# Optimal Energy and Delay Tradeoffs for Multi-User Wireless Downlinks

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

***Abstract*— We consider the fundamental delay tradeoffs for minimizing energy expenditure in a multi-user wireless downlink with randomly varying channels. First, we extend the Berry-Gallager bound to a multi-user context, demonstrating that any algorithm that yields average power within $O(1/V)$ of the minimum power required for network stability must also have an average queueing delay greater than or equal to $\Omega(\sqrt{V})$. We then develop a class of algorithms, parameterized by $V$, that come within a logarithmic factor of achieving this fundamental tradeoff. The algorithms overcome an exponential state space explosion, and can be implemented in real time without a-priori knowledge of traffic rates or channel statistics. Further, we discover a "super-fast" scheduling mode that beats the Berry-Gallager bound in the exceptional case when power functions are piecewise linear.**

***Index Terms*— queueing analysis, stability, optimization, stochastic control, asymptotic tradeoffs**

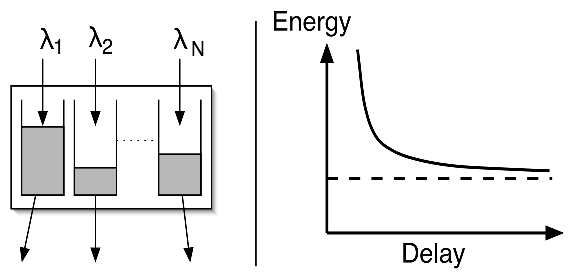


Fig. 1. A wireless downlink with multiple input streams, with an associated energy-delay tradeoff curve.

## I. Introduction

In this paper we consider the fundamental tradeoff between energy and delay in a multi-user wireless network. We focus on the case of a wireless downlink that transmits to $N$ different users over $N$ time varying channels (Fig. 1). Transmission rates depend on current channel conditions and current power allocation decisions. We assume that time is slotted and that channel conditions can be measured every timeslot. The goal is to allocate power in reaction to current channel states and current queue backlogs to stabilize the system while minimizing energy expenditure and maintaining low delay. This objective is important for satellite and wireless downlinks, as well as for wireless nodes that transmit to neighbors within a larger ad-hoc network. It is crucial to understand the fundamental performance limits of such systems, as these limits must be pushed to their maximum to support the demands that will be placed on future networks.

The fundamental energy-delay tradeoff was characterized by Berry and Gallager in [2] for the special case of a single queue that stores data for transmission over a single fading channel. Average energy for such a system can be improved by accumulating data for more efficient future transmission, at the expense of increasing queueing congestion and delay. It was shown in [2] that, subject to strict convexity assumptions on the rate-power curve of the system, any set of algorithms that yield average power within $O(1/V)$ of the minimum power required for stability (for increasingly large positive numbers $V$), must also have average queueing delay greater than or equal to $\Omega(\sqrt{V})$.[1] This demonstrates that any effort to reduce the power expended to support incoming traffic will necessarily increase delay. Further, an algorithm for achieving this fundamental tradeoff was proposed, based on the concept of buffer partitioning.

Related work on minimizing energy in a static wireless node with known arrival times and a single transmitter is considered in [3] [4], and a problem with a static link but stochastic arrivals is treated using filter theory in [5]. Similar problems of minimizing energy subject to delay constraints or minimizing delay subject to energy constraints are treated for single link satellite and wireless systems in [4] [6] [7] using stochastic differential equations, dynamic programming, and Markov decision theory. While such techniques might improve the delay coefficient in comparison to the Berry-Gallager algorithm, they cannot overcome the fundamental $\Omega(\sqrt{V})$ tradeoff curve.

However, there has been little work to extend this theory of energy-delay tradeoffs to multi-user networks. This is largely due to the complexity explosion associated with increasing the number of queues beyond one. Indeed, the number of backlog vectors and channel state vectors both increase exponentially with the number of queues in the system, making dynamic programming approaches and Markov decision theory approaches prohibitive. Indeed, even for the case of a single data link, it is difficult to implement the proposed stochastic algorithms ([2] [4] [6] [7]), as these algorithms require scheduling policies to be pre-computed based on full knowledge of the input rate and the steady state channel probabilities. While it may be

Michael J. Neely is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089. (website: http://www-rcf.usc.edu/∼mjneely).

This work was presented in part at the IEEE INFOCOM conference, Barcelona, April 2006. This material is based on work supported in part by one or more of the following: the National Science Foundation under grant OCE 0520324, the DARPA IT-MANET program.

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

possible to estimate these statistics in the single link case when the number of channel states is relatively small, it is not practical to envision estimating the exponentially growing number of parameters in the multi-user case, nor is it practical to envision solving the corresponding optimization problems even if all of these parameters were known exactly.

The complexity explosion problem is one of the major obstacles that we overcome in this paper. We do so by combining the concept of buffer partitioning with the recently developed theory of *performance optimal Lyapunov scheduling* [8]-[11]. Specifically, [8]-[11] develops Lyapunov techniques for treating stability and performance optimization simultaneously (extending the Lyapunov stability results developed, for example, in [14]-[19]). These results are used in [10] to develop simple stabilizing algorithms for general multi-hop wireless networks that yield average power within $O(1/V)$ of the minimum power required for stability (for any control parameter $V > 0$). However, the resulting end-to-end delay of these algorithms was shown to grow linearly with $V$, even for the special case of a single data link. Thus, the algorithms do not yield the optimal delay tradeoff in the single link case, suggesting that improved tradeoffs might be possible.

In this paper, we extend the theory of stochastic optimal networking to treat optimal delay tradeoffs. First, we extend the Berry-Gallager bound to multi-user systems, establishing that the $\Omega(\sqrt{V})$ tradeoff curve also applies in these more general scenarios. Next, we construct a simple algorithm that achieves the fundamental energy-delay tradeoff to within a logarithmic factor. This is the first algorithm to achieve such performance for multi-user systems. Furthermore, the algorithm does not require knowledge of input rates or channel statistics, and is simple to implement in real time for systems with any number of users. The technique introduces a novel method of "drift steering" that maintains a set of virtual queues with Lyapunov coefficients that turn "ON" or "OFF" based on congestion thresholds. For simplicity of exposition, we focus on the case of a single downlink with $N$ channels, although the technique readily extends to the more general scenario of multi-node, multi-hop networks using the backpressure techniques developed in [14], [8]-[11], [19].

Our analysis further reveals an important exceptional case where "super-fast" convergence is possible. In particular, if power curves have certain piecewise linear properties, such as when power allocation is restricted to using either zero power or full power at any server, then our algorithm can be modified to achieve similar energy performance with delay that grows only logarithmically in the $V$ parameter. This demonstrates that it is possible to outperform the Berry-Gallager bound in systems with piecewise linearities.

Previous work in the area of capacity and stable scheduling for multi-user networks is found in [12]-[19], and energy efficiency is considered in [20]-[24], [10]. Most work in network optimization is closely tied to static optimization theory and convex duality, including [25]-[29] for static networks and [20], [30]-[32] for stochastic gradient algorithms and fluid limit models. A Lyapunov method for performance optimization is developed in [8] [9] [10] which yields strategies similar to those suggested by gradient optimization approaches, and also yields explicit performance and delay bounds (see also [11]). This paper builds on the Lyapunov method to achieve optimal delay tradeoffs, and makes a significant contribution to the field by developing new scheduling algorithms that go beyond the classical gradient methods of optimization theory.

An outline of this paper is as follows. In the next section, we define the system model and prove the $\Omega(\sqrt{V})$ lower bound on delay for any algorithm implemented on systems with strict convexity properties. In Section IV we review the main features of the original Berry-Gallager algorithm from [2] and outline the complexity challenges associated with the multi-dimensional problem. In Section V we introduce Lyapunov drift theory and present the control algorithm. Performance analysis and extensions to super-fast scheduling are treated in Sections VI and VII. Simulations are presented in Section VIII.

## II. Problem Formulation

Consider a wireless downlink with $N$ time-varying channels, each for a different wireless user. The system operates in slotted time with slots normalized to one unit. We let $\vec{S}(t) = (S_1(t), \ldots, S_N(t))$ represent the vector of current channel states for each link during slot $t$, for $t \in \{0, 1, 2, \ldots\}$. These states can represent current fading levels, attenuation, and/or noise levels associated with the channel during slot $t$. It is assumed that channels hold their states for the duration of a timeslot, but potentially change on slot boundaries. For simplicity, we assume there are a finite number of channel states $\vec{S}$, and that the channel process is i.i.d. from one timeslot to the next.[2] We define $\pi_{\vec{S}}$ as the occurrence probability for each channel state $\vec{S}$. These channel probabilities determine the capacity region of the network [19], but are not necessarily known to the downlink controller.

Every timeslot, the controller observes the current channel states and chooses transmission rates by allocating power as a vector $\vec{P}(t) = (P_1(t), \ldots, P_N(t))$ subject to an instantaneous power constraint $\vec{P}(t) \in \Pi$, where $P_i(t)$ represents the power allocated to link $i$ during slot $t$, and $\Pi$ is a compact set that specifies the collection of admissible power vectors. For example, a system with only a peak power constraint $P_{peak}$ can be modeled with a power set $\Pi$ consisting of all non-negative power vectors $\vec{P}$ that satisfy $\sum_{i=1}^{N} P_i \leq P_{peak}$. Additional constraints on instantaneous power transmission can be incorporated simply by modifying the power set $\Pi$. Transmission rates for each link are determined by the current channel state vector $\vec{S}(t)$ and the current power allocation vector $\vec{P}(t)$ according to a general rate-power curve $\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(\vec{P}, \vec{S}), \ldots, \mu_N(\vec{P}, \vec{S}))$. We assume that $\vec{\mu}(\vec{P}, \vec{S})$ is a continuous function of the power vector for each channel state vector $\vec{S}$. Let $\mu_{max}$ represent a bound on the maximum transmission rate of a single link, maximized over all channel states $\vec{S}$ and all power vectors $\vec{P} \in \Pi$.

Data arrives in packetized form according to $N$ random processes, and each packet is stored in one of $N$ internal queues according to its destination (see Fig. 1). We let $\vec{A}(t) = (A_1(t), \ldots, A_N(t))$ represent the vector of new packet arrivals

[2] Extensions to non-i.i.d. systems can be treated via techniques in [8] [19].

every slot, where $A_i(t)$ is the number of bits that arrive for user $i$ during slot $t$. We assume arrival vectors $\vec{A}(t)$ are i.i.d. over slots, and define the rate vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)$, where $\lambda_i = \mathbb{E}\{A_i(t)\}$ represents the arrival rate to queue $i$ in units of bits/slot. For simplicity, we assume that $\lambda_i > 0$ for all $i$. Let $U_i(t)$ denote the *unfinished work* in queue $i$ at slot $t$, representing the backlog of bits waiting to be transmitted over channel $i$. Let $\vec{r}(t) = \vec{\mu}(\vec{P}(t), \vec{S}(t))$ represent the transmission rate vector at slot $t$. Queue dynamics thus proceed according to the equation:

$$U_i(t+1) = \max[U_i(t) - r_i(t), 0] + A_i(t) \tag{1}$$

where $r_i(t) = \mu_i(\vec{P}(t), \vec{S}(t))$, and $\vec{P}(t) \in \Pi$. The goal of the network controller is to allocate power subject to the power constraints so that all queues are stabilized and, ideally, average power is minimized. It turns out that minimum power cannot be achieved without infinite average delay, and hence our precise objective is to stabilize the system with average power that can be pushed arbitrarily close to the minimum power required for stability, with an optimal delay tradeoff.

Note that the above formulation for a wireless downlink is quite general and can equally model single-hop networks with multiple nodes and $N$ data links. The only difference in the network case is that queues and transmitters are distributed over the different nodes of the network, and so extra coordination might be required for control decisions.

### A. Example Rate-Power Functions

In the special case where there is only a peak power constraint and links are independent with no inter-channel interference, rate-power functions have the form:

$$\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(P_1, S_1), \ldots, \mu_N(P_N, S_N)) \tag{2}$$

where power is allocated so that $\sum_i P_i(t) \leq P_{peak}$. In cases when there is inter-channel interference, transmission rates on each link depend on the full vector of channel states and power allocations. For example, under a signal-to-interference ratio model, the rate functions are given by:

$$\mu_i(\vec{P}, \vec{S}) = f_i(SINR_i(\vec{P}, \vec{S})) \tag{3}$$

where $SINR_i(\vec{P}, \vec{S})$ is the signal-to-interference-plus-noise ratio on link $i$ when power vector $\vec{P}$ is allocated under channel state $\vec{S}$, and $f_i(\cdot)$ is any function of $SINR$. The power set $\Pi$ might specify further system constraints, such as allowing transmissions over at most one channel during any slot and/or restricting allocations to either full power or zero power.

### B. The Minimum Energy Function

Here we describe the minimum power required to stabilize all queues of the downlink system described above. In [19], the *capacity region* $\Lambda$ is defined as the closure of the set of all input rate vectors $\vec{\lambda}$ stabilizable under some power allocation algorithm that conforms to the power constraint $\vec{P}(t) \in \Pi$. Throughout this paper, we assume that the input rate vector is strictly interior to the capacity region $\Lambda$, and so the system is stabilizable. In [10], the minimum average power required for stability is shown to be the solution to an optimization problem associated with a stationary randomized power allocation strategy. Below we present a generalized statement that considers minimizing the time average of a *power cost function* $h(\vec{P}(t))$. We assume that $h(\vec{P})$ is non-negative and continuous in the power vector $\vec{P}$. We define the *average power cost* as follows:

$$h_{av} \triangleq \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{h(\vec{P}(\tau))\right\}$$

The minimum average power problem corresponds to a cost function:

$$h(\vec{P}) = \textstyle\sum_{i=1}^N P_i$$

Alternate cost metrics, such as second moments of power, can be modeled as desired by choosing different $h(\vec{P})$ functions.

*Theorem 1:* If the functions $h(\vec{P})$ and $\vec{\mu}(\vec{P}, \vec{S})$ are continuous in the power vector $\vec{P}$, then the minimum average power cost $h^*_{av}$ is given by the solution to the following optimization problem (defined in terms of auxiliary probabilities $\gamma_k^{\vec{S}}$ and power vectors $\vec{P}_k^{\vec{S}}$ for all $\vec{S}$ and for $k \in \{1, \ldots, N+2\}$):[3]

$$\begin{aligned} \text{Minimize:} \quad & h_{av} = \textstyle\sum_{\vec{S}} \pi_{\vec{S}} \sum_{k=1}^{N+2} \gamma_k^{\vec{S}} h(\vec{P}_k^{\vec{S}}) \\ \text{Subject to:} \quad & \vec{\mu}_{av} \triangleq \textstyle\sum_{\vec{S}} \pi_{\vec{S}} \sum_{k=1}^{N+2} \gamma_k^{\vec{S}} \vec{\mu}(\vec{P}_k^{\vec{S}}, \vec{S}) \geq \vec{\lambda} \\ & \vec{P}_k^{\vec{S}} \in \Pi, \gamma_k^{\vec{S}} \geq 0 \quad \text{for all } k, \vec{S} \\ & \textstyle\sum_{k=1}^{N+2} \gamma_k^{\vec{S}} = 1 \quad \text{for all } \vec{S} \end{aligned} \tag{4}$$

Thus, the minimum power cost for stability is achieved among the class of stationary policies that measure the current channel state $\vec{S}(t)$ and then randomly allocate a power vector $\vec{P}_k^{\vec{S}}$ with probability $\gamma_k^{\vec{S}}$. This result is proven in [10] for the case when $h(\vec{P}) = \sum_i P_i$ by showing that no stabilizing algorithm can yield average power lower than $h^*_{av}$, but that stabilizing algorithms can be constructed with average power that is arbitrarily close to $h^*_{av}$. The fact that a probabilistic combination of $N+2$ power vectors is used for every channel state $\vec{S}$ follows by extending the dimensionality of the system from $N$ to $N+1$ (due to the single minimum power objective) and using Caratheodory's theorem, as described in [10]. The proof for general $h(\vec{P})$ functions is similar to [10] and is omitted for brevity.

Note that the minimum power cost required for stability depends on the traffic statistics only through the input rate vector $\vec{\lambda}$. We thus define the *minimum energy function* $\Phi(\vec{\lambda})$ as the minimum power cost required to stabilize the input rate vector $\vec{\lambda}$, considering all conceivable algorithms. That is, $\Phi(\vec{\lambda}) = h^*_{av}$, where $h^*_{av}$ is the solution of the above optimization problem. An equivalent but more compact way to define $\Phi(\vec{\lambda})$ is as follows.

*Definition 1:* The minimum energy function $\Phi(\vec{\lambda})$ is defined as the value $h^*_{av}$ that achieves the minimum $h_{av}$ value over the class of all feasible stationary randomized power

[3]The same holds more generally for rate-power curves $\mu(\vec{P}, \vec{S})$ that are *upper semi-continuous* in the power vector, and cost functions $h(\vec{P})$ that are *lower semi-continuous*.

allocation algorithms that satisfy:

$$h_{av} = \mathbb{E}\left\{h(\vec{P}(t))\right\}$$
$$\mathbb{E}\left\{\vec{\mu}(\vec{P}(t), \vec{S}(t))\right\} \geq \vec{\lambda}$$

It is not difficult to show that any stationary randomized algorithm that satisfies the above can be modified to yield average transmission rates that are exactly equal to the input rate vector $\vec{\lambda}$.[4] Thus, for any vector $\vec{\lambda}$ inside the capacity region $\Lambda$, there is a particular power allocation policy with power allocations $\vec{P}^*(t) = (P_1^*(t), \ldots, P_N^*(t))$ and transmission rates $\vec{r}^*(t) = (r_1^*(t), \ldots, r_N^*(t))$ that are independent of queue backlogs, and that yield:

$$\mathbb{E}\left\{h(\vec{P}^*(t))\right\} = \Phi(\vec{\lambda}) \quad (5)$$
$$\mathbb{E}\left\{\vec{r}^*(t)\right\} = \vec{\lambda} \quad (6)$$

where the expectation is with respect to the random channel state vector and the randomized control decisions. While such a policy exists for all vectors $\vec{\lambda}$, it would be difficult to construct such a policy, as that would involve solving the optimization problem (4) and would require full knowledge of the rate vector $\vec{\lambda}$ and the channel probabilities $\pi_{\vec{S}}$.

The minimum energy function $\Phi(\vec{\lambda})$ has an important convexity property for all vectors $\vec{\lambda}$ in the capacity region $\Lambda$, as described by the following lemma.

*Lemma 1:* The function $\Phi(\vec{\lambda})$ is convex over $\vec{\lambda} \in \Lambda$, and is non-decreasing in each entry $\lambda_i$ ($i \in \{1, \ldots, N\}$).

*Proof:* Omitted for brevity. □

It is known that any convex multi-variable function is twice differentiable almost everywhere [33]. Hence, throughout this paper we shall assume that the $\Phi(\vec{\lambda})$ function is twice differentiable at the point $\vec{\lambda}$ of interest.

## III. The Fundamental Energy-Delay Bound

Here we extend the Berry-Gallager bound to multi-user systems. Specifically, we show that under a strict convexity assumption on the minimum energy function $\Phi(\vec{\lambda})$, any sequence of policies (parameterized by increasing positive numbers $V$) that yield average power cost $h_{av}$ within $O(1/V)$ of the minimum average cost required for stability must also have average delay greater than or equal to $\Omega(\sqrt{V})$. Our proof closely follows the work in [2] for the single user case, and in particular in this section we shall restrict attention to the same class of *admissible strategies*:

*Definition 2:* A sequence of scheduling strategies, parameterized by increasing positive numbers $V$ that tend to infinity, is *admissible* if

1) The policies make *stationary* (and possibly randomized) power allocation decisions based on the current queue backlog and channel state vectors $\vec{U}(t)$ and $\vec{S}(t)$.
2) For each $V$, the corresponding policy stabilizes the system and forms an ergodic Markov chain with steady state queue backlog distribution $\pi(\vec{U})$. Furthermore, the steady state average backlog $\mathbb{E}\left\{\sum_{i=1}^N U_i\right\}$ is finite for all $V$ and increases to infinity as $V \to \infty$.
3) There exist positive values $\theta_1$, $\theta_2$ such that for all timeslots $t$ and for each $i \in \{1, \ldots, N\}$, we have:

$$Pr\left[A_i(t) - \mu_i(\vec{P}(t), \vec{S}(t)) \geq \theta_2 \mid \vec{U}(t)\right] \geq \theta_1$$

The third assumption above states that there is a positive probability that the backlog of any particular queue increases by at least $\theta_2$ during a single timeslot, regardless of the current backlog value. This assumption holds whenever there is a non-zero probability of an outage on channel $i$ during a particular timeslot (that is, having a channel state that yields zero data rate on channel $i$ for every power allocation), or whenever there is a non-zero probability that $A_i(t) > \mu_{max}$. The first two assumptions on stationarity and ergodicity simplify the proof of the fundamental bound at the expense of slightly reducing the class of scheduling policies considered. However, the assumptions themselves are not very restrictive, in that algorithms that yield optimal energy-delay tradeoffs can be formulated according to Markov decision theory, which leads to stationary policies that base decisions on the system state vectors $\vec{U}(t)$ and $\vec{S}(t)$.

*Theorem 2:* (Multi-User Berry-Gallager Bound) If the input rate vector $\vec{\lambda}$ is strictly interior to the capacity region $\Lambda$ and if the minimum cost function $\Phi(\vec{\lambda})$ has a positive definite matrix of partial derivatives $\nabla^2\Phi(\vec{\lambda})$ at the point $\vec{\lambda}$, then any sequence of admissible policies that yield average power cost within $O(1/V)$ of $\Phi(\vec{\lambda})$ must have average delay greater than or equal to $\Omega(\sqrt{V})$.

*Proof:* See Appendix A. □

The positive definite assumption in the above theorem ensures that the second order terms of the Taylor series expansion of $\Phi(\vec{\lambda} + \vec{\epsilon})$ are non-zero about the point $\vec{\lambda}$. This condition holds for a wide class of systems with nonlinear rate-power curves, and is related to the notion of *strict convexity*. It is not difficult to show that $\Phi(\vec{\lambda})$ is strictly convex whenever the feasible power set $\Pi$ includes only a peak power constraint and when $h(\vec{P}) = \sum_i P_i$ and rate-power curves have the form specified in (2) with functions $\mu_i(P_i, S_i)$ that are strictly concave in $P_i$ for each link $i \in \{1, \ldots, N\}$. However, in systems with only a discrete set of power allocation options (such as systems that can allocate either zero power or full power), the corresponding $\Phi(\vec{\lambda})$ function is piecewise linear and hence the energy-delay properties of such systems are not necessarily governed by the Berry-Gallager bound.

## IV. Buffer Partitioning

To introduce the concept of buffer partitioning, we first review the basic Berry-Gallager threshold algorithm developed in [2] for the case of a single queue with a single time varying data link (i.e., the case $N = 1$), and for the minimum average power objective with cost function $h(P) = P$. For such a system, let $U(t)$ represent the current unfinished work in the queue, let $A(t)$ represent the arrivals on slot $t$, and let $\lambda = \mathbb{E}\{A(t)\}$ represent the arrival rate. Define $\Phi(\gamma)$ as the minimum average power required to support an average transmission rate of $\gamma$. The Berry-Gallager algorithm switches

[4] Strictly speaking, one can modify the constraint on decision variable $r_i(t)$ in (1) so that $0 \leq r_i(t) \leq \mu_i(\vec{P}(t), \vec{S}(t))$, and then formally consider policies that sometimes set $r_i(t) = 0$ to ensure the equality of (6).

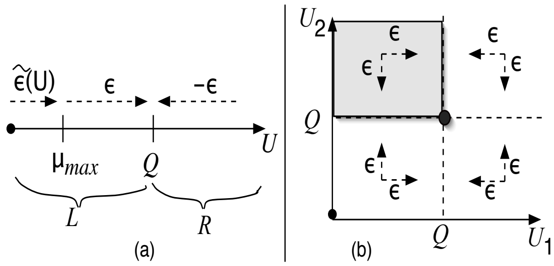


Fig. 2. (a) A partitioning of the single dimensional state space into Left and Right regions for a single link system, with drift directions illustrated. (b) An example partitioning of a 2-user system with the multi-dimensional drift vectors shown. The backlog vector is currently in the upper left region.

between two different transmission policies depending on a buffer occupancy threshold. Specifically, the buffer is partitioned into two halves according to a positive threshold $Q$ (where $Q > \mu_{max}$). When $U(t) < Q$, power is allocated using the stationary policy that yields an average transmission rate of $\lambda - \epsilon$ and an average power expenditure of $\Phi(\lambda-\epsilon)$ (for some specified value of $\epsilon$ such that $0 < \epsilon < \lambda$). When $U(t) \geq Q$, power is allocated according to the stationary policy that yields an average transmission rate of $\lambda + \epsilon$ and an average power expenditure of $\Phi(\lambda+\epsilon)$.

Thus, for this algorithm the queue backlog $U(t)$ only affects power allocation by determining which of the two transmission policies is used. Once a particular policy is determined, power is allocated based only on the current channel state $S(t)$. Now define $\tilde{r}(t)$ as the amount of bits transmitted by the queue on timeslot $t$, and define the *drift* as the expected difference between $A(t)$ and $\tilde{r}(t)$ given the current backlog level $U(t)$. Note that the drift is equal to $-\epsilon$ when $U(t) \geq Q$, and is equal to $\epsilon$ when $U(t) < Q$, with the exceptional case when the queue is near empty and the drift is equal to $\tilde{\epsilon}(U) \geq \epsilon$ (see Fig. 2a). This is an edge effect that can only occur on the interval $0 \leq U < \mu_{max}$, where it is possible that $\tilde{r}(t)$ may be less than the scheduled transmission rate $r(t) = \mu(P(t), S(t))$ due to little or no data being present in the queue. The Berry-Gallager algorithm is designed to achieve the optimal energy-delay tradeoff curve for the single link case (through suitable choices of the $Q$ and $\epsilon$ parameters). This performance is due to the bi-modal transmission policy, and cannot be achieved by policies that have uni-modal drift properties [2].

### A. Multi-Dimensional Buffer Partitioning

Here we extend the buffer partitioning concept to a multi-user context. We consider the minimum energy function $\Phi(\vec{\lambda})$ that corresponds to any continuous and non-negative power cost metric $h(\vec{P})$. The algorithm we develop in this subsection is not meant to be a practical control strategy. Rather, it is intended to highlight the challenges and design principles associated with the multi-user problem. A practical control strategy is developed in the next section based on these principles.

Recall that all backlog vectors $\vec{U}$ take values within the $N$ dimensional state space $[0,\infty)^N$. Consider partitioning this state space into $2^N$ regions according to the buffer threshold parameter $Q$. Specifically, we define $\vec{H}(\vec{U})$ as a vector with binary entries, where the $i$th entry depends on whether $U_i$ is to the right or left of the $Q$ threshold:

$$H_i(U_i) \triangleq \begin{cases} \text{“}L\text{”} & \text{if } U_i < Q \\ \text{“}R\text{”} & \text{if } U_i \geq Q \end{cases}$$
$$\vec{H}(\vec{U}) \triangleq (H_1(U_1), \ldots, H_N(U_N))$$

In this way, the vector $\vec{H}(\vec{U}(t))$ indicates which of the $2^N$ regions currently contains the backlog vector $\vec{U}(t)$ (see Fig. 2b). Consider now an algorithm that switches between $2^N$ different transmission policies depending on the current region of the backlog vector. The drift of each policy is designed to push each component of the backlog vector closer to the $Q$ threshold. Specifically, we consider a drift parameter $\epsilon$ such that $0 < \epsilon < \min_{i\in\{1,\ldots,N\}}\{\lambda_i\}$, and define the vector $\vec{\epsilon}(\vec{H})$ as follows:

$$\epsilon_i(H_i) \triangleq \begin{cases} -\epsilon & \text{if } H_i = \text{“L”} \\ +\epsilon & \text{if } H_i = \text{“R”} \end{cases}$$
$$\vec{\epsilon}(\vec{H}) \triangleq (\epsilon_1(H_1), \ldots, \epsilon_N(H_N))$$

Assume that $\vec{\lambda} + \vec{\epsilon}(\vec{H}) \in \Lambda$ for all $\vec{H}$. Whenever $\vec{H}(\vec{U}(t)) = \vec{H}$, the algorithm uses the stationary policy that allocates power based only on the current channel state $\vec{S}(t)$ to yield an expected transmission rate of $\vec{\lambda} + \vec{\epsilon}(\vec{H})$ with an average power expenditure of $\Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H}))$. Such a policy is guaranteed to exist by (5) and (6). Specifically, if $\vec{P}^*(t)$ and $\vec{r}^*(t)$ represent the actual power vectors and transmission vectors used by the policy on a given timeslot $t$, we have (compare with (5), (6)):

$$\mathbb{E}\left\{h(\vec{P}^*(t)) \mid \vec{H}(\vec{U}(t)) = \vec{H}\right\} = \Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H})) \quad (7)$$
$$\mathbb{E}\left\{\vec{r}^*(t) \mid \vec{H}(\vec{U}(t)) = \vec{H}\right\} = \vec{\lambda} + \vec{\epsilon}(\vec{H}) \quad (8)$$

Note that this algorithm requires the pre-computation of $2^N$ different transmission policies, one for each region $\vec{H}$. Each individual policy is computed by solving the optimization problem (4) for the corresponding rate vector $\vec{\lambda} + \vec{\epsilon}(\vec{H})$, and each such optimization requires a-priori knowledge of the exponential number of channel state probabilities $\pi_{\vec{S}}$. Thus, such a policy cannot be practically implemented in a real network. However, it is useful to consider the performance of this strategy as an aid to analyzing the performance of the more practical algorithm that we develop in the next section.

Consider an implementation of the policy and let $\alpha_{\vec{H}}(t)$ represent the probability that the backlog vector is within the region $\vec{H}$ at a given timeslot $t$. The expected transmission rate on link $i$ during slot $t$ is thus:

$$\begin{aligned} \mathbb{E}\{r_i^*(t)\} &= \sum_{\vec{H}} \alpha_{\vec{H}}(t)\mathbb{E}\left\{r_i^*(t) \,\middle|\, \vec{H}(\vec{U}(t)) = \vec{H}\right\} \\ &= \sum_{\vec{H}} \alpha_{\vec{H}}(t)(\lambda_i + \epsilon_i(H_i)) \\ &= \lambda_i + \sum_{\vec{H}} \alpha_{\vec{H}}(t)\epsilon_i(H_i) \quad (9) \end{aligned}$$

Note that:

$$\sum_{\vec{H}} \alpha_{\vec{H}}(t)\epsilon_i(H_i) = \epsilon\left[\sum_{\{\vec{H}\,|\,\epsilon_i(H_i)=\epsilon\}}\alpha_{\vec{H}}(t) - \sum_{\{\vec{H}\,|\,\epsilon_i(H_i)=-\epsilon\}}\alpha_{\vec{H}}(t)\right] = \epsilon\left[\alpha_i^R(t) - \alpha_i^L(t)\right] \quad (10)$$

where we define $\alpha_i^R(t)$ and $\alpha_i^L(t)$ as the probability that $U_i(t)$ is to the right and to the left of the $Q$ threshold, respectively. Specifically:

$$\alpha_i^R(t) \triangleq Pr[U_i(t) \geq Q] \ , \ \alpha_i^L(t) \triangleq Pr[U_i(t) < Q]$$

Using (10) in (9), we have for all $i \in \{1, \ldots, N\}$:

$$\mathbb{E}\left\{r_i^*(t)\right\} = \lambda_i + \epsilon\left[\alpha_i^R(t) - \alpha_i^L(t)\right] \quad (11)$$

Similarly, we can use (7) to compute the expected power expenditure on slot $t$:

$$\mathbb{E}\left\{h(\vec{P}^*(t))\right\} = \sum_{\vec{H}} \alpha_{\vec{H}}(t)\Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H})) \quad (12)$$

Because $\Phi(\vec{\lambda})$ is convex and twice differentiable at $\vec{\lambda}$, it follows by the multi-dimensional Taylor theorem that:

$$\Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H})) \leq \Phi(\vec{\lambda}) + \sum_{i=1}^{N} \frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon_i(H_i) + N\epsilon^2\beta \quad (13)$$

for a fixed value $\beta > 0$. In cases when $\Phi(\vec{\lambda})$ is twice differentiable about the neighborhood $(-\epsilon, \epsilon)^N$ of the point $\vec{\lambda}$, the value of $\beta$ is given by:

$$\beta \triangleq \max_{\vec{\delta} \in (-\epsilon,\epsilon)^N} \frac{||\nabla^2\Phi(\vec{\lambda} + \vec{\delta})||}{2}$$

where $||\nabla^2\Phi(\vec{\lambda} + \vec{\delta})||$ represents the matrix norm of the matrix of second partials. Using (13) in (12) yields:

$$\begin{aligned}
&\mathbb{E}\left\{h(\vec{P}^*(t))\right\} \leq \\
&\quad \textstyle\sum_{\vec{H}} \alpha_{\vec{H}}(t)\left[\Phi(\vec{\lambda}) + N\epsilon^2\beta + \sum_{i=1}^{N}\frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon_i(H_i)\right] \\
&\quad = \Phi(\vec{\lambda}) + N\epsilon^2\beta + \textstyle\sum_{i=1}^{N}\frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\sum_{\vec{H}}\alpha_{\vec{H}}(t)\epsilon_i(H_i) \\
&\quad = \Phi(\vec{\lambda}) + N\epsilon^2\beta + \textstyle\sum_{i=1}^{N}\frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon\left[\alpha_i^R(t) - \alpha_i^L(t)\right] \quad (14)
\end{aligned}$$

where the last equality follows from (10).

To gain intuition, assume the system is ergodic and time averages exist. Taking time averages of (14) and (11) thus yields the following bound on time average power cost and the following expressions for time average transmission rates:

$$\overline{h} \leq \Phi(\vec{\lambda}) + N\epsilon^2\beta + \sum_{i=1}^{N}\frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon\left[\overline{\alpha}_i^R - \overline{\alpha}_i^L\right] \quad (15)$$

$$\overline{r}_i = \lambda_i + \epsilon\left[\overline{\alpha}_i^R - \overline{\alpha}_i^L\right] \quad \text{for } i \in \{1, \ldots, N\} \quad (16)$$

where $\overline{\alpha}_i^R, \overline{\alpha}_i^L$ are time average probabilities for backlog in queue $i$ being either right or left of the threshold. Note that for any stable system, the time average transmission rates are greater than or equal to the arrival rates, so that $\lambda_i \leq \overline{r}_i$. By (16), this implies that $\epsilon\left[\overline{\alpha}_i^R - \overline{\alpha}_i^L\right] \geq 0$ for all $i$. Because the partial derivatives $\partial\Phi(\vec{\lambda})/\partial\lambda_i$ are also non-negative, the final summation term in (15) is non-negative and hence cannot be neglected. However, note that the time average transmission rate on any link can only exceed the arrival rate due to edge effects. Indeed, the amount of bits delivered over any link $i$ on a given timeslot $t$ is exactly equal to the offered transmission rate $r_i(t)$ whenever $U_i(t) \geq \mu_{max}$. Hence, defining $\overline{\alpha}_i^E$ as the fraction of time that $0 \leq U_i(t) < \mu_{max}$, we have for all $i$:

$$\begin{aligned}
\lambda_i &\geq \text{time average bit output rate of queue } i \\
&\geq \overline{r}_i - \overline{\alpha}_i^E\mu_{max} \quad (17)
\end{aligned}$$

which holds because the transmission rate on any link is always less than or equal to $\mu_{max}$. From (17) and (16) we have:

$$\epsilon[\overline{\alpha}_i^R - \overline{\alpha}_i^L] \leq \overline{\alpha}_i^E\mu_{max} \quad (18)$$

It follows from (18) and (15) that if the edge probabilities $\overline{\alpha}_i^E$ are less than or equal to $O(\epsilon^2)$, then $\overline{h} - \Phi(\vec{\lambda}) \leq O(\epsilon^2)$. Fortunately, the system is designed to have a postive drift (away from the near empty edge region) whenever queue backlogs are below the $Q$ threshold. Hence, the edge probabilities can be made arbitrarily small by increasing the value of $Q$. This yields improved energy performance at the expense of increasing average queue occupancy and average delay. By setting $Q$ greater than or equal to $\Omega(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ and defining $V = 1/\epsilon^2$, we can show that the resulting algorithm is indeed ergodic and comes within a logarithmic factor of achieving the optimal energy-delay tradeoff curve of Theorem 2. Rather than proving this result, in the next section we use these ideas to prove a similar result for a more practical control strategy.

## V. The Tradeoff-Optimal Control Algorithm

To construct a more practical strategy, we use the concept of *Lyapunov drift*. Lyapunov drift theory has been useful in the development of stabilizing control algorithms for wireless networks [14]-[19], and recent extensions that treat stability and performance optimization simultaneously are developed in [8]-[11]. Here we extend the theory to treat performance optimization with near-optimal delay tradeoffs. The basic idea is to define a Lyapunov function that measures current queue congestion, and to make greedy decisions to minimize the Lyapunov function every timeslot based on current queue states and channel states. Such greedy decisions do not require knowledge of channel statistics or arrival rates, and hence offer a potential means of overcoming the complexity explosion problem described in the previous section.

### A. Algorithm Design Strategy

We shall consider the following Lyapunov function $L(\vec{U})$ consisting of a sum of exponentials:

$$L(\vec{U}) = \sum_i \left[e^{\omega(U_i - Q)} + e^{\omega(Q - U_i)} - 2\right] \quad (19)$$

where $Q$ is a positive buffer threshold and $\omega$ is a positive value affecting the rate of exponential increase of the Lyapunov function. We assume that $Q > \mu_{max}$ as before. This Lyapunov function reaches its minimum value $L(\vec{U}) = 0$ when $U_i = Q$ for all $i \in \{1, \ldots, N\}$, and increases exponentially when any of the $U_i$ components deviates from $Q$ either to the right or to the left. Rather than using the quadratic Lyapunov functions as in [14]-[19], [10], this exponential Lyapunov function is chosen to ensure a sufficiently small probability that queue backlog is within the near-empty edge region.

Allocating power to minimize the expected change in this Lyapunov function from one slot to the next *usually* creates a positive drift in the $i$th queue when $U_i(t) < Q$ and a negative drift in the $i$th queue when $U_i(t) \geq Q$. However, this is not always the case, and more structure is needed to achieve the same energy savings as the multi-modal drift algorithm of the previous section. In particular, we will find it is crucial to ensure that inequalities of the type (18) are satisfied.To this end, we define $\alpha_i^E(t) = Pr[U_i(t) < \mu_{max}]$, and define time average probabilities as follows:

$$\overline{\alpha}_i^L(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\alpha_i^L(\tau) \ , \quad \overline{\alpha}_i^R(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\alpha_i^R(\tau) \ ,$$
$$\overline{\alpha}_i^E(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\alpha_i^E(\tau) \ , \quad \text{for all } i \in \{1, \ldots, N\}$$

Note by definition that $\overline{\alpha}_i^L(t) + \overline{\alpha}_i^R(t) = 1$ (because backlog is either to the right or to the left of the $Q$ threshold), and $\overline{\alpha}_i^E(t) \leq \overline{\alpha}_i^L(t)$ (because if backlog is in the "near empty" edge region, then it must also be left of the $Q$ threshold). Similarly define the time average transmission rates $\overline{r}_i(t)$ as follows:

$$\overline{r}_i(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{r_i(\tau)\right\}$$

We have the following Lemma:

*Lemma 2:* For any queueing system described by the update equation (1), if the following conditions are satisfied for all $i \in \{1, \ldots, N\}$:

$$\liminf_{t\to\infty}\left(\overline{r}_i(t) - \lambda_i - \epsilon\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right]\right) \geq 0 \qquad (20)$$

then the following inequality holds:

$$\epsilon \limsup_{t\to\infty}\sum_i\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right] \leq \mu_{max}\limsup_{t\to\infty}\sum_i\overline{\alpha}_i^E(t)$$

*Proof:* The proof is given in Appendix B. □

For intuition, the reader can compare the above lemma to inequalities (16)-(18) from the previous section. To ensure that constraints (20) are satisfied, we use the notion of a *virtual queue* developed in [10]. For each $i$, we define a virtual queue $X_i(t)$ with the following update equation:

$$X_i(t+1) = \max[X_i(t) - (r_i(t) + \epsilon 1_i^L(t)), 0] + A_i(t) + \epsilon 1_i^R(t) \quad (21)$$

This update equation is identical to the equation representing a discrete time queue with inputs and transmission rates as shown in Fig. 3.

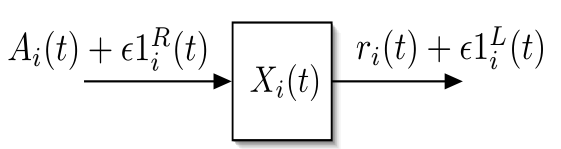


Fig. 3. An illustration of the virtual queue $X_i(t)$ associated with update equation (21).

*Definition 3:* A discrete time queue with an unfinished work process $U(t)$ is *strongly stable* if:

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

*Lemma 3:* If the queues $X_i(t)$ are strongly stable, then the time average conditions of (20) are satisfied for all $i$.

*Proof:* The proof of the lemma follows directly from the fact that if a queue with a bounded transmission rate is strongly stable then the $\liminf$ of the difference between the time average transmission rate and the time average arrival rate is greater than or equal to zero (see [10]). □

One of the objectives in our dynamic control algorithm shall be to stabilize both the actual and virtual queues of the system. We highlight the fact that this cannot be done via traditional Lyapunov stability theory. Indeed, Lyapunov theory is typically used by comparing a dynamic queue-length aware control strategy to a stationary queue-length independent control strategy (see [19] [11] for a detailed discussion of this). However, the inputs and server processes of the virtual queues $X_i(t)$ are highly dependent on the queue state of the system, as they are affected by queue backlogs $U_i(t)$ through the indicator functions $1_i^L(t)$ and $1_i^R(t)$. Our solution to this state-dependent control problem represents another significant contribution of the paper.

### B. The Dynamic Control Algorithm

The design principles we have developed lead to the following dynamic control algorithm. The algorithm is implemented for any control parameter $V > 0$, and for given positive parameters $\omega, Q, \epsilon$ (to be determined later as functions of $V$).

*Tradeoff Optimal Control Algorithm (TOCA):* The network controller performs the following operations every timeslot $t$:

1) Observe the current backlog and channel state vectors $\vec{U}(t)$, $\vec{S}(t)$ and allocate a power vector $\vec{P}(t) = \vec{P}$, where $\vec{P}$ solves the following problem:

$$\text{Minimize:} \quad Vh(\vec{P}) - \sum_{i=1}^N \hat{W}_i(t)\mu_i(\vec{P}, \vec{S}(t))$$
$$\text{Subject to:} \quad \vec{P} \in \Pi$$

where $\hat{W}_i(t) \triangleq \max[W_i(t), 0]$, and:

$$W_i(t) \triangleq 1_i^R(t)\left[\omega e^{\omega(U_i(t)-Q)} + 2X_i(t)\right] + 1_i^L(t)\left[-\omega e^{\omega(Q-U_i(t))} + 2X_i(t)\right]$$

2) Data is transmitted with rates $\vec{r}(t) = (r_1(t), \ldots, r_N(t))$, where:

$$r_i(t) = \begin{cases} \mu_i(\vec{P}(t), \vec{S}(t)) & \text{if } W_i(t) \geq 0 \\ 0 & \text{if } W_i(t) < 0 \end{cases}$$

3) Observe the current arrivals $A_i(t)$ and update the virtual queues $X_i(t)$ according to (21) (using the transmission rates $r_i(t)$ from step 2).

Note that the algorithm bases decisions only on the current system state, and does not require knowledge of traffic rates or channel statistics. The weights $W_i(t)$ contain terms that switch ON or OFF depending on whether $U_i(t)$ is to the left or right of the $Q$ threshold. This abrupt change in the weight functions effectively steers the drift so that queue backlogs spend the appropriate amount of time in each region.

The above algorithm requires only $O(1)$ multiply-add-exponentiate operations per link in order to update the virtual queue backlogs and to compute the weights $W_i(t)$. The power allocation optimization is the most complex part of the algorithm, although it can be solved easily for many systems, including for systems with concave rate-power curves with the structure given in (2), and for systems where there is only a finite (and small) number of power allocation options. This is discussed in more detail in Appendix D. Further note that queues do not need to be physically located in the same node, and hence the same algorithm and analysis applies to single-hop networks with multiple nodes. However, this would require distributed implementation of the power allocation optimization, which is simple in the case of separable cost functions $h(\cdot)$ and separable power curves as in (2), but may require more structured multiple access schemes for systems with inter-channel interference [10] [19]. A similar buffer partitioned strategy can be designed to treat multi-hop networks using *backpressure* [14], [8]-[11], [19], although we omit this topic for brevity.

### C. Performance

To simplify performance analysis, we assume that arrivals are bounded by a constant $A_{max}$ every slot, so that $A_i(t) \leq A_{max}$ for all $t$. Define $\delta_{max} \triangleq \max[\mu_{max}, A_{max}]$ as the maximum change in the backlog of any queue during a single timeslot. Further assume that the arrival rate matrix $\vec{\lambda}$ is strictly interior to the capacity region $\Lambda$ and has all positive entries. Define $\epsilon_{max}$ as the largest value $\epsilon$ such that $\epsilon \leq \lambda_i$ for all $i \in \{1, \ldots, N\}$, and such that $\vec{\lambda} + \vec{\epsilon} \in \Lambda$ (where $\vec{\epsilon}$ is a vector with all entries equal to $\epsilon$).

*Theorem 3:* (TOCA Performance) Suppose that $\Phi(\vec{\lambda})$ is twice differentiable at the point $\vec{\lambda}$. For any $V > \mu_{max}$ and for any $\omega > 0, \epsilon > 0$ chosen such that $\epsilon < \epsilon_{max}$ and satisfying:

$$\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon/\delta_{max} \tag{22}$$

the TOCA algorithm implemented with these parameters stabilizes all virtual and actual queues of the system. Furthermore:

(a) $\overline{\frac{1}{N}\sum_i U_i} \leq Q + \frac{1}{\omega}\log\left(\frac{D+Vh_{max}/N}{\omega\epsilon/2}\right)$

(b) $\overline{h} - \Phi(\vec{\lambda}) \leq \frac{ND}{V} + N\beta\epsilon^2 + \limsup_{t\to\infty}\sum_i \frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right]$

(c) $\limsup_{t\to\infty}\sum_i \frac{\epsilon}{\mu_{max}}\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right] \leq \frac{ND+Vh_{max}}{\omega\epsilon/2}e^{\omega(\mu_{max}-Q)}$

where the constant $\beta$ is from (13), $D$ and $h_{max}$ are:

$$D \triangleq e^{\omega(\mu_{max}+A_{max}-Q)} + \omega(\delta_{max}+\epsilon) + (A_{max}+\epsilon)^2 + (\mu_{max}+\epsilon)^2$$

$$h_{max} \triangleq \max_{\vec{P}\in\Pi} h(\vec{P})$$

and where the time averages are defined:

$$\overline{\frac{1}{N}\sum_i U_i} \triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\frac{1}{N}\sum_{i=1}^N U_i(\tau)\right\} \tag{23}$$

$$\overline{h} \triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{h(\vec{P}(\tau))\right\} \tag{24}$$

Theorem 3 is perhaps best interpreted through the following corollary:

*Corollary 1:* Under the assumptions of Theorem 3, for any $V > \mu_{max}$, if we choose:

$$\omega \triangleq \frac{\epsilon}{\delta_{max}^2}e^{-\epsilon/\delta_{max}} \tag{25}$$

and $\epsilon \triangleq 1/\sqrt{V}$, $Q \triangleq \frac{6}{\omega}\log(1/\epsilon)$, then TOCA yields:[5]

$$\begin{aligned}\overline{\frac{1}{N}\sum_i U_i} &\leq O(\sqrt{V}\log(V)) \\ \overline{h} - \Phi(\vec{\lambda}) &\leq O(1/V)\end{aligned}$$

It follows that per-bit delay is also less than or equal to $O(\sqrt{V}\log(V))$. This holds regardless of whether or not $\nabla^2\Phi(\vec{\lambda})$ is positive definite. If it is positive definite, then performance is governed by the Berry-Gallager bound of Theorem 2, and hence the TOCA algorithm yields performance within a logarithmic factor of the optimal energy-delay tradeoff curve. In Section VII we show that delay can be improved beyond the Berry-Gallager bound when $\Phi(\vec{\lambda})$ is piecewise linear (the case when the positive definite assumption fails).

The proof of Corollary 1 follows by showing that the definition of $\omega$ given in (25) satisfies the inequality in (22), and that $\omega = \Omega(1/\sqrt{V})$, $e^{-\omega Q} = O(1/V^3)$ (proof presented below). The proof of Theorem 3 is presented in the next section.

*Proof:* (Corollary 1) Note that the definition of $\omega$ in (25) implies that $\omega\delta_{max} \leq \epsilon/\delta_{max}$, and hence:

$$\begin{aligned}\omega\delta_{max}e^{\omega\delta_{max}} &\leq \omega\delta_{max}e^{\epsilon/\delta_{max}} \\ &= \epsilon/\delta_{max}\end{aligned}$$

where the final equality follows by the definition of $\omega$ in (25). Thus, the inequality (22) (required for Theorem 3) is satisfied. It follows that parts (a), (b), and (c) of Theorem 3 hold.

Because $Q = \frac{6}{\omega}\log(1/\epsilon)$, and $\epsilon = 1/\sqrt{V}$, we have:

$$e^{-\omega Q} = \epsilon^6 = 1/V^3 \tag{26}$$

Further, note that from (25) and the fact that $V > \mu_{max}$ we have:

$$\omega \leq \frac{\epsilon}{\delta_{max}^2} = \frac{1}{\delta_{max}^2\sqrt{V}} \leq \frac{1}{\delta_{max}^2\sqrt{\mu_{max}}} = O(1)$$

Thus:

$$e^{\omega\mu_{max}} = O(1) \tag{27}$$

Using (26) and (27), it follows that $e^{\omega(\mu_{max}-Q)} = O(1/V^3)$. Likewise, because $\epsilon = 1/\sqrt{V}$, and $V > \mu_{max}$, we have from (25):

$$\omega \geq \frac{1}{\delta_{max}^2\sqrt{V}}e^{-1/(\delta_{max}\sqrt{\mu_{max}})}$$

and hence $\omega = \Omega(1/\sqrt{V})$ and $1/(\omega\epsilon) = O(V)$. Therefore:

$$\begin{aligned}\frac{ND+Vh_{max}}{\omega\epsilon/2}e^{\omega(\mu_{max}-Q)} &\leq \frac{ND+Vh_{max}}{\omega\epsilon/2}\cdot O(1/V^3) \\ &= O(V+V^2)\cdot O(1/V^3) \\ &= O(1/V)\end{aligned}$$

[5]Throughout this paper, the $\log(\cdot)$ function denotes the natural logarithm.

where we have used the fact that $D = O(1)$. It follows from (b) and (c) of Theorem 3 that $\overline{h} - \Phi(\vec{\lambda}) \leq O(1/V)$. Similarly, it follows from (a) of Theorem 3 that $\overline{\frac{1}{N}\sum_i U_i} \leq O(\sqrt{V}\log(V))$. □

## VI. Performance Analysis

To prove Theorem 3, we first present the main results concerning Lyapunov drift with performance optimization from [8] [9] [10], presented here in a modified form. Consider any discrete time system that evolves according to a Markov chain with state space $\vec{Z}(t)$. Let $\Psi(\vec{Z})$ be a non-negative function of the state space vector. We call $\Psi(\vec{Z})$ a *Lyapunov function*, and define the *conditional Lyapunov drift* $\Delta(\vec{Z}(t))$ as follows:

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

*Lemma 4:* (Lyapunov Drift) If there exist random processes $f(t)$ and $g(t)$ such that for every timeslot and for all possible values of $\vec{Z}(t)$, the Lyapunov drift satisfies:

$$\Delta(\vec{Z}(t)) \leq \mathbb{E}\left\{f(t) - g(t) \mid \vec{Z}(t)\right\} \tag{28}$$

then:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g(\tau)\right\} \leq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau-0}^{t-1}\mathbb{E}\left\{f(\tau)\right\}$$

*Theorem 4:* (Lyapunov Optimization) Assume the $\vec{Z}(t)$ state space represents a set of queue backlog values. If there exist values $\epsilon > 0$, $B > 0$, $V > 0$, non-negative and upper bounded cost functions $c^*(t)$ and $c(t)$, and a non-negative function $q(\vec{Z})$ such that every timeslot and for all $\vec{Z}(t)$ values the Lyapunov drift sastisfies:

$$\Delta(\vec{Z}(t)) + V\mathbb{E}\left\{c(t) \mid \vec{Z}(t)\right\} \leq$$
$$B - \epsilon q(\vec{Z}(t)) + V\mathbb{E}\left\{c^*(t) \mid \vec{Z}(t)\right\}$$

then time average performance satisfies:

$$\overline{q(\vec{Z})} \leq \frac{B + V\overline{c^*}}{\epsilon}$$
$$\overline{c} \leq \overline{c^*} + B/V$$

where $\overline{q(\vec{Z})}, \overline{c}$, and $\overline{c^*}$ are the $\limsup$ expected time averages of their corresponding processes. That is:

$$\overline{q(\vec{Z})} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{q(\vec{Z}(\tau))\right\}$$
$$\overline{c} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{c(\tau)\right\}$$
$$\overline{c^*} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{c^*(\tau)\right\}$$

□

Lemma 4 follows by taking expectations of (28) and summing the telescoping series, and Theorem 4 follows from the lemma (see [11] [10] for details). The main idea of the theorem is that the difference between $\overline{c}$ and a target value $\overline{c^*}$ can be made arbitrarily small according to the control parameter $V$, with a corresponding increase in the time average of $q(\vec{Z}(t))$ that is linear in $V$.

### A. Computing the Drift

For the dynamic system under the TOCA algorithm, consider the Lyapunov function $\Psi(\vec{U}, \vec{X}) = L(\vec{U}) + J(\vec{X})$, where $L(\vec{U})$ is the exponential Lyapunov function of (19) associated with the actual system queues, and $J(\vec{X}) \triangleq \sum_i X_i^2$ is a quadratic Lyapunov function associated with the virtual queues. For ease of notation, we define the state variable $\vec{Z}(t) \triangleq [\vec{U}(t), \vec{X}(t)]$ and define $\Delta(\vec{Z}(t)) = \Delta_L(\vec{Z}(t)) + \Delta_J(\vec{Z}(t))$ to be the drift of the Lyapunov function $\Psi(\cdot)$, where $\Delta_L(\vec{Z}(t))$ and $\Delta_J(\vec{Z}(t))$ are drift components associated with the actual queues and virtual queues, respectively.

*Lemma 5:* If $\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon/\delta_{max}$, then:

(a) $$\Delta_L(\vec{Z}(t)) \leq Ne^{\omega(\mu_{max}+A_{max}-Q)} + N\omega(\delta_{max} + \tfrac{\epsilon}{2})$$
$$-\omega\textstyle\sum_i 1_i^R(t)e^{\omega(U_i(t)-Q)}\mathbb{E}\left\{r_i(t) - A_i(t) - \tfrac{\epsilon}{2} \mid \vec{Z}(t)\right\}$$
$$-\omega\textstyle\sum_i 1_i^L(t)e^{\omega(Q-U_i(t))}\mathbb{E}\left\{A_i(t) - r_i(t) - \tfrac{\epsilon}{2} \mid \vec{Z}(t)\right\}$$

(b) $$\Delta_J(\vec{Z}(t)) \leq N(A_{max} + \epsilon)^2 + N(\mu_{max} + \epsilon)^2$$
$$-2\textstyle\sum_i 1_i^L(t)X_i(t)\mathbb{E}\left\{r_i(t) - A_i(t) + \epsilon \mid \vec{Z}(t)\right\}$$
$$-2\textstyle\sum_i 1_i^R(t)X_i(t)\mathbb{E}\left\{r_i(t) - A_i(t) - \epsilon \mid \vec{Z}(t)\right\}$$

*Proof:* Part (b) of the lemma follows by squaring the virtual queue equations (21) and using a standard quadratic Lyapunov drift argument [11] (calculation omitted for brevity). Part (a) is proven in Appendix C by using the dynamic equation (1). □

By defining $\tilde{D} = e^{\omega(\mu_{max}+A_{max}-Q)} + \omega(\delta_{max} + \frac{\epsilon}{2}) + (A_{max} + \epsilon)^2 + (\mu_{max} + \epsilon)^2$ and using Lemma 5, we have:

$$\Delta(\vec{Z}(t)) + V\mathbb{E}\left\{h(\vec{P}(t)) \mid \vec{Z}(t)\right\} \leq \tilde{D}N$$
$$-\omega\textstyle\sum_i 1_i^R(t)e^{\omega(U_i(t)-Q)}\mathbb{E}\left\{r_i(t) - A_i(t) - \tfrac{\epsilon}{2} \mid \vec{Z}(t)\right\}$$
$$-\omega\textstyle\sum_i 1_i^L(t)e^{\omega(Q-U_i(t))}\mathbb{E}\left\{A_i(t) - r_i(t) - \tfrac{\epsilon}{2} \mid \vec{Z}(t)\right\}$$
$$-2\textstyle\sum_i 1_i^L(t)X_i(t)\mathbb{E}\left\{r_i(t) - A_i(t) + \epsilon \mid \vec{Z}(t)\right\}$$
$$-2\textstyle\sum_i 1_i^R(t)X_i(t)\mathbb{E}\left\{r_i(t) - A_i(t) - \epsilon \mid \vec{Z}(t)\right\}$$
$$+V\mathbb{E}\left\{h(\vec{P}(t)) \mid \vec{Z}(t)\right\} \tag{29}$$

The right hand side of the above drift expression depends on the allocation decisions $\vec{P}(t)$ and $\vec{r}(t)$ made by the controller at time $t$ (where $0 \leq r_i(t) \leq \mu_i(\vec{P}(t), \vec{S}(t))$). The construction of the TOCA algorithm from the previous section is now apparent: Allocating power according to the TOCA algorithm at timeslot $t$ minimizes the right hand side of the above drift expression over all possible power allocations $\vec{P}(t) \in \Pi$ and all possible rates $\vec{r}(t)$ such that $\vec{0} \leq \vec{r}(t) \leq \vec{\mu}(\vec{P}(t), \vec{S}(t))$. Indeed, it is clear from above that each weight $W_i(t)$ is simply the sum of coefficients multiplying the $r_i(t)$ variables. It follows that $\Delta(\vec{Z}(t))$ is less than or equal to the resulting expression when the power and rate decision variables $P_i(t)$ and $r_i(t)$ on the right hand side are replaced by the values $P_i^*(t)$, $r_i^*(t)$ corresponding to power and rates of the multi-modal stationary drift algorithm of Section IV with drift parameter $\epsilon$. The $P_i^*(t)$ and $r_i^*(t)$ values depend only on the current region of the $\vec{U}(t)$ vector within the set of $2^N$ regions of the backlog space, and satisfy (7) and (8). In particular, we have the identities:

$$\begin{aligned}\mathbb{E}\left\{r_i^*(t) - A_i(t) \mid U_i \geq Q\right\} &= (\lambda_i + \epsilon) - \lambda_i = \epsilon \\ \mathbb{E}\left\{A_i(t) - r_i^*(t) \mid U_i < Q\right\} &= \lambda_i - (\lambda_i - \epsilon) = \epsilon\end{aligned}$$

Using decision variables $r_i^*(t)$ and $P_i^*(t)$ together with these identities in the right hand side of the above drift expression makes the $X_i(t)$ terms vanish, and furthermore we have:

$$\begin{aligned}\Delta(\vec{Z}(t)) + V\mathbb{E}\left\{h(\vec{P}(t)) \mid \vec{Z}(t)\right\} \leq N\tilde{D} \\ -\omega \textstyle\sum_i 1_i^R(t)e^{\omega(U_i(t)-Q)}\frac{\epsilon}{2} \\ -\omega \textstyle\sum_i 1_i^L(t)e^{\omega(Q-U_i(t))}\frac{\epsilon}{2} \\ +V\mathbb{E}\left\{h(\vec{P}^*(t)) \mid \vec{Z}(t)\right\} \quad (30)\end{aligned}$$

Now note that $\frac{\omega\epsilon}{2}\sum_{i=1}^N 1_i^L(t)\left[1 - e^{\omega(U_i(t)-Q)}\right] \geq 0$ because $U_i(t) < Q$ whenever $1_i^L(t) = 1$. Similarly, $\frac{\omega\epsilon}{2}\sum_i 1_i^R(t)\left[1 - e^{\omega(Q-U_i(t))}\right] \geq 0$. Adding these non-negative terms to the right hand side of (30) and using the fact that $1_i^L(t) + 1_i^R(t) = 1$ yields:

$$\begin{aligned}\Delta(\vec{Z}(t)) + V\mathbb{E}\left\{h(\vec{P}(t)) \mid \vec{Z}(t)\right\} \leq ND \\ -\tfrac{\omega\epsilon}{2}\textstyle\sum_i \left[e^{\omega(U_i(t)-Q)} + e^{\omega(Q-U_i(t))}\right] \\ +V\mathbb{E}\left\{h(\vec{P}^*(t)) \mid \vec{Z}(t)\right\} \quad (31)\end{aligned}$$

where $D \triangleq \tilde{D} + \epsilon\omega/2$. The Lyapunov drift condition (31) is in the exact form for application of Theorem 4, and hence:

$$\overline{\sum_i e^{\omega(U_i-Q)}} \leq \frac{ND + Vh_{max}}{\omega\epsilon/2} \quad (32)$$

$$\overline{\sum_i e^{\omega(Q-U_i)}} \leq \frac{ND + Vh_{max}}{\omega\epsilon/2} \quad (33)$$

$$\overline{h} \leq \overline{h^*} + ND/V \quad (34)$$

where the overbar notation denotes the $\limsup$ expected time average, as in Theorem 4. Specifically, we note that $\overline{h^*}$ is defined:

$$\overline{h^*} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{h(\vec{P}^*(\tau))\right\}$$

It is important to note that $\overline{h^*}$ is *not* the time average cost that would be incurred if the multi-modal stationary drift algorithm from Section IV-A were used for all timeslots. That is because $\mathbb{E}\{h(\vec{P}^*(t))\}$ represents the expected cost that would be incurred if this algorithm were used only for slot $t$, but where the mode of the algorithm is based on the observed queue backlogs on slot $t$ which arose from having implemented TOCA on all previous slots. We now prove Theorem 3 directly from the inequalities (32), (33), and (34).

*Proof:* (Theorem 3 part (a)): By Jensen's inequality and convexity of the function $e^x$, we have:

$$e^{\omega\left(\frac{1}{N}\sum_{i=1}^N (U_i - Q)\right)} \leq \overline{\tfrac{1}{N}\textstyle\sum_{i=1}^N e^{\omega(U_i-Q)}}$$

Taking the $\log$ of both sides and using (32) yields:

$$\omega\overline{\tfrac{1}{N}\textstyle\sum_{i=1}^N (U_i - Q)} \leq \log\left(\frac{D + Vh_{max}/N}{\omega\epsilon/2}\right)$$

proving part (a) of Theorem 3. □

*Proof:* (Theorem 3 part (b)): Recall that $\vec{P}^*(t)$ is the power allocation vector that would be chosen by the stationary randomized policy of Section IV-A if $\vec{U}(t)$ is the observed backlog vector. If $\vec{U}(t)$ is in a specific region $\vec{H}$ of the state space on slot $t$, then this stationary randomized policy is designed to incur a conditional expected cost of $\Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H}))$. As in (12), the unconditional expected cost on timeslot $t$ is thus given by conditioning on all possible regions $\vec{H}$:

$$\mathbb{E}\left\{h(\vec{P}^*(t))\right\} = \sum_{\vec{H}} \alpha_{\vec{H}}(t)\Phi(\vec{\lambda} + \vec{\epsilon}(\vec{H})) \quad (35)$$

The above equation expresses $\mathbb{E}\{h(\vec{P}^*(t))\}$ in terms of the $\alpha_{\vec{H}}(t)$ probabilities, and is identical to the equation (12) used in Section IV-A for analysis of the multi-modal stationary drift algorithm. However, the probabilities $\alpha_{\vec{H}}(t)$ used in (12) correspond to the probabilities that the backlog vector is in the $\vec{H}$ region on slot $t$ when the multi-modal stationary drift policy is used for all time, while the $\alpha_{\vec{H}}(t)$ probabilities in (35) are different and correspond to the system that is implementing TOCA every timeslot.

Using the same derivation as in Section IV-A, where it is shown that manipulating the right hand side of (12) leads directly to (14), we find that the right hand side of (35) can be re-written to have the same form as the right hand side of (14). Specifically, from (14) we have for all $t$:

$$\mathbb{E}\left\{h(\vec{P}^*(t))\right\} \leq \Phi(\vec{\lambda}) + N\epsilon^2\beta + \sum_{i=1}^N \frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon\left[\alpha_i^R(t) - \alpha_i^L(t)\right]$$

where the probabilities $\alpha_i^R(t)$ and $\alpha_i^L(t)$ used above correspond to the system that implements TOCA every slot. Summing over $\tau \in \{0, \ldots, t-1\}$ and taking a $\limsup$ yields:

$$\begin{aligned}\overline{h^*} &\triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t-1}\mathbb{E}\left\{h(\vec{P}^*(\tau))\right\} \\ &\leq \Phi(\vec{\lambda}) + N\epsilon^2\beta \\ &\quad + \limsup_{t\to\infty}\sum_{i=1}^N \frac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\epsilon\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right]\end{aligned}$$

where we recall that $\overline{\alpha}_i^R(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\alpha_i^R(\tau)$, and $\overline{\alpha}_i^L(t)$ is defined similarly. Using this inequality together with (34) proves part (b) of Theorem 3. □

*Lemma 6:* All actual and virtual queues are strongly stable.

*Proof:* The time average bound in part (a) of the Theorem demonstrates strong stability of all actual queues $\vec{U}(t)$. Similarly, it can be shown that all virtual queues $\vec{X}(t)$ are strongly stable. As an outline of this, note that instead of substituting the power allocation policy $P_i^*(t)$ into (29) (which yields $\mathbb{E}\{r_i^*(t) \mid U_i(t) < Q\} = \lambda_i - \epsilon$ and $\mathbb{E}\{r_i^*(t) \mid U_i(t) \geq Q\} = \lambda_i + \epsilon$), one can consider the policy $\hat{P}_i(t)$ that yields $\mathbb{E}\{\hat{r}_i(t) \mid U_i(t) < Q\} = \lambda_i - \epsilon/2$ and $\mathbb{E}\{\hat{r}_i(t) \mid U_i(t) \geq Q\} = \lambda_i + \epsilon + \delta$, where $\delta > 0$ and satisfies $\vec{\lambda} + \vec{\epsilon} + \vec{\delta} \in \Lambda$. Such a policy yields drift coefficients of $-\epsilon$ and $-2\delta$, respectively, multiplying the $1_i^L(t)X_i(t)$ and $1_i^R(t)X_i(t)$ terms. □

*Proof:* (Theorem 3 part (c)): Recall that $\alpha_i^E(t) = Pr[U_i(t) < \mu_{max}]$. Clearly we have:

$$\begin{aligned} \mathbb{E}\left\{e^{\omega(Q-U_i(t))}\right\} &\geq \mathbb{E}\left\{e^{\omega(Q-U_i(t))} \mid U_i(t) < \mu_{max}\right\}\alpha_i^E(t) \\ &\geq e^{\omega(Q-\mu_{max})}\alpha_i^E(t) \end{aligned}$$

Summing over $\tau \in \{0, \ldots, t-1\}$ and $i \in \{1, \ldots, N\}$ yields:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{i=1}^{N}\mathbb{E}\left\{e^{\omega(Q-U_i(\tau))}\right\} \geq \sum_{i=1}^{N}\overline{\alpha}_i^E(t)e^{\omega(Q-\mu_{max})}$$

Taking the $\limsup$ of both sides and using (33) yields:

$$\limsup_{t\to\infty}\sum_{i=1}^{N}\overline{\alpha}_i^E(t) \leq \frac{ND + Vh_{max}}{\omega\epsilon/2}e^{\omega(\mu_{max}-Q)} \quad (36)$$

From Lemma 6 we know that all actual and virtual queues of the system are stable. It follows from Lemma 3 in Section V that conditions (20) are satisfied, and hence the result of Lemma 2 holds. Specifically, this implies:

$$\frac{\epsilon}{\mu_{max}}\limsup_{t\to\infty}\sum_{i=1}^{N}\left[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)\right] \leq \limsup_{t\to\infty}\sum_{i=1}^{N}\overline{\alpha}_i^E(t)$$

The above inequality places a lower bound on the left hand side of (36), proving part (c) of Theorem 3. □

## VII. BEYOND THE BERRY-GALLAGER BOUND

Here we demonstrate a mode of "super-fast" convergence in the case when the minimum energy function $\Phi(\vec{\lambda})$ is piecewise linear about the rate vector $\vec{\lambda}$. Such cases are important and occur when there are only a finite number of power or rate options, such as when we can add either full power to a single queue or no power to any queue, or when practical coding schemes restrict to a finite set of transmission rates. In such cases, the $\Phi(\vec{\lambda})$ function has a polyhedral structure with a finite number of vertices. We assume the $\vec{\lambda}$ point is not a vertex.

In this case, the Taylor expansion of $\Phi(\vec{\lambda}+\vec{\epsilon}(\vec{H}))$ in (13) has a second order coefficient $\beta$ that is equal to zero whenever $\epsilon$ is within some range $0 \leq \epsilon \leq \epsilon_{max}$ (for some value $\epsilon_{max}$). Using the fact that $\beta = 0$ and combining the performance bounds in parts (b) and (c) of Theorem 3 yields the following bound on average power for any algorithm satisfying the conditions of Theorem 3:

$$\begin{aligned} \overline{h} - \Phi(\vec{\lambda}) \leq \tfrac{ND}{V} + \\ \max_i\left\{\tfrac{\partial\Phi(\vec{\lambda})}{\partial\lambda_i}\right\}\left(\tfrac{ND+Vh_{max}}{\omega\epsilon/(2\mu_{max})}\right)e^{\omega\mu_{max}}e^{-\omega Q} \quad (37) \end{aligned}$$

This leads to the following corollary:

*Corollary 2:* For the case $\beta = 0$, implementing the TOCA algorithm with parameters $V, \epsilon$ such that $V > \mu_{max}$, $0 < \epsilon < \epsilon_{max}$, and choosing $\omega = \frac{\epsilon}{\delta_{max}^2}e^{-\epsilon/\delta_{max}}$, $Q = \frac{2}{\omega}\log(V)$ yields:

$$\begin{aligned} \overline{\tfrac{1}{N}\textstyle\sum_i U_i} &\leq O(\log(V)) \\ \overline{h} - \Phi(\vec{\lambda}) &\leq O(1/V) \end{aligned}$$

*Proof:*

To prove the average backlog bound, note that $\epsilon$ and $\omega$ no longer depend on $V$. Thus, the constants $\epsilon$, $\omega$, and $D$ are all $O(1)$, and so (37) yields:

$$\overline{h} - \Phi(\vec{\lambda}) \leq O(1/V) + O(V)\cdot e^{-\omega Q}$$

However,

$$e^{-\omega Q} = e^{-2\log(V)} = 1/V^2$$

and hence we have $\overline{h} - \Phi(\vec{\lambda}) \leq O(1/V)$. Likewise, the proof of the backlog bound follows immediately from part (a) of Theorem 3. □

One might expect that the logarithmic delay bound is an artifact of the exponential Lyapunov function that we have chosen, and that another Lyapunov function (perhaps doubly exponential) might yield sub-logarithmic delay. However, this is not the case. Indeed, below we present a simple example of a system with a piecewise linear $\Phi(\vec{\lambda})$ function for which it is impossible to design an algorithm that achieves an energy-delay tradeoff curve better than the one we have proven in the above corollary. Hence, our analysis captures the tightest possible tradeoff.

### A. A Simple Example

Consider a single queue with a single input stream of rate $\lambda$ (in units of packets/slot). Every timeslot the controller decides to either transmit or remain idle, expending one Watt of power when transmitting and 0 Watts when idle. The channel state varies in an i.i.d. fashion between "Good" and "Bad" states, each equally likely. Two packets can be transmitted in a single slot during a "Good" channel state, while only one packet can be transmitted during a "Bad" channel state. It is not difficult to show that the capacity region for this system is the set of all rates $\lambda$ such that $0 \leq \lambda \leq 1.5$, and for the objective function $h(P) = P$ the minimum energy function $\Phi(\lambda)$ is given by the following piecewise linear curve:

$$\Phi(\lambda) = \begin{cases} \lambda/2 & \text{if } 0 \leq \lambda \leq 1 \\ 0.5 + (\lambda - 1) & \text{if } 1 < \lambda \leq 1.5 \end{cases}$$

Intuitively, the minimum energy policy is to use as many of the "Good" channel states as possible, and transmit in "Bad" channel states only when absolutely necessary. Suppose the input to the system is i.i.d. and such that 2 packets arrive during a timeslot with probability $p$, and no packets arrive otherwise. The input rate is thus $\lambda = 2p$ (in units of packets per slot), and we assume that $p = 1.25/2$ so that $\lambda = 1.25$ in this example. Thus, $\Phi(\lambda) = 0.75$ Watts.

Consider any control algorithm that stabilizes the system, and for simplicity assume the algorithm is ergodic and yields a well defined steady state. Using standard interchange arguments, it is not difficult to show that any algorithm that does not transmit during a "Good" channel state when the queue has at least two packets can be improved in both energy and delay by transmitting the packets. Hence, we assume the policy always transmits in such a scenario. Define $\lambda_g$ as the rate that the algorithm delivers packets to the destination, considering only packets transmitted in the "Good" state when there are two or more packets in the queue. It follows that $(1.25-\lambda_g)$ is the rate of all other packets, which includes packets transmitted in "Bad" channel states and packets transmitted in "Good" channel states that are "under-utilized." We thus have:

$$\overline{P} = \lambda_g\frac{1}{2} + (1.25 - \lambda_g) = 1.25 - \lambda_g/2$$

and hence $\overline{P} - \Phi(\lambda) = 0.5 - \lambda_g/2$.

Let $U(t)$ represent the number of packets in the system at time $t$, and let $\overline{U}$ represent the steady state average. Note that, in steady state, $Pr[U(t) \leq 2\overline{U}] \geq 1/2$ (by the Markov inequality for non-negative random variables $U$). Every timeslot in which the system has at least two packets, the queue independently decreases by two packets with probability $(1-p)/2$ (the probability that no new packet arrives, and the channel state is "Good" so that two packets are transmitted). Define $T$ as the smallest integer larger than $\overline{U}$. The probability that the system has fewer than two packets at a particular time $t$ is thus greater than or equal to the probability that $U(t-T) \leq 2\overline{U}$, and then having $T$ successive timeslots when no packets arrive but channel states are "Good." Thus:

$$Pr[\text{fewer than two packets}] \geq \frac{1}{2}\left(\frac{1-p}{2}\right)^{\overline{U}+1} \triangleq \delta \qquad (38)$$

where we have defined $\delta$ as the lower bound on the probability the system has fewer than two packets. Because a "Good" channel state arises every timeslot with probability 1/2, and this is independent of whether or not the queue has two or more packets at the beginning of that slot, it follows that the fraction of unused or under-utilized "Good" channel states is at least $\delta$. Hence, $\lambda_g \leq 2(\frac{1}{2} - \frac{\delta}{2}) = 1 - \delta$, and so:

$$\begin{aligned} \overline{P} - \Phi(\lambda) &= 0.5 - \lambda_g/2 \\ &\geq 0.5 - (1-\delta)/2 \\ &= \delta/2 \end{aligned}$$

Defining $V \triangleq 2/\delta$, we have $\overline{P} - \Phi(\lambda) \geq 1/V$. However, by definition of $\delta$ in (38), we have:

$$(\overline{U}+1)\log((1-p)/2) = \log(2\delta) = \log(4/V) \qquad (39)$$

and hence:

$$\overline{U} = \frac{\log(V/4)}{\log(2/(1-p))} - 1$$

This proves that average delay increases logarithmically in the $V$ parameter while the distance to the minimum energy level $\Phi(\lambda)$ is necessarily greater than or equal to $1/V$.

## VIII. SIMULATIONS

Our simulations of TOCA (presented below) reveal that average queue backlog is very close to $Q$, which is not surprising because the drift is designed to push backlog towards this value. We note that the value $Q = \frac{6}{\omega}\log(1/\epsilon)$ for ordinary TOCA in Corollary 1, and $Q = \frac{2}{\omega}\log(V)$ for "super-fast" TOCA in Corollary 2, were both chosen only to ensure a sufficiently small probability that queue backlog is in the "near empty" edge region. However, our analysis was conservative, and numerous simulations revealed that average queue backlog was in the edge region only for 4 or 5 timeslots out of a total duration of 10 million slots. In practice, a constant factor improvement in average queue backlog can be obtained by appropriately reducing the value of $Q$, without significantly affecting edge probability or average power expenditure. This suggests a modified version of TOCA in which the parameter $Q$ is adaptively adjusted according to an empirical edge probability measurement. In the following simulations we present results for the TOCA algorithm with the original threshold value $Q$, a reduced threshold $Q/f$ (for some pre-specified value $f \geq 1$), as well as a heuristic algorithm that makes online adjustments to the threshold level.

### A. A Single Data Link

Consider first a single queue with a single time varying channel and with a rate-power curve given by:

$$\mu(P(t), S(t)) = \log(1 + S(t)P(t))$$

where $0 \leq P(t) \leq P_{max}$, and where there are two channel states chosen independently and equally likely every timeslot: $S(t) \in \{\sigma_1, \sigma_2\}$. This simple system is considered to enable a direct comparison between TOCA and the Berry-Gallager algorithm of Section IV when both use the same $Q$ and $\epsilon$ parameters. Indeed, in this special case with only two channel states, we can solve for the appropriate powers $P_{L,\sigma_1}, P_{L,\sigma_2}, P_{R,\sigma_1}, P_{R,\sigma_2}$ required to implement the Berry-Gallager algorithm:

$$\begin{aligned} P_{L,\sigma_1} &= \min\left[\max\left[\frac{e^{\lambda-\epsilon}}{\sqrt{\sigma_1\sigma_2}} - \frac{1}{\sigma_1}, 0\right], P_{max}\right] \\ P_{L,\sigma_2} &= \frac{e^{2(\lambda-\epsilon)}}{\sigma_2(1+\sigma_1 P_{L,\sigma_1})} - \frac{1}{\sigma_2} \end{aligned}$$

and where $P_{R,\sigma_1}$ and $P_{R,\sigma_2}$ have similar expressions with "$-\epsilon$" replaced with "$\epsilon$." The Berry-Gallager algorithm proceeds by allocating power $P(t) = P_{L,\sigma_1}$ whenever the queue backlog $U(t)$ is to the left of the $Q$ threshold and when $S(t) = \sigma_1$, and by similarly allocating the other three power levels when the system is in the corresponding state.

We consider system parameters $\sigma_1 = 1, \sigma_2 = 2$, $A_{max} = 2\log(2.5)$, $P_{max} = 2$, $\lambda = \log(2.5)$, and assume the input process is i.i.d. over slots with an arrival of $A_{max}$ bits with probability $1/2$ and 0 bits with probability $1/2$. The $Q$ and $\epsilon$ parameters are given by:

$$\epsilon = 1/\sqrt{V} \ , \ Q = \frac{6}{\omega}\log(1/\epsilon)$$

where $\omega$ is defined in (25).

Figure 4 compares the energy-delay performance of the Berry-Gallager algorithm of Section IV and the TOCA algorithm of Section V. Experiments were performed for $V \in \{5, 10, 20, 40, 80, 160, 320, 640, 1280, 5000\}$, and each experiment was simulated for 10 million timeslots. The curves "$Q$ TOCA" and "$Q$ Berry-Gallager" represent results using the original threshold $Q = \frac{6}{\omega}\log(1/\epsilon)$ (from Corollary 1), and the corresponding "$Q/10$" curves show results when this threshold is reduced by a factor of 10. We found that this factor of 10 reduction did not significantly effect edge probabilities, and hence this reduction simply reduced average delay by the same amount without significantly effecting average power expenditure.

Note that the TOCA and Berry-Gallager algorithms yield curves that are very close to each other (for both sets of threshold values). This demonstrates that the dynamic decisions of TOCA can closely match the performance of the

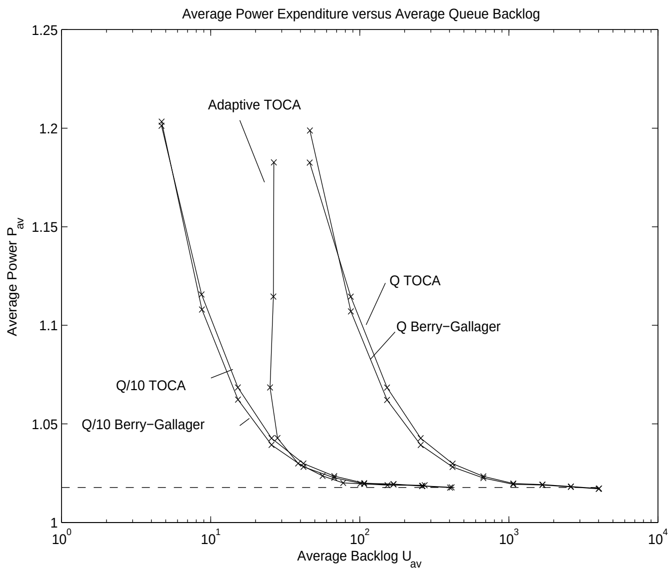


Fig. 4. Comparing TOCA and the Berry-Gallager algorithm for a single queue with a logarithmic rate-power curve. Thresholds $Q$ and $Q/10$ are used. Also shown is the threshold-adaptive version of TOCA.

Berry-Gallager algorithm, without requiring any of the pre-computation or a-priori statistical knowledge of the Berry-Gallager algorithm. We further note that TOCA can be implemented just as easily in the case when there are 500 or 5000 possible channel states $\{\sigma_i\}$ with different and unknown likelihoods (rather than just 2 equally probable channel states). It would be difficult or impossible to solve for the appropriate power allocations required to implement the Berry-Gallager algorithm in such scenarios.

### B. Threshold Adaptive TOCA

The results of Fig. 4 show that significant (constant factor) reductions in average delay can be achieved by reducing the threshold value $Q$. Using TOCA with any fixed threshold value that maintains an edge probability less than or equal to $O(1/V)$ will lead to the desired energy-delay performance. However, reducing the threshold too much can effect edge probabilities, leading to a loss of energy efficiency. Here we consider a modified TOCA algorithm that automatically adapts the threshold to ensure edge effects are sufficiently rare. Specifically, we define the parameter $\tilde{Q}(t)$ as follows:

$$\tilde{Q}(t) = \min[Q, \max[10, Q/f(t)]]$$

where $f(t)$ is a suitable reduction factor (with $f(t) \in \{1, \ldots, 100\}$). The parameter $\tilde{Q}(t)$ acts as a replacement for the original threshold $Q$ during slot $t$. Let $f(0) = f_0$ represent the initial threshold reduction factor (we use $f_0 = 10$ in our experiments). Let $T_{frame}$ represent an integer frame length parameter. A timer variable is initialized to zero and is incremented by 1 at the end of every slot, being reset to zero whenever it reaches the value $T_{frame}$. At the end of every slot $t$, if $U_i(t) < \mu_{max}$ for any queue $i$, an "edge event" is declared: The timer is reset to zero and we choose $f(t+1) = \max[f(t) - 1, 1]$. If there is no edge event during slot $t$ but the timer variable is less than $T_{frame}$, we choose $f(t+1) = f(t)$. If there is no edge event and the timer is equal to $T_{frame}$, then we reset the timer to zero and choose $f(t+1) = \min[f(t) + 1, 100]$.

Thus, $f(t)$ is reduced whenever edge events occur, and is increased when there are no edge events during a frame of $T_{frame}$ slots. We choose:

$$T_{frame} = \max[20000, 10V]$$

This is designed to intuitively maintain an edge probability less than or equal to $\min[.00005, 1/(10V)]$, and this intuition is confirmed in our simulation experiments. Fig. 4 shows the performance of this adaptive TOCA algorithm applied to the single queue downlink of the previous section. The curve is almost identical under different initial reduction factors (such as $f_0 \in \{1, 2, 5, 10\}$). The algorithm quickly "learns" to use threshold factors in the range of 8 and 12 for this particular system.

### C. A 2-User Downlink

Here we implement the "super-fast" version of TOCA (from Corollary 2) for a 2-queue downlink with a piecewise linear minimum energy function. Data arrives from two independent streams with rates $(\lambda_1, \lambda_2)$ (in units of packets/slot), and the first stream of packets must be transmitted over channel 1 while the second must be transmitted over channel 2. Every timeslot, the network controller decides either to allocate one unit of power to a single channel or to allocate no power to any channel. We use the same channel model as in [10], where each of the two channels can be in "Good," "Medium," or "Bad" channel states. If power is allocated to a particular channel when it is in the "Good" state, 3 packets can be transmitted from queue $i$ during that slot (for $i \in \{1, 2\}$). Two packets can be transmitted in the "Medium" state, and only one packet during the "Bad" state. Channel state vectors are independent and identically distributed over slots, with probabilities $Pr(State_1, State_2)$ given by:

$$Pr(G, M) = 1/3, Pr(M, B) = 2/9, Pr(M, M) = 1/9$$
$$Pr(G, B) = 2/9, Pr(M, G) = 1/9$$

Inputs are Bernoulli with packet arrival probabilities given by $\lambda_1 = 8/9, \lambda_2 = 5/9$. The minimum average energy expenditure required for stability of this system can be exactly computed, and is given by $P_{av}^* = 14/27 \approx 0.518519$. We implement the "super-fast" TOCA with $\epsilon = 2/9$. Figs. 5 and 5 illustrate performance for simulations of super-fast TOCA (from Corollary 2) with$Q = \frac{2}{\omega}\log(V)$. Also shown are simulations for $\tilde{Q} = Q/20$, for the adaptive threshold version of TOCA as described in the previous subsection, and for the EECA algorithm of [10]. Note that the EECA algorithm is designed to achieve an energy-delay tradeoff of $[O(1/V), O(V)]$.

Fig. 5 illustrates average power expenditure versus the $V$ parameter (where $V$ is varied from 5 to 5000). Note that the curves for "TOCA $Q$" and "TOCA $Q/20$" are almost indistinguishable, as the threshold reduction does not significantly affect edge probability. All curves demonstrate similar

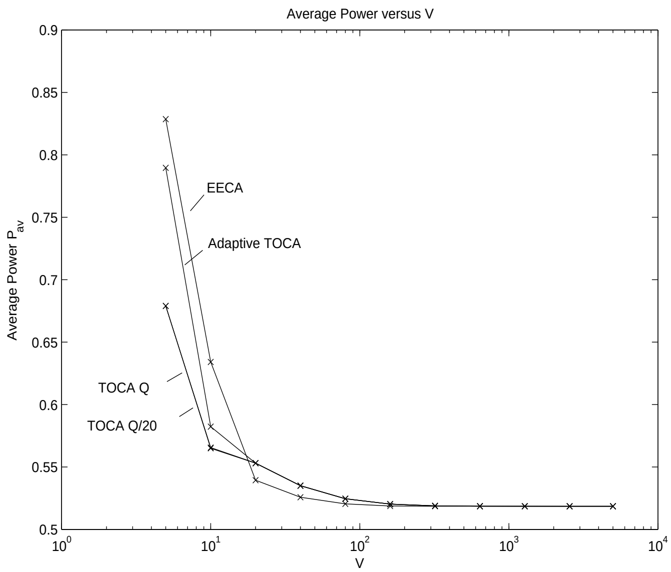


Fig. 5. Simulation of the $Q$, $Q/20$, and threshold adaptive versions of "super-fast" TOCA for the 2 queue downlink, together with the EECA algorithm of [10].

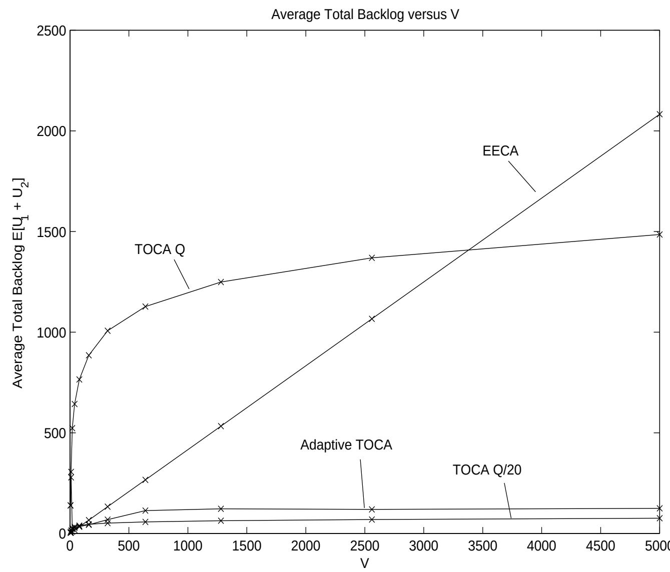


Fig. 6. Average backlog for the same experiments as in Fig. 5, illustrating logarithmic behavior for the TOCA algorithms and linear behavior for EECA.

performance, showing that average energy converges to within $O(1/V)$ of the minimum energy value $P_{av}^*$. In Fig. 6 the corresponding average backlog data is plotted for these same experiments. This reveals that the super-fast TOCA algorithms have average backlog (and hence, average delay) that is logarithmic in the $V$ parameter, while EECA is linear. The $Q/20$ and threshold adaptive versions of TOCA are seen to yield significant delay savings than the original TOCA. We note that at the value $V = 5000$, all algorithms yield average power within four or five significant digits of $P_{av}^*$. However, it is important to note that, while the EECA algorithm does not provably yield a "super-fast" logarithmic delay tradeoff, we found its performance to be quite competitive with the adaptive TOCA algorithms when average power is required to be within only 1 or 2 significant digits of optimality.

## IX. Conclusions

We have established a fundamental tradeoff between energy and delay for multi-user wireless networks. This work extends the tradeoff results developed for a single link by Berry and Gallager, and demonstrates for the first time that the square-root delay tradeoff is both necessary and achievable (to within a logarithmic factor) for general systems with multiple queues, multiple users, and non-linear power curves. Furthermore, we discovered an important class of piecewise linear systems that beat the tradeoff to achieve "super-fast" logarithmic delay. Our algorithms make use of a novel technique of "Lyapunov drift steering" that switches discontinuously between different weights to drive average delay toward the optimal tradeoff curve. This approach overcomes an inherent state space explosion associated with delay optimization in multi-user systems. The resulting control algorithms are simple and do not require knowledge of traffic rates or channel statistics. While our analysis focused on the case of a multi-user downlink, we note that these same techniques can be used to solve multi-node, multi-hop networking problems, and are likely to yield fundamental improvements in other networking, control, and optimization contexts.

## Appendix A — Multi-User Berry-Gallager Bound

Here we prove Theorem 2. Without loss of generality, we can assume that $\lambda_i > 0$ for all $i$. Consider any policy within a sequence of admissible policies of the type specified in Section III. Recall that such a policy makes stationary (potentially randomized) power allocation decisions based only on the current backlog and channel state vectors $\vec{U}(t)$ and $\vec{S}(t)$. Let $\vec{P}(\vec{U}, \vec{S})$ be the random vector representing the power allocated under state $\vec{U}, \vec{S}$. Define $\vec{\gamma}(\vec{U}) \triangleq \mathbb{E}\left\{\vec{\mu}(\vec{P}(\vec{U}, \vec{S}), \vec{S}) \mid \vec{U}(t) = \vec{U}\right\}$ as the average transmission rate vector when the queue backlog is $\vec{U}$. Let $\pi(\vec{U})$ represent the steady state distribution of the $\vec{U}$ vector, and let $\mathbb{E}\{U_i\}$ be the steady state average backlog in queue $i$ for $i \in \{1, \ldots, N\}$. Further, define the *expected queue drift* $\vec{\delta}(\vec{U})$ as follows:

$$\vec{\delta}(\vec{U}) \triangleq \vec{\gamma}(\vec{U}) - \vec{\lambda} \tag{40}$$

The value of $\vec{\delta}(\vec{U})$ represents the expected difference between transmission rates and arrivals when the queue state is $\vec{U}$. The following lemma, which is a modified version of a similar lemma given in [2], bounds the tail behavior of $\vec{\delta}(\vec{U})$:

*Lemma 7:* For any policy within an admissible sequence of policies with constants $\theta_1, \theta_2$ as defined in Section III, and for any $i \in \{1, \ldots, N\}$, there exists a value $u_i$ such that:

$$\int \cdots \int_{\{\vec{U} \mid U_i > u_i\}} \delta_i(\vec{U}) d\pi(\vec{U}) \geq \frac{\theta_1 \theta_2^2}{16\mathbb{E}\{U_i\}} \tag{41}$$

The proof of the lemma is almost identical to the proof of a similar statement given in [2], and is omitted for brevity. Now let $\overline{h}$ represent the average power cost of the given admissible policy.

*Lemma 8:* For any admissible policy with average power cost $\overline{h}$ and total average congestion $\overline{U}_{tot} \triangleq \sum_{i=1}^{N} \mathbb{E}\{U_i\}$, we have:

$$\overline{h} - \Phi(\vec{\lambda}) \geq \Omega\left((1/\overline{U}_{tot})^2\right)$$

*Proof:* Our proof closely follows the original proof by Berry and Gallager in [2], and generalizes the result to the case of multiple dimensions. Note that $\overline{h}$ can be written in terms of the random $\vec{P}(\vec{U}, \vec{S})$ vectors as follows:

$$\overline{h} = \int \ldots \int_{\vec{U}} \mathbb{E}\left\{h(\vec{P}(\vec{U}, \vec{S})) \mid \vec{U}\right\} d\pi(\vec{U})$$

The value of $\mathbb{E}\left\{h(\vec{P}(\vec{U}, \vec{S})) \mid \vec{U}\right\}$ is the power cost expended to achieve an average transmission rate vector $\vec{\gamma}(\vec{U})$. By definition, $\Phi(\vec{\gamma}(\vec{U}))$ is the minimum cost required to achieve an average rate vector $\vec{\gamma}(\vec{U})$, and hence $\mathbb{E}\left\{h(\vec{P}(\vec{U}, \vec{S})) \mid \vec{U}\right\} \geq \Phi(\vec{\gamma}(\vec{U}))$. We thus have:

$$\begin{aligned} \overline{h} &\geq \int \ldots \int_{\vec{U}} \Phi(\vec{\gamma}(\vec{U})) d\pi(\vec{U}) \\ &= \int \ldots \int_{\vec{U}} \Phi(\vec{\lambda} + \vec{\delta}(\vec{U})) d\pi(\vec{U}) \\ &= \int \ldots \int_{\vec{U}} \Big[\Phi(\vec{\lambda}) + \nabla\Phi(\vec{\lambda}) \cdot \vec{\delta}(\vec{U}) \\ &\qquad + G(\vec{\delta}(\vec{U}))\Big] d\pi(\vec{U}) \end{aligned} \quad (42)$$

where $\nabla\Phi(\vec{\lambda})$ represents the vector of partial derivatives of $\Phi(\vec{\lambda})$ at the point $\vec{\lambda}$, and $G(\vec{\delta})$ is the error term in the first order Taylor expansion of $\Phi(\vec{\lambda})$ about the point $\vec{\lambda}$. Note that $G(\vec{0}) = 0$ and $\nabla G(\vec{0}) = \vec{0}$. Because $\Phi(\vec{\lambda})$ is convex, it follows that $G(\vec{\delta})$ is non-negative and convex for all $\vec{\delta} \in \mathbb{R}^N$. Further, because $\Phi(\vec{\lambda})$ is twice differentiable at $\vec{\lambda}$ with a positive definite matrix of partial derivatives $\nabla^2\Phi(\vec{\lambda})$, it follows by the multi-dimensional Taylor theorem that:

$$G(\vec{\delta}) \geq \Omega(||\vec{\delta}||^2) \quad (43)$$

Now note that because all queues of the system are stable, it must be the case that the average transmission rate on each link $i$ is greater than or equal to $\lambda_i$, and so we have the following entrywise inequality: $\vec{\lambda} \leq \int \ldots \int_{\vec{U}} \vec{\gamma}(\vec{U}) d\pi(\vec{U})$. Hence, by the definition $\vec{\delta}(\vec{U}) \triangleq \vec{\gamma}(\vec{U}) - \vec{\lambda}$ we have:

$$\int \ldots \int_{\vec{U}} \vec{\delta}(\vec{U}) d\pi(\vec{U}) \geq \vec{0} \quad (44)$$

Using (44) in (42) together with the fact that all entries of $\nabla\Phi(\vec{\lambda})$ are non-negative, we have:

$$\overline{h} \geq \Phi(\vec{\lambda}) + \int \ldots \int_{\vec{U}} G(\vec{\delta}(\vec{U})) d\pi(\vec{U}) \quad (45)$$

Now let $i^* = \arg\min_{i \in \{1,\ldots,N\}} \mathbb{E}\{U_i\}$, and note that $\mathbb{E}\{U_{i^*}\} \leq \overline{U}_{tot}/N$. Let $u_i^*$ represent the corresponding value satisfying (41) of Lemma 7 for $i = i^*$. By non-negativity of $G(\vec{\delta})$ for all $\vec{\delta}$, from (45) we have:

$$\begin{aligned} \overline{h} &\geq \Phi(\vec{\lambda}) + \int \ldots \int_{\{\vec{U} \mid U_{i^*} > u_i^*\}} G(\vec{\delta}(\vec{U})) d\pi(\vec{U}) \\ &= \Phi(\vec{\lambda}) + \int \ldots \int_{\{\vec{U} \mid U_{i^*} > u_i^*\}} G(\vec{\delta}(\vec{U})) d\pi(\vec{U}) \\ &\qquad + \int \ldots \int_{\{\vec{U} \mid U_{i^*} \leq u_i^*\}} G(\vec{0}) d\pi(\vec{U}) \end{aligned}$$

where the the final equality holds because $G(\vec{0}) = 0$. From Jensen's inequality and the fact that $G(\vec{\delta})$ is convex, we have:

$$\overline{h} \geq \Phi(\vec{\lambda}) + G\left(\int \ldots \int_{\{\vec{U} \mid U_{i^*} > u_i^*\}} \vec{\delta}(\vec{U}) d\pi(\vec{U}) + \vec{0}\right)$$

From (43) it follows that:

$$\begin{aligned} \overline{h} &\geq \Phi(\vec{\lambda}) + \Omega\left(\left|\left|\int \ldots \int_{\{\vec{U} \mid U_{i^*} > u_i^*\}} \vec{\delta}(\vec{U}) d\pi(\vec{U})\right|\right|^2\right) \\ &\geq \Phi(\vec{\lambda}) + \Omega\left(\left|\int \ldots \int_{\{\vec{U} \mid U_{i^*} > u_i^*\}} \delta_{i^*}(\vec{U}) d\pi(\vec{U})\right|^2\right) \\ &\geq \Phi(\vec{\lambda}) + \Omega\left(\frac{\theta_1^2\theta_2^4}{16^2(\mathbb{E}\{U_{i^*}\})^2}\right) \quad (46) \\ &\geq \Phi(\vec{\lambda}) + \Omega\left(\frac{N^2\theta_1^2\theta_2^4}{16^2(\overline{U}_{tot})^2}\right) \end{aligned}$$

where (46) follows by Lemma 7. This proves the result. □

Theorem 2 follows directly from Lemma 8. Indeed, if $\overline{h} - \Phi(\vec{\lambda}) \leq O(1/V)$, then from Lemma 8 we have: $O(1/V) \geq \Omega((1/\overline{U}_{tot})^2)$, and so $\overline{U}_{tot} \geq \Omega(\sqrt{V})$. By Little's Theorem for average delay: $\overline{W} = \overline{U}_{tot}/\sum_i \lambda_i$, and so $\overline{W} \geq \Omega(\sqrt{V})$.

## APPENDIX B — PROOF OF LEMMA 2

*Proof:* Suppose that the $\liminf$ conditions (20) are satisfied. Because the $\liminf$ of a sum is greater than or equal to the sum of the $\liminf$s, we have:

$$\liminf_{t \to \infty} \sum_{i=1}^{N} \left(\overline{r}_i(t) - \lambda_i - \epsilon[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)]\right) \geq 0 \quad (47)$$

Consider a particular queue $i$ and recall that $r_i(t)$ represents the transmission rate offered to the queue during slot $t$, and that $r_i(t) \leq \mu_{max}$ for all $t$. Note that $r_i(t)$ is exactly equal to the bits that depart during slot $t$ whenever $U_i(t) \geq \mu_{max}$. For simplicity, assume that all queues are initially empty. Because the bits that arrive during the first $t$ slots must be greater than or equal to the total departures, we have:

$$\begin{aligned} \sum_{\tau=0}^{t-1} A_i(\tau) &\geq \sum_{\tau=0}^{t-1} r_i(\tau) - \sum_{\tau=0}^{t-1} r_i(\tau) 1_i^E(\tau) \\ &\geq \sum_{\tau=0}^{t-1} r_i(\tau) - \sum_{\tau=0}^{t-1} \mu_{max} 1_i^E(\tau) \end{aligned}$$

where $1_i^E(t)$ is an indicator function that takes the value 1 if $U_i(t) < \mu_{max}$, and takes the value 0 else. Dividing both sides by $t$ and taking expectations yields:

$$\lambda_i \geq \overline{r}_i(t) - \mu_{max}\overline{\alpha}_i^E(t) \quad (48)$$

Using (48) in (47) yields:

$$\liminf_{t\rightarrow\infty}\sum_{i=1}^{N}\left(\mu_{max}\overline{\alpha}_i^E(t) - \epsilon[\overline{\alpha}_i^R(t) - \overline{\alpha}_i^L(t)]\right) \geq 0$$

Now note that for any functions $f(t), g(t)$ that satisfy $\liminf_{t\rightarrow\infty}[f(t) - g(t)] \geq 0$, we have $\limsup_{t\rightarrow\infty} f(t) \geq \limsup_{t\rightarrow\infty} g(t)$. Applying this fact to the above inequality proves the lemma. □

## Appendix C — Proof of Lemma 5a

*Proof:* Recall that $r_i(t) \leq \mu_{max}$ for all $t$. Define $\tilde{r}_i(t)\triangleq\min[r_i(t), U_i(t)]$, which is the actual amount of bits that depart from queue $i$ during slot $t$. Equation (1) thus becomes:

$$U_i(t+1) = U_i(t) - \tilde{r}_i(t) + A_i(t) \quad (49)$$

From (49) we have:

$$\begin{aligned} e^{\omega U_i(t+1)} &= e^{\omega U_i(t)}e^{-\omega(\tilde{r}_i(t)-A_i(t))} \\ &\leq e^{\omega(\mu_{max}+A_{max})} + e^{\omega U_i(t)}e^{-\omega(r_i(t)-A_i(t))} \end{aligned}$$

which follows because if $U_i(t) \geq \mu_{max}$ then $\tilde{r}_i(t) = r_i(t)$ and so the second term acts as a bound, while if $U_i(t) < \mu_{max}$ then the first term acts as a bound. Defining $\delta_i(t) = r_i(t) - A_i(t)$ and multiplying by $e^{-\omega Q}$ yields:

$$e^{\omega(U_i(t+1)-Q)} \leq e^{\omega(\mu_{max}+A_{max}-Q)} + e^{\omega(U_i(t)-Q)}e^{-\omega\delta_i(t)}$$

However, for any function $\delta_i(t)$ such that $|\delta_i(t)| \leq \delta_{max}$, we have by Taylor's theorem:

$$\begin{aligned} e^{-\omega\delta_i(t)} &\leq 1 - \omega\delta_i(t) + \tfrac{(\omega\delta_{max})^2}{2}e^{\omega\delta_{max}} \\ &\leq 1 - \omega\delta_i(t) + \tfrac{\omega\epsilon}{2} \end{aligned}$$

where the last inequality follows from the assumption that $\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon/\delta_{max}$. Thus:

$$\begin{aligned} G_i(\vec{U}(t))&\triangleq e^{\omega(U_i(t+1)-Q)} - e^{\omega(U_i(t)-Q)} \leq \\ &e^{\omega(\mu_{max}+A_{max}-Q)} - \omega e^{\omega(U_i(t)-Q)}(\delta_i(t) - \epsilon/2) \quad (50) \end{aligned}$$

where we have defined $G_i(\vec{U}(t))$ above for notational convenience. It can similarly be shown that:

$$\begin{aligned} K_i(\vec{U}(t))&\triangleq e^{\omega(Q-U_i(t+1))} - e^{\omega(Q-U_i(t))} \\ &\leq -\omega e^{\omega(Q-U_i(t))}(-\delta_i(t) - \epsilon/2) \quad (51) \end{aligned}$$

Combining (50) and (51), we have:

$$\begin{aligned} G_i(\vec{U}(t)) + K_i(\vec{U}(t)) &\leq e^{\omega(\mu_{max}+A_{max}-Q)} \\ &-\omega e^{\omega(U_i(t)-Q)}(\delta_i(t) - \tfrac{\epsilon}{2}) - \omega e^{\omega(Q-U_i(t))}(-\delta_i(t) - \tfrac{\epsilon}{2}) \\ &\leq e^{\omega(\mu_{max}+A_{max}-Q)} + \omega(\delta_{max} + \epsilon/2) \\ &-1_i^R(t)\omega e^{\omega(U_i(t)-Q)}(\delta_i(t) - \tfrac{\epsilon}{2}) \\ &-1_i^L(t)\omega e^{\omega(Q-U_i(t))}(-\delta_i(t) - \tfrac{\epsilon}{2}) \end{aligned}$$

where the final inequality follows because if $1_i^R(t) = 1$ then $1_i^L(t) = 0$, $U_i(t) \geq Q$, and $-\omega e^{\omega(Q-U_i(t))}(-\delta_i(t) - \frac{\epsilon}{2}) \leq \omega(\delta_{max} + \epsilon/2)$ (a similar inequality holds if $1_i^L(t) = 1$). Summing the final inequality over all $i$ and taking conditional expectations proves the lemma. □

## Appendix D – Implementation Complexity Examples

Here we illustrate the complexity involved in allocating power to minimize the weighted sum of transmission rates in the first part of the TOCA algorithm. Similar weighted minimizations also arise in the algorithms of [11], [14]-[20], and can often be computed quite easily. Consider first an example where the $N$ channels are orthogonal so that the transmission rate on each channel $i$ depends only on $P_i(t)$ and $S_i(t)$. In this case, we have:

$$\mu_i(\vec{P}(t), \vec{S}(t)) = f_i(P_i(t), S_i(t))$$

Further assume that each $f_i(P, S)$ function is concave in the power variable $P$, and that the constraint set $\Pi$ is given by:

$$\Pi = \left\{\vec{P} \mid \sum_{i=1}^{N} P_i \leq P_{max}, P_i \geq 0 \text{ for each } i \in \{1, \ldots, N\}\right\}$$

Suppose the cost function is given by $h(\vec{P}) = \sum_{i=1}^{N} P_i$. In this case, the TOCA algorithm observes the weights $W_i(t)$ and channel states $S_i(t)$ every timeslot $t$, and allocates a power vector $\vec{P}(t)$ that solves:

$$\begin{aligned} &\text{Maximize:} && \textstyle\sum_{i=1}^{N}[W_i(t)f_i(P_i, S_i(t)) - VP_i] \\ &\text{Subject to:} && (P_1, \ldots, P_N) \in \Pi \end{aligned}$$

This is a simple problem of maximizing a sum of concave separable functions subject to a simplex constraint, for which there are well known Lagrange multiplier solutions that equalize derivatives over all users that use non-zero power [34]. In particular, a solution can be obtained by first rank-ordering the $N$ channels in order of largest to smallest values of:

$$-V + W_i(t)\left.\frac{df_i(P_i, S_i(t))}{dP}\right|_{P_i=0}$$

For each integer $k \in \{0, \ldots, N\}$, let $\mathcal{L}_k$ be the collection of the first $k$ channels in this rank ordering (where $k = 0$ corresponds to the empty set). It can be shown that the channels with non-zero power will be those defined by one of the $\mathcal{L}_k$ sets. For a given $k$, a candidate power vector $\vec{P}[k]$ is computed as the vector that equalizes the derivatives of the $W_i(t)f_i(P_i, S_i(t)) - VP_i$ functions over each channel $i \in \mathcal{L}_k$, subject to the power constraint and to the constraint that $P_i = 0$ for all $i \notin \mathcal{L}_k$. If the derivatives cannot be equalized, then this sub-collection $\mathcal{L}_k$ is not the correct one to use. The optimal solution then compares the resulting solutions for each of the (at most $N$) valid sub-collections and chooses the best one. In the special case when we have:

$$f_i(P_i, S_i) = B_i\log(1 + S_iP_i)$$

then it is easy to show that for any set $\mathcal{L}_k$, the optimal power allocations are:

$$P_i[k] = \frac{B_iW_i(t)[P_{max} + \sum_{j\in\mathcal{L}_k} 1/S_j(t)]}{\sum_{j\in\mathcal{L}_k} B_jW_j(t)} - \frac{1}{S_i(t)}$$

assuming that these $P_i$ values are non-negative for all $i \in \mathcal{L}_k$ (else, it is not possible to equalize derivatives for this particular $\mathcal{L}_k$ set). In summary, the computation is quite simple and involves only computing $N$ values, comparing them, and choosing the best one.

Now consider the case of general $\vec{\mu}(\vec{P}, \vec{S})$ functions but when the set $\Pi$ constrains power vectors to only $O(N)$ choices. In this case, the solution is also quite simple: Just choose the vector $\vec{P}$ that maximizes:

$$-Vh(\vec{P}) + \sum_{i=1}^{N} W_i(t)\mu_i(\vec{P}, \vec{S}(t)) \quad (52)$$

over each of the $O(N)$ vector choices in $\Pi$. A natural example is a system where a fixed power $P_{max}$ can be allocated to at most one channel, and all other powers are set to 0. The set $\Pi$ in this case contains $N+1$ vectors: the all-zero vector together with the $N$ vectors that have a single entry equal to $P_{max}$ and all other entries equal to zero. If $\mu_i(\vec{P}, \vec{S}) = f_i(P_i, S_i)$ and $h(\vec{P}) = \sum_{i=1}^{N} P_i$, this amounts to serving the channel with the largest positive value of $-VP_{max} + W_i(t)f_i(P_{max}, S_i(t))$, and remaining idle to save power if none of these values are positive.

## REFERENCES


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

[2] 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.

[3] 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.

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

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

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

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

[8] 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.

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

[10] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, July 2006.

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

[12] L. Li and A. Goldsmith. Capacity and optimal resource allocation for fading broadcast channels: Part i: Ergodic capacity. *IEEE Trans. Inform. Theory*, pp. 1083-1102, March 2001.

[13] D. Tse and S. Hanly. Multi-access fading channels: Part 1: Polymatroid structure, optimal resource allocation, and throughput capacities. *IEEE Trans. on Information Theory*, vol. 44, no. 7, pp. 2796-2815, Nov. 1998.

[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, pp. 1936-1949, 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*, vol.40,.n.2, pp.251-260, Feb. 1995.

[17] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, vol. 39, no.2, pp.150-154, 2001.

[18] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multi-beam satellites with time varying channels. *IEEE Transactions on Networking*, vol. 11, no. 1, pp. 138-152, Feb. 2003.

[19] 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*, vol. 23, no. 1, pp. 89-103, January 2005.

[20] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, vol. 41, no. 4, pp. 451-474, March 2003.

[21] R. Cruz and A. Santhanam. Optimal routing, link scheduling, and power control in multi-hop wireless networks. *Proc. IEEE INFOCOM*, April 2003.

[22] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas. A framework for cross-layer design of energy-efficient communication with qos provisioning in multi-hop wireless networks. *INFOCOM*, 2004.

[23] E. M. Yeh and A. S. Cohen. Throughput optimal power and rate control for multiaccess and broadcast communications. *Proc. of the International Symposium on Information Theory*, June 2004.

[24] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. *Proc. IEEE INFOCOM*, March 2005.

[25] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation for wireless networks. *Proc. of the 39th Annual Allerton Conf. on Comm., Control, Comput.*, Oct. 2001.

[26] D. Julian, M. Chiang, D. O'Neill, and S. Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. *Proc. IEEE INFOCOM*, 2002.

[27] S. H. Low. A duality model of tcp and queue management algorithms. *IEEE Trans. on Networking*, vol. 11(4), August 2003.

[28] B. Krishnamachari and F. Ordonez. Analysis of energy-efficient, fair routing in wireless sensor networks through non-linear optimization. *IEEE Vehicular Technology Conf.*, Oct. 2003.

[29] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall. Energy-delay tradeoff for data collection in sensor networks. *To appear at ICC*, 2005.

[30] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *IEEE Transactions on Wireless Communications*, vol. 5, no.6, pp. 1506-1515, June 2006.

[31] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *Proc. IEEE INFOCOM*, March 2005.

[32] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, pp. 401-457, 2005.

[33] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions*. Studies in Advanced Mathematics, CRC Press, Boca Raton, Florida, 1992.

[34] D. P. Bertsekas. *Nonlinear Programming*. Athena, Belmont, MA, 1995.