appendix A is modified to:

$$e^{\omega(Q(t)-U(t+1))} \le e^{\omega(Q(t)-U(t))} e^{\omega(\mu(t)-A(t))} + 1$$

and hence:

$$e^{\omega(Q(t+1)-U(t+1))} \le \left[e^{\omega(Q(t+1)-U(t+1))} - e^{\omega(Q(t)-U(t+1))}\right] + e^{\omega(Q(t)-U(t))} e^{\omega(\mu(t)-A(t))} + 1 \quad (35)$$

The term in brackets on the right hand side of (35) can be bounded by:

$$\left[e^{\omega(Q(t+1)-U(t+1))} - e^{\omega(Q(t)-U(t+1))}\right] \le e^{\omega s} - 1 \quad (36)$$

Inequality (36) is true because the left hand side is only positive if the threshold is increased on slot t (so that $Q(t) < Q(t+1)$), and any increase is by exactly s and occurs only when $U(t+1) = Q(t)$. Therefore, using (35) and (36) and following the same steps as in Appendix A, we have the following bound:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\le B + (e^{\omega s} - 1) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} - \omega e^{\omega(Q(t)-U(t))} \\ &\quad \times [\lambda - \mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - \lambda(1-\rho-\epsilon)/2] \\ &\quad - X(t)[\mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - (\rho+\epsilon)\lambda]\end{aligned}$$

Because the threshold-adaptive dropping policy makes power allocation decisions to minimize the right hand side of the above drift expression, we have:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\le B + (e^{\omega s} - 1) + V\mathbb{E}\{P^*(t)|\boldsymbol{Z}(t)\} - \omega e^{\omega(Q(t)-U(t))} \\ &\quad \times [\lambda - \mathbb{E}\{\mu^*(t)|\boldsymbol{Z}(t)\} - \lambda(1-\rho-\epsilon)/2] \\ &\quad - X(t)[\mathbb{E}\{\mu^*(t)|\boldsymbol{Z}(t)\} - (\rho+\epsilon)\lambda]\end{aligned}$$

where $P^*(t)$ corresponds to any alternative feasible power allocation policy, and $\mu^*(t) = C(P^*(t), S(t))$. Consider the queue backlog-independent policy from (22) and (23) that yields $\mathbb{E}\{\mu^*(t)\} = (\rho+\epsilon)\lambda$ and $\mathbb{E}\{P^*(t)\} = \Phi((\rho+\epsilon)\lambda)$. We thus have:

$$\begin{aligned}\Delta(\boldsymbol{Z}(t)) + V\mathbb{E}\{P(t)|\boldsymbol{Z}(t)\} &\le B + (e^{\omega s} - 1) \\ &\quad + V\Phi((\rho+\epsilon)\lambda) - \omega e^{\omega(Q(t)-U(t))}[\lambda(1-\rho-\epsilon)/2]\end{aligned}$$

The above drift inequality holds for all time t. Using Lemma 5, we have (defining $\tilde{B} \triangleq B + (e^{\omega s} - 1)$):

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{P(\tau)\} \le \Phi((\rho+\epsilon)\lambda) + \tilde{B}/V$$
$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{e^{\omega(Q(\tau)-U(\tau))}\right\} \le \frac{\tilde{B} + V\left[\Phi((\rho+\epsilon)\lambda) - \overline{P}_{inf}\right]}{\omega\lambda(1-\rho-\epsilon)/2}$$

Repeating the same argument from Theorem 1, which shows $\Phi((\rho + \epsilon)\lambda) \leq \Phi^* + O(1/V)$, yields part (a) of Theorem 4. Now note that part (b) of the Theorem 4 follows immediately by noting that $U(t) \leq Q_{max}$ for all t. Finally, similar to the proof of Theorem 1, it can be shown that the virtual queue $X(t)$ is stable, and thus that the time average throughput is greater than or equal to $\rho\lambda$ whenever the time average edge probability α is less than or equal to $\epsilon\lambda/\mu_{max}$. This latter property occurs whenever the virtual queue $Y(t)$ is rate stable. However, the queue $Y(t)$ is *always* rate stable, because if $Y(t)$ spends a long duration above the θ threshold, then $Q(t)$ will rise to Q_{max}, and the system will run according to the dynamics of the original policy until either $Y(t)$ again drops below the θ threshold (ensuring a time average edge probability that is within θ/t of the desired bound), or until the dynamics of the original policy create an edge probability that has the desired bound.

IX. Simulations of the Dynamic Packet Dropping Policy

For simplicity, we simulate a single queue with rate-power curve $C(P,S) = \log(1 + P\alpha_S)$. The channel state α_S is i.i.d. over timeslots and can take two possible values, both equally likely: $\alpha_S \in \{1,2\}$. Every timeslot, a power variable $P(t)$ is allocated subject to the constraint that $0 \leq P(t) \leq 2$ (i.e., $P_{max} = 2$). Thus, we have $\mu_{max} = \log(5)$. The arrival process is i.i.d., where a single packet of size A_{max} arrives with probability 1/2 every timeslot, and else no packet arrives. We use the value $A_{max} = 2\log(2.5)$, so that $\lambda = \log(2.5)$. Further, because $A(t) \in \{0, A_{max}\}$, we have $\hat{A}^2_{max} = \mathbb{E}\{A(t)^2\} = \lambda A_{max}$, and $\sigma^2 = \hat{A}^2_{max} + \max[0, \mu^2_{max} - 2\lambda\mu_{max}] = \lambda A_{max}$. The value of ω is given by (19).

We implement the dynamic packet dropping policy with the original constant value of Q given in Theorem 1. We also simulate the same policy but with a reduced threshold $\tilde{Q} = Q/15$. Finally, we simulate the threshold-adaptive algorithm. We use $\rho = 0.95$ for all simulations (so that the drop rate must be less than 5%). In the threshold-adaptive algorithm, we use $Q_{max} = Q$, $Q_{min} = \max[10\mu_{max}, Q/30]$, $s = (Q_{max} - Q_{min})/100$, and $\theta = 5$. The value of V is varied from 1 to 8000, and all simulations start with an empty system and run over a duration of 15 million timeslots.

The average power expenditure versus V is shown in Fig. 3 (where V is plotted on a logarithmic scale). We see immediately that the average power performance is almost the same for all three algorithms, as all three curves lie almost exactly on top of each other. We note that the constant Q algorithm and the threshold-adaptive algorithm are both analytically guaranteed to meet the edge probability constraint $\alpha \leq \lambda\epsilon/\mu_{max}$, and thus to analytically ensure an acceptance ratio of at least ρ. This fact is consistent with our simulations for both of these algorithms, where the empirical edge probability constraint was satisfied for all values of V, and the empirical average rate of accepted traffic was larger than $\rho\lambda$ for $V < 300$, and differed from $\rho\lambda$ only in the fourth significant digit for $V \geq 300$. The $Q/15$ algorithm does not have analytical guarantees on the edge probability and acceptance ratio, although it was observed to meet both constraints for $V < 400$. However, the edge probability constraint was not always met for $V \geq 400$ (for example, for $V = 500$ the actual edge probability was $\alpha = 0.000173$ while the required bound was $\lambda\epsilon/\mu_{max} = 0.000036$). Correspondingly, it was observed that the rate of accepted traffic was slightly lower than $\rho\lambda = 0.870476$ for $V \geq 400$, although the lowest acceptance rate was 0.869230 at $V = 8000$ (still very close to $\rho\lambda$).

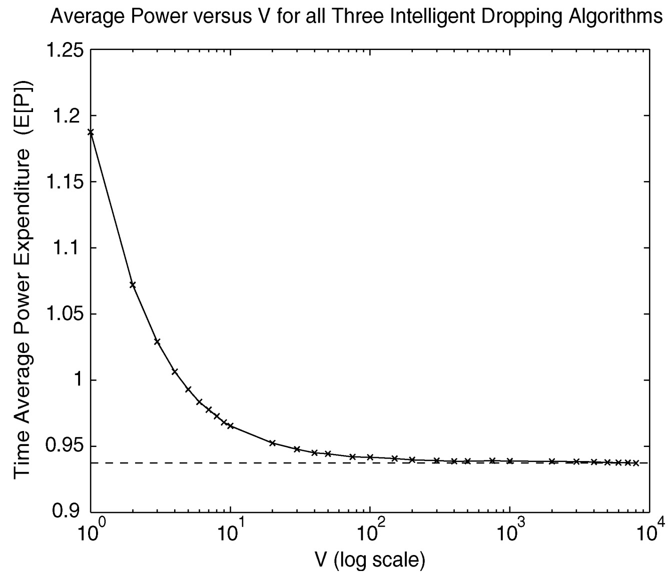


Fig. 3. Simulated performance of average power expenditure versus V for the dynamic packet dropping policy and two variations: The $Q/15$ algorithm and the threshold-adaptive (dynamic Q) algorithm. The three curves lie almost exactly on top of each other and are indistinguishable in the above plot.

The difference between the three algorithms is apparent from the simulated average congestion illustrated in Fig. 4. The original Q algorithm has average backlog that indeed grows only logarithmically in V, although it is still quite large due to our conservative analysis for the constant coefficient of the Q threshold. The average backlog is reduced by a significant constant factor in the adaptive threshold algorithm, without sacrificing energy performance or violating the required acceptance ratio. The average backlog is reduced even further in the $Q/15$ algorithm (roughly by a factor of 15 from the original dynamic dropping policy).

X. Conclusion

This work demonstrates that allowing for a small fraction of packet dropping can fundamentally change the energy-delay tradeoff law for wireless transmitters from the square root law (derived by Berry and Gallager) to a logarithmic law. A dynamic algorithm was constructed to achieve this logarithmic tradeoff using Lyapunov scheduling together with an auxiliary queueing state that regulates edge probabilities. The resulting algorithm does not require knowledge of channel probabilities and can be implemented in real time. The techniques developed here are quite powerful and can likely be used in other contexts to provide delay-aware network control algorithms with low complexity.

Appendix A Proof of Lemma 6

Here we prove Lemma 6 of Section IV-B. Recall that $\boldsymbol{Z}(t) = (U(t), X(t))$. Define $J(X) \triangleq (1/2)X^2$, and note that $\Psi(\boldsymbol{Z}(t)) = L(U(t)) + J(X(t))$. Let $\Delta_L(\boldsymbol{Z}(t))$ and $\Delta_J(\boldsymbol{Z}(t))$

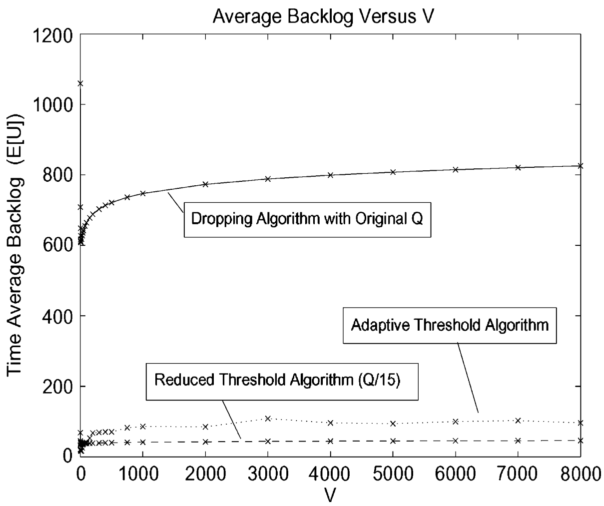


Fig. 4. Simulated performance of average backlog versus V for the original Q algorithm, the $Q/15$ algorithm, and the threshold-adaptive algorithm.

represent the components of the Lyapunov drift associated with $L(U(t))$ and $J(X(t))$, respectively:

$$\Delta_L(\boldsymbol{Z}(t)) \triangleq \mathbb{E}\{L(U(t+1)) - L(U(t))|\boldsymbol{Z}(t)\}$$
$$\Delta_J(\boldsymbol{Z}(t)) \triangleq \mathbb{E}\{J(X(t+1)) - J(X(t))|\boldsymbol{Z}(t)\}$$

The Lyapunov drift of the $\Psi(\boldsymbol{Z}(t))$ function is thus:

$$\Delta(\boldsymbol{Z}(t)) = \Delta_L(\boldsymbol{Z}(t)) + \Delta_J(\boldsymbol{Z}(t))$$

We prove the drift bound in (17) by computing individual bounds on $\Delta_L(\boldsymbol{Z}(t))$ and $\Delta_J(\boldsymbol{Z}(t))$.

Lemma 7: If ω is positive and satisfies (16), then for all t we have:

$$\text{(a) } \Delta_J(\boldsymbol{Z}(t)) \leq \frac{\mu_{max}^2 + (\rho+\epsilon)^2 \hat{A}_{max}^2}{2} - X(t)\left[\mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - (\rho+\epsilon)\lambda\right]$$
$$\text{(b) } \Delta_L(\boldsymbol{Z}(t)) \leq 1 - \omega e^{\omega(Q-U(t))} \times \left[\lambda - \mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - \lambda(1-\rho-\epsilon)/2\right]$$

The proof of part (a) follows by squaring the virtual queue update equation (10) and using a standard quadratic Lyapunov drift computation (see, for example, [14]), and is omitted for brevity. Below we prove part (b).

Proof: (Lemma 7 part (b)): Recall that $L(U) = e^{\omega(Q-U)}$, and note by the finite buffer queueing equation (3) that the $U(t)$ process satisfies:

$$U(t+1) \geq \min\left[Q, U(t) - \mu(t) + A(t)\right]$$

Hence, for any value $\omega > 0$ we have:

$$e^{-\omega U(t+1)} \leq e^{-\omega(U(t)-\mu(t)+A(t))} + e^{-\omega Q}$$

Therefore:

$$e^{\omega(Q-U(t+1))} \leq e^{\omega(Q-U(t))} e^{\omega(\mu(t)-A(t))} + 1 \tag{37}$$

Now note by the Taylor theorem that for any value x that is upper bounded by some maximum value x_{max}, we have:

$$e^x \leq 1 + x + \frac{x^2}{2} e^{x_{max}}$$

Because $\omega(\mu(t) - A(t)) \leq \omega\mu_{max}$ for all t, we have:

$$e^{\omega(\mu(t)-A(t))} \leq 1 + \omega(\mu(t) - A(t)) + \frac{\omega^2}{2}(\mu(t) - A(t))^2 e^{\omega\mu_{max}} \tag{38}$$

Using (38) in (37) yields:

$$e^{\omega(Q-U(t+1))} - e^{\omega(Q-U(t))} \leq 1 - \omega e^{\omega(Q-U(t))} \times \left[A(t) - \mu(t) - \frac{\omega}{2}(\mu(t) - A(t))^2 e^{\omega\mu_{max}}\right]$$

Taking conditional expectations of both sides of the above inequality yields:

$$\begin{aligned}\Delta_L(\boldsymbol{Z}(t)) &\leq 1 - \omega e^{\omega(Q-U(t))} \\ &\quad \times \Big[\lambda - \mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} \\ &\qquad - \frac{\omega}{2} e^{\omega\mu_{max}} \mathbb{E}\left\{(\mu(t) - A(t))^2 |\boldsymbol{Z}(t)\right\}\Big] \\ &\leq 1 - \omega e^{\omega(Q-U(t))} \\ &\quad \times \left[\lambda - \mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - \frac{\omega}{2} e^{\omega\mu_{max}} \sigma^2\right] \quad (39) \\ &\leq 1 - \omega e^{\omega(Q-U(t))} \\ &\quad \times \left[\lambda - \mathbb{E}\{\mu(t)|\boldsymbol{Z}(t)\} - \lambda(1-\rho-\epsilon)/2\right] \quad (40)\end{aligned}$$

where (39) follows from (15), and (40) follows because ω satisfies the inequality $\omega e^{\omega\mu_{max}} \leq \lambda(1-\rho-\epsilon)/\sigma^2$ (given in (16)). This proves part (b) of the lemma. □

Summing the drift components $\Delta_L(\boldsymbol{Z}(t))$ and $\Delta_J(\boldsymbol{Z}(t))$ from Lemma 7 establishes Lemma 6.

Appendix B
Proof of Lemma 3

Here we derive the edge probability bound given in Lemma 3 for the Positive Drift Algorithm. We also compute a simple bound on the constant θ^*. Let $U(t)$ represent the buffer occupancy of the positive drift algorithm, with dynamic equation given by the finite buffer queueing equation (3), and with positive drift given by (8). Define the *inverted process* $Y(t) \triangleq Q - U(t)$.

Lemma 8: $Y(t) \leq \hat{Y}(t)$ for all $t \geq 0$, where $\hat{Y}(t)$ is a process defined with the same initial condition as $Y(t)$, and with update equation:

$$\hat{Y}(t+1) = \max\left[\hat{Y}(t) - A(t) + \mu(t), 0\right]$$

Proof: (Lemma 8): Note that $Y(t)$ has dynamics:

$$Y(t+1) = Y(t) - \tilde{A}(t) + \tilde{\mu}(t) \tag{41}$$

where $\tilde{\mu}(t)$ is the actual data served from queue $U(t)$ (and satisfies $\tilde{\mu}(t) \leq \mu(t)$), and $\tilde{A}(t)$ is the actual data admitted to queue $U(t)$ according to the finite buffer queueing equation (3). Suppose now that $Y(\tau) \leq \hat{Y}(\tau)$ for all $\tau \in \{0, \ldots, t\}$ (note that it holds for $t = 0$ because $Y(0) = \hat{Y}(0)$). We show it also holds for time $\tau = t+1$. If $\tilde{A}(t) \neq A(t)$, then some new data was dropped from the finite buffer queue at time t, and so by (3) we have $U(t+1) = Q$ and hence $Y(t+1) = 0$. Thus, we trivially have $Y(t+1) \leq \hat{Y}(t+1)$ in this case. In the opposite case when $\tilde{A}(t) = A(t)$, we have from (41):

$$\begin{aligned} Y(t+1) &= Y(t) - A(t) + \tilde{\mu}(t) \\ &\leq \max\left[Y(t) - A(t) + \tilde{\mu}(t), 0\right] \\ &\leq \max\left[\hat{Y}(t) - A(t) + \mu(t), 0\right] \\ &= \hat{Y}(t+1) \end{aligned}$$

proving the lemma. ■

The system $\hat{Y}(t)$ evolves like a discrete time $GI/GI/1$ queue with an inverted arrival and transmission rate process, and has *negative drift* given by:

$$\mathbb{E}\left\{-A(t) + \mu(t)\right\} = -\lambda(1 - \rho - \epsilon) \quad (42)$$

Using the *Kingman bound* [23], in steady state we have:

$$Pr[\hat{Y} > Q - \mu_{max}] \leq e^{-\theta^*(Q - \mu_{max})}$$

where the constant θ^* is given by the positive root of the following equation:

$$\mathbb{E}\left\{e^{\theta^*(\mu(t) - A(t))}\right\} = 1$$

It is well known that the above equation has a positive root θ^* whenever the $A(t)$ and $\mu(t)$ processes satisfy the negative drift criterion (42) [23]. The following lemma presents a simple lower bound on θ^* in terms of known constants.

Lemma 9: If the drift condition (42) holds, then θ^* is greater than or equal to any constant θ that satisfies the following inequality:

$$\theta e^{\theta \mu_{max}} \leq 2\lambda(1 - \rho - \epsilon)/\sigma^2 \quad (43)$$

where $\mathbb{E}\{A(t)\} = \lambda$, $\mu(t) \leq \mu_{max}$ for all t, and where σ^2 is any constant that satisfies (15).

As in the bound for the constant ω in Lemma 6, it is not difficult to show that a particular solution of (43) is given by:

$$\theta \triangleq \frac{2\lambda(1 - \rho - \epsilon)}{\sigma^2} e^{-2\lambda\mu_{max}(1-\rho-\epsilon)/\sigma^2}$$

and hence the above value can be used in replacement of θ^* in (9) to yield an acceptable Q value for use in the positive drift algorithm.

Proof: (Lemma 9): Define $\delta(t) \triangleq \mu(t) - A(t)$, and define $f(\theta) \triangleq \mathbb{E}\{e^{\theta\delta(t)}\}$. Because the negative drift expression (42) holds, it is well known that (see, for example, [23]):

- $f(\theta) < 1$ for all θ such that $0 < \theta < \theta^*$
- $f(0) = f(\theta^*) = 1$
- $f(\theta) > 1$ whenever $\theta > \theta^*$

For any $\theta \geq 0$, a Taylor expansion of $e^{\theta\delta(t)}$ yields:

$$e^{\theta\delta(t)} \leq 1 + \theta\delta(t) + \frac{\theta^2\delta(t)^2}{2} e^{\theta\mu_{max}}$$

Taking expectations of both sides and using (42) yields:

$$f(\theta) \leq 1 - \theta\lambda(1 - \rho - \epsilon) + \frac{\theta^2\sigma^2}{2} e^{\theta\mu_{max}}$$

Define $g(\theta)$ as the right hand side in the above inequality. It follows that $f(\theta) \leq g(\theta)$ whenever $\theta \geq 0$. Therefore, any nonnegative constant θ that satisfies $g(\theta) \leq 1$ must also satisfy $f(\theta) \leq 1$, and hence $\theta \leq \theta^*$. But the condition $g(\theta) \leq 1$ is equivalent to the condition (43), proving the lemma. □

Finally, combining the bound on the tail behavior of $\hat{Y}(t)$ and the fact that $Y(t) \leq \hat{Y}(t)$, we have in steady state:

$$\begin{aligned} Pr[U < \mu_{max}] &= Pr[Y > Q - \mu_{max}] \\ &\leq Pr[\hat{Y} > Q - \mu_{max}] \\ &\leq e^{-\theta^*(Q - \mu_{max})} \end{aligned}$$

which proves Lemma 3.

References

[1] M. J. Neely, "Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks," in *Proc. WiOpt*, Boston, MA, Apr. 2006, pp. 179–188.

[2] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inform. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[3] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 487–499, Aug. 2002.

[4] M. Zafer and E. Modiano, "A calculus approach to minimum energy transmission policies with quality of service guarantees," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 548–559.

[5] M. A. Khojastepour and A. Sabharwal, "Delay-constrained scheduling: Power efficiency, filter design, and bounds," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1938–1949.

[6] W. Chen, M. J. Neely, and U. Mitra, "Energy-efficient transmissions with individual packet delay constraints," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2090–2109, May 2008.

[7] A. Fu, E. Modiano, and J. Tsitsiklis, "Optimal energy allocation for delay-constrained data transmission over a time-varying channel," in *Proc. IEEE INFOCOM*, 2003, pp. 1095–1105.

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

[9] M. J. Neely, "Optimal energy and delay tradeoffs for multi-user wireless downlinks," *IEEE Trans. Inform. Theory*, vol. 53, no. 9, pp. 3095–3113, Sep. 2007.

[10] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE J. Selected Areas Commun.*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.

[11] M. J. Neely, "Dynamic Power Allocation and Routing for Satellite and Wireless Networks With Time Varying Channels," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 2003.

[12] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 1723–1734.

[13] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[14] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Networking*, vol. 1, no. 1, pp. 1–149, 2006.

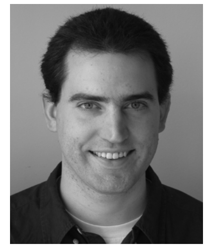
[15] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Control*, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[16] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.

[17] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Automat. Control*, vol. 40, no. 2, pp. 251–260, Feb. 1995.

[18] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in *Proc. IEEE INFOCOM*, Mar. 1996, pp. 296–302.

[19] N. Kahale and P. E. Wright, "Dynamic global packet routing in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 1997, pp. 1414–1421.
[20] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, 2001.
[21] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in *Proc. IEEE INFOCOM*, 2001, pp. 1095–1103.
[22] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Selected Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[23] R. Gallager, *Discrete Stochastic Processes*. Boston, MA: Kluwer Academic Publishers, 1996.
[24] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for dynamic multiserver wireless systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1506–1515, Jun. 2006.



Michael J. Neely (S'00–M'03–SM'08) received B.S. degrees in both electrical engineering and mathematics from the University of Maryland, College Park, in 1997 and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1999 and 2003, respectively.

In 2004, he joined the faculty of the Electrical Engineering Department, University of Southern California, Los Angeles, where he is currently an Assistant Professor. His research is in the area of stochastic network optimization for wireless and *ad hoc* mobile networks.

Dr. Neely received the NSF Career Award in 2008 and a Department of Defense NDSEG Fellowship. He is a member of Tau Beta Pi and Phi Beta Kappa.