Opportunism, Backpressure, and Stochastic Optimization with the Wireless Broadcast Advantage

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

***Abstract*— This paper provides a tutorial treatment of recent stochastic network optimization techniques, including Lyapunov network optimization, backpressure, and max-weight decision making. A new technique of *place holder bits* that improves delay for networking problems with general costs is also presented. An example application is given for the problem of energy-aware scheduling and routing in a wireless mobile network with channel errors and multi-receiver diversity. The Diversity Backpressure Routing algorithm (DIVBAR, Neely and Urgaonkar 2006, 2008) is illustrated and simulated in comparison to the Extremely Opportunistic Routing strategy (ExOR, Biswas and Morris 2005).**

I. INTRODUCTION

This paper provides a tutorial treatment of a recent theory of stochastic network optimization. This theory provides online control strategies for time varying networks with general classes of penalties, rewards, and utility functions. We first describe the general technique, which involves concepts of Lyapunov optimization, backpressure, and max-weight decision making. This material is taken largely from [1], where a more detailed treatment is given. We then provide a new extension of the theory that uses *place holder bits* to improve congestion and delay in networks with general costs [2]. Finally, we present an example application for a wireless mobile ad-hoc network with multi-receiver diversity. In this network, a single wireless transmission can be overheard by multiple receivers, each with different success probabilities. The throughput and energy-optimal Diversity Backpressure Routing algorithm (DIVBAR) from [3] [4] is presented for this context. An example simulation that compares DIVBAR to the Extremely Opportunistic Routing algorithm (ExOR) of [5] is given, along with a simulation of an enhanced E-DIVBAR algorithm that combines backpressure and shortest-path techniques for further delay improvement.

II. BACKGROUND

The technique of using Lyapunov drift to stabilize a multi-hop packet radio network was introduced by Tassiulas and Ephremides in [6], where backpressure routing and max-weight scheduling principles are derived. This technique has had wide success in developing dynamic algorithms for stability in computer networks [7] [8] [9] [10], wireless systems with opportunistic scheduling [11] [12] [13], and mobile ad-hoc networks [14]. Network stability can be achieved in these systems without requiring knowledge of traffic arrival rates or channel probabilities.

An extended Lyapunov drift theorem that allows joint stability and performance optimization is developed in [15] [16] [17] to treat utility-optimal flow control in stochastic networks. This is extended in [18] for energy optimization, and in [1] for a general class of stochastic penalty, reward, and utility metrics. Performance of these metrics can be pushed to within $O(1/V)$ of optimality (where V is a control parameter that can be made arbitrarily large), with a corresponding tradeoff in average network congestion and delay that is $O(V)$. This can be applied to treat metrics of reliability [20], distortion [2], revenue [21] [22], and to treat networks with non-ergodic, non-repeatable mobility [23]. This stochastic network optimization technique is related to classical duality theory and static convex programming (see [15] [14] [1] for discussion of the similarities, and [24] [25] for applications of backpressure to distributed computation). An interesting observation is that stochastic problems are often *easier to solve* than corresponding static problems, as they do not require convexity assumptions (the time average objective tends to "convexify" any non-convexities of the problem).

Related alternative approaches to stochastic network optimization are developed in [26] [27] [28]. Work in [26] considers scheduling in a wireless downlink (similar to [15] [16]), and uses a fluid model transformation to relate the stochastic problem to a static problem. Optimality can be approached with this method if the scaled queue backlogs are assumed to converge to appropriate Lagrange multipliers. Work in [27] treats networks with more general penalties and rewards, and uses a related fluid model transformation together with a primal-dual algorithm from static convex optimization theory. Work in [28] considers utility maximization via a stochastic gradient.

This paper focuses on the technique of [1]. This technique does not require fluid model transformations, and directly leads to the explicit performance-delay tradeoff $[O(1/V), O(V)]$. While this tradeoff holds for general networks and metrics, we note that it is not necessarily optimal for particular systems. Indeed, an improved energy-delay tradeoff of $[O(1/V), O(\sqrt{V})]$ is shown to be optimal for wireless downlinks [29] [30], and a utility-delay tradeoff of $[O(1/V), O(\log(V))]$ is shown to be optimal for systems with flow control [31] [32]. The work in [30] [31] [32] extends the stochastic network optimization

This material is supported in part by one or more of the following: the DARPA IT-MANET program grant W911NF-07-0028, the NSF grant OCE 0520324, the NSF Career grant CCF-0747525.

techniques of [1] to achieve these optimal delay tradeoffs, although the resulting dynamic algorithms are more complex and, for brevity, shall not be considered here.

III. Stochastic Network Optimization

A. A General Stochastic Network

Consider a stochastic queueing network that operates in slotted time with normalized slots $t \in \{0, 1, 2, \ldots\}$. Let $\mathcal{Q}$ represent the set of all queues in the network, and for each $q \in \mathcal{Q}$, let $Q_q(t)$ be the current backlog in queue q on slot t (assumed to be non-negative). The particular units of backlog, such as integer-valued units of packets or real-valued units of bits, can be assigned as appropriate for a given problem. Let $\boldsymbol{Q}(t) = (Q_q(t))_{q\in\mathcal{Q}}$ be the vector of all queue backlogs. This vector changes from slot to slot according to a partially controllable stochastic process, which depends on a *network state variable* $S(t)$ and a *control action* $I(t)$. The network state variable $S(t)$ represents an uncontrollable source of randomness associated with the network condition on slot t (such as node locations or channel conditions), and is assumed to evolve independently of any control actions taken in the network. For convenience, throughout this paper we assume that $S(t)$ is i.i.d. over slots and takes values on some abstract (possibly infinite) state space $\mathcal{S}$, although all of our results hold also when $S(t)$ evolves according to a finite state, irreducible (possibly non-aperiodic) Markov chain [12] [14] [20] [22]. At the beginning of slot t, the network controller observes the current $S(t)$ and the current $\boldsymbol{Q}(t)$, and makes a control action $I(t) \in \mathcal{I}_{S(t)}$, where $\mathcal{I}_{S(t)}$ is an abstract (possibly infinite) set of all feasible control options under network state $S(t)$. The control action determines how queue backlog is added, removed, or transferred from one queue to another in the network, and may involve a collection of flow control, routing, and transmission rate scheduling decisions.

On a given slot t, the transition probabilities associated with the queue vector are determined only by the current $\boldsymbol{Q}(t)$, $S(t)$, and $I(t) \in \mathcal{I}_{S(t)}$:

$$\boldsymbol{Q}(t) \overset{S(t),I(t)}{\longmapsto} \boldsymbol{Q}(t+1) \tag{1}$$

The equation (1) can be viewed as a Markov relation. However, traditional Dynamic Programming and Markov Decision Theory approaches to this problem involve offline computations with very high complexity and require full a-priori knowledge of all system statistics. Our approach is quite different and allows for simple online decision making, often without requiring the underlying probability distributions, provided that the system has a particular general structure to be made precise in the next section. To begin, we assume the queueing dynamics satisfy the following inequality for all $q \in \mathcal{Q}$:[1]

$$Q_q(t+1) \leq \max[Q_q(t) - R_q^{out}(I(t), S(t)), 0] + R_q^{in}(I(t), S(t)) \tag{2}$$

where $R_q^{out}(\cdot)$ represents the (potentially random) amount of backlog that can be shifted out of queue q under network state $S(t)$ and control action $I(t)$, and $R_q^{in}(\cdot)$ is the (potentially random) amount that can arrive (considering both exogenous arrivals from the transport layer, and possible endogenous arrivals from other queues). Such a structure is typical for single or multi-hop networks, and in particular it applies to all the networks treated in the references of this paper (see [1] for examples and more details on the queueing dynamics for networks). We further assume that the $R_q^{out}(\cdot)$ and $R_q^{in}(\cdot)$ values have bounded conditional second moments, regardless of the control action $I(t)$ taken on slot t.

B. Stochastic Penalties

We say that the queueing network is *strongly stable* if:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{Q_q(\tau)\right\} < \infty \text{ for all } q \in \mathcal{Q}$$

We consider the problem of minimizing the time average of some general network penalty function, subject to network stability and to a collection of additional time average penalty constraints. Specifically, let $p(I(t), S(t))$ be a random penalty process associated with the current network state $S(t)$ and control action $I(t)$ for slot t. Let $x_k(I(t), S(t))$ be additional random penalty processes for $k \in \{1, \ldots, K\}$. Note that $p(\cdot)$ and $x_k(\cdot)$ can be deterministic functions, in which case we do not require any convexity or continuity assumptions, or can be random functions.[2] We assume only that they are non-negative. Define the following time average expectations:

$$\begin{aligned} \overline{p} &\triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{p(I(\tau), S(\tau))\right\} \\ \overline{x}_k &\triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{x_k(I(\tau), S(\tau))\right\} \end{aligned}$$

Let $\overline{\boldsymbol{x}} \triangleq (\overline{x}_1, \ldots, \overline{x}_K)$. Our objective is to design a network control algorithm that achieves the following:

$$\begin{aligned} &\text{Minimize:} && \overline{p} \\ &\text{Subject to:} \quad 1) && \overline{\boldsymbol{x}} \leq \boldsymbol{x}_{av} \\ & \quad\quad\quad\quad\quad\;\; 2) && \text{Network Stability} \end{aligned}$$

where $\boldsymbol{x}_{av}$ is a vector of positive average penalty constraints. For example, the penalty function $p(\cdot)$ might represent the sum power expended in the network, and the additional penalties $x_k(\cdot)$ might represent powers expended by each particular node of the network, which enforces average power constraints at each node [18]. Alternatively, some of the penalties might represent signal processing distortions [2]. This framework can be used to treat *rewards* $g(I(t), S(t))$ by defining a penalty process as a negative reward process: $p(I(t), S(t)) = g_{max} - g(I(t), S(t))$, assuming that rewards are upper bounded by a finite constant g_{max}.

[1] The inequality in (2) is typical, as the $R_q^{in}(\cdot)$ and $R_q^{out}(\cdot)$ functions should depend only on $I(t)$ and $S(t)$ and not on queue backlog. Hence, these functions represent transmission rates *offered* by the network on slot t, although these rates may not fully be utilized if there is little or no backlog to send (see [1]). For example, we might have $R_q^{in}(t) = A_q(t) + \sum_{a\in\Omega_q(t)} R_a^{out}(t)$, where $A_q(t)$ is a random amount of exogenous arrivals, and $\Omega_q(t)$ is the set of all queues a that are currently transmitting to q.

[2] The work in [1] treats all penalties as deterministic functions, although random functions $p(I(t), S(t))$ can be treated in this way by using a corresponding deterministic function $\hat{p}(I(t), S(t)) \triangleq \mathbb{E}\left\{p(I(t), S(t)) \mid I(t), S(t)\right\}$.

The above problem is in fact a special case of the more general stochastic network optimization framework developed in [1]. Indeed, the work in [1] considers the problem with K different penalties $\overline{\boldsymbol{x}} = (\overline{x}_1, \ldots, \overline{x}_K)$ and M different constraints, where the objective is to minimize $f_0(\overline{\boldsymbol{x}})$ subject to constraints $f_m(\overline{\boldsymbol{x}}) \leq b_m$ for $m \in \{1, \ldots, M\}$, where $f_0(\cdot), f_1(\cdot), \ldots, f_M(\cdot)$ are general convex multi-variable functions that are non-decreasing in each variable. The objective of minimizing a nonlinear function of a time average is quite different than minimizing the time-average itself, and can be solved by introducing *auxiliary variables* and *flow state queues*, see [1] and [16] [17].

C. Virtual Queues and Lyapunov Drift

For each of the penalty constraints $\overline{x}_k \leq x_{k,av}$ (for $k \in \{1, \ldots, K\}$), we create a *virtual queue* $Z_k(t)$ with update equation as follows:

$$Z_k(t+1) = \max[Z_k(t) - x_{k,av}, 0] + x_k(I(t), S(t)) \quad (3)$$

This queue is implemented only in software, and has a virtual constant service rate of $x_{k,av}$ and a virtual input process equal to the penalty $x_k(I(t), S(t))$ incurred by our control action on slot t. Ensuring that such a queue is strongly stable implies that the $\limsup$ time average expectation of the input process is less than or equal to the virtual service rate $x_{k,av}$, and hence ensures that the time average penalty constraint $\overline{x}_k \leq x_{k,av}$ is satisfied [18]. This transforms the constrained optimization problem into a pure problem of stabilizing (virtual and actual) queues while minimizing the time average objective $\overline{p}$. Such virtual queues can be viewed as a stochastic version of a Lagrange multiplier, and were introduced in [18] [19] to maximize network throughput subject to average power constraints.

Let $\boldsymbol{Z}(t) = (Z_1(t), \ldots, Z_K(t))$ be the vector of virtual queues. Recall that $\boldsymbol{Q}(t)$ is the vector of actual queues, and define $\boldsymbol{\Theta}(t) = (\boldsymbol{Q}(t), \boldsymbol{Z}(t))$ as the combined queue vector. Define the following quadratic Lyapunov functions:

$$L_Z(\boldsymbol{\Theta}(t)) \triangleq \tfrac{1}{2}\textstyle\sum_{k=1}^K Z_k(t)^2 \ , \ L_Q(\boldsymbol{\Theta}(t)) \triangleq \tfrac{1}{2}\sum_{q\in\mathcal{Q}} Q_q(t)^2$$
$$L(\boldsymbol{\Theta}(t)) \triangleq L_Z(\boldsymbol{\Theta}(t)) + L_Q(\boldsymbol{\Theta}(t))$$

Define the *one-step conditional Lyapunov drift* $\Delta_Z(\boldsymbol{\Theta}(t))$, $\Delta_Q(\boldsymbol{\Theta}(t))$, and $\Delta(\boldsymbol{\Theta}(t))$ as follows:

$$\begin{aligned}
\Delta_Z(\boldsymbol{\Theta}(t)) &\triangleq \mathbb{E}\left\{L_Z(\boldsymbol{\Theta}(t+1)) - L_Z(\boldsymbol{\Theta}(t)) \mid \boldsymbol{\Theta}(t)\right\} \\
\Delta_Q(\boldsymbol{\Theta}(t)) &\triangleq \mathbb{E}\left\{L_Q(\boldsymbol{\Theta}(t+1)) - L_Q(\boldsymbol{\Theta}(t)) \mid \boldsymbol{\Theta}(t)\right\} \\
\Delta(\boldsymbol{\Theta}(t)) &\triangleq \Delta_Z(\boldsymbol{\Theta}(t)) + \Delta_Q(\boldsymbol{\Theta}(t)) \quad (4)
\end{aligned}$$

Squaring the equation (3) and taking expectations, it can be shown via a standard argument that $\Delta_Z(\boldsymbol{\Theta}(t)) \leq \overline{\Delta}_Z(\boldsymbol{\Theta}(t))$, where (see, for example, [1]):

$$\overline{\Delta}_Z(\boldsymbol{\Theta}(t)) \triangleq B_Z - \sum_{k=1}^K Z_k(t)\mathbb{E}\left\{x_{k,av} - x_k(I(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \quad (5)$$

where B_Z is a finite constant that depends on the maximum second moment of the penalty processes, assumed to be upper bounded regardless of the control action $I(t)$. Similarly, squaring (2) yields $\Delta_Q(\boldsymbol{\Theta}(t)) \leq \overline{\Delta}_Q(\boldsymbol{\Theta}(t))$, where:

$$\begin{aligned}
\overline{\Delta}_Q(\boldsymbol{\Theta}(t)) \triangleq B_Q &- \sum_{q\in\mathcal{Q}} Q_q(t)\mathbb{E}\left\{R_q^{out}(I(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \\
&+ \sum_{q\in\mathcal{Q}} Q_q(t)\mathbb{E}\left\{R_q^{in}(I(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \quad (6)
\end{aligned}$$

where B_Q is a finite constant that depends on the maximum second moments of the $R_q^{out}(\cdot)$ and $R_q^{in}(\cdot)$ processes.

D. Lyapunov Optimization and the Max-Weight Algorithm

Here we present the basic Lyapunov optimization theorems required for our stochastic analysis (see proofs in [1]). Let $\boldsymbol{\Theta}(t) = (\Theta_q(t))_{q\in\mathcal{D}}$ be any stochastic vector of queue backlogs that evolves in discrete time, where $\mathcal{D}$ is an index set for the queues. Let $L(\boldsymbol{\Theta}(t))$ be a non-negative function of the queue backlog vector, and define $\Delta(\boldsymbol{\Theta}(t)) \triangleq \mathbb{E}\left\{L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) \mid \boldsymbol{\Theta}(t)\right\}$.

Theorem 1: (Lyapunov drift [1]) Let $f(t)$ and $g(t)$ be any stochastic processes, possibly related to $\boldsymbol{\Theta}(t)$. If for all t and all $\boldsymbol{\Theta}(t)$ we have:

$$\Delta(\boldsymbol{\Theta}(t)) \leq \mathbb{E}\left\{g(t) - f(t) \mid \boldsymbol{\Theta}(t)\right\}$$

then for all t we have:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{f(\tau)\right\} \leq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g(\tau)\right\} + \frac{\mathbb{E}\left\{L(\boldsymbol{\Theta}(0))\right\}}{t} \ \square$$

Now let $x(t)$ be any (possibly negative) stochastic penalty process associated with the system, and let x^* represent a "target" penalty. Theorem 1 implies the following.

Theorem 2: (Lyapunov Optimization [1] [18] [15]) If there are constants $B \geq 0$, $\epsilon \geq 0$, $V \geq 0$ such that the following holds for all t and all $\boldsymbol{\Theta}(t)$:

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{x(t) \mid \boldsymbol{\Theta}(t)\right\} \quad \leq \quad B - \epsilon\sum_{q\in\mathcal{D}}\Theta_q(t) + Vx^*$$

then:

$$\overline{x} \leq x^* + B/V$$
$$\limsup_{t\to\infty} \tfrac{1}{t}\textstyle\sum_{\tau=0}^{t-1}\sum_{q\in\mathcal{D}}\mathbb{E}\left\{\Theta_q(\tau)\right\} \leq \frac{B+V(x^*-\overline{x}_{inf})}{\epsilon}$$

where $\overline{x}$ is the $\limsup$ time average expectation of $x(t)$ and $\overline{x}_{inf}$ is the the $\liminf$ time average expectation of $x(t)$. $\square$

Thus, if the condition of Theorem 2 holds, the time average penalty $\overline{x}$ is at most B/V beyond the target x^*. If V is a control parameter that can be chosen independently of B, then B/V can be made arbitrarily small, with a corresponding linear tradeoff in time average congestion. For our system, we can define $x(t) = p(I(t), S(t))$. Hence, Theorem 2 suggests the control strategy that observes $S(t)$, $\boldsymbol{\Theta}(t)$ on each slot t and chooses $I(t) \in \mathcal{I}_{S(t)}$ to minimize (subject to a given $\boldsymbol{\Theta}(t)$):

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{p(I(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \quad (7)$$

where the above expectation includes averaging over all possible $S(t)$ values. However, given that we can first observe $S(t)$, the optimal policy is to greedily minimize the "inside" of the expectation subject to $S(t)$ knowledge (recall that $\mathbb{E}\left\{X\right\} =$

$\mathbb{E}_Y\{\mathbb{E}\{X \mid Y\}\}$ for any random variables X and Y).[3] Hence, we do not require knowledge of the $S(t)$ probabilities. Actually, achieving the minimum of (7) to within any additive constant will suffice. Using $\Delta(\boldsymbol{\Theta}(t))$ as defined in the previous subsection and noting that $\Delta(\boldsymbol{\Theta}(t)) \leq \overline{\Delta}_Z(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q(\boldsymbol{\Theta}(t))$ leads to the following:

The Generalized Max-Weight Policy: Every timeslot t, observe $S(t)$ and $\boldsymbol{\Theta}(t)$, and choose $I(t) \in \mathcal{I}_{S(t)}$ to greedily minimize (subject to a given $\boldsymbol{\Theta}(t)$):

$$\overline{\Delta}_Z(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{p(I(t), S(t)) \mid \boldsymbol{\Theta}(t)\} \quad (8)$$

with $\overline{\Delta}_Z(\cdot)$, $\overline{\Delta}_Q(\cdot)$ defined in (5), (6). Then update the virtual queues $Z_k(t)$ for each $k \in \{1, \ldots, K\}$ according to (3) (the actual queue dynamics proceed according to (2)). Using the definitions of $\overline{\Delta}_Z(\boldsymbol{\Theta}(t))$ and $\overline{\Delta}_Q(\boldsymbol{\Theta}(t))$, the minimization of (8) is equivalent to choosing $I(t) \in \mathcal{I}_{S(t)}$ to minimize:

$$V\hat{p}(I(t), S(t)) + \sum_{k=1}^{K} Z_k(t)\hat{x}_k(I(t), S(t))$$
$$- \sum_{q \in \mathcal{Q}} Q_q(t)[\hat{R}_q^{out}(I(t), S(t)) - \hat{R}_q^{in}(I(t), S(t))] \quad (9)$$

where $\hat{p}(I(t), S(t)) \triangleq \mathbb{E}\{p(I(t), S(t)) \mid I(t), S(t)\}$, and where $\hat{x}_k(I(t), S(t))$, $\hat{R}_q^{out}(I(t), S(t))$, and $\hat{R}_q^{in}(I(t), S(t))$ are similarly defined as the conditional expectations given $I(t)$ and $S(t)$. These functions of $I(t)$ and $S(t)$ are assumed to be known. Note that for a given slot t, the observed network state $S(t)$ and queue backlogs $\boldsymbol{Q}(t)$ and $\boldsymbol{Z}(t)$ act as known constants in the minimization of (9). Hence, this minimization does not require knowledge of the probability distribution of $S(t)$. Further, other than computing expectations associated with the $\hat{p}(\cdot)$, $\hat{x}(\cdot)$, $\hat{R}_q^{out}(\cdot)$, and $\hat{R}_q^{in}(\cdot)$ functions, it does not require knowledge of any other statistics that affect the network evolution, such as traffic arrival rates.[4]

IV. Algorithm Performance

Consider the following class of control policies that choose $I(t) \in \mathcal{I}_{S(t)}$ as a stationary and potentially randomized function only of the current network state $S(t)$, and independent of queue backlog. We call such policies *S-only policies*. Note that, by the law of large numbers and the assumption that $S(t)$ is i.i.d. over slots, this class of policies leads to well defined system time averages. Let p^* be the infimum time average penalty $\overline{p}$ incurred over the class of S-only algorithms $I^*(t)$ that satisfy the following:

$$\begin{aligned} \mathbb{E}\{p(I^*(t), S(t))\} &= \overline{p} \quad (10)\\ \mathbb{E}\{x_k(I^*(t), S(t))\} &\leq x_{k,av}\ ,\ \forall k \in \{1, \ldots, K\} \\ \mathbb{E}\left\{R_q^{out}(I^*(t), S(t))\right\} &\geq \mathbb{E}\left\{R_q^{in}(I^*(t), S(t))\right\}\ ,\ \forall q \in \mathcal{Q} \end{aligned}$$

[3]Specifically, $\mathbb{E}\{X \mid \boldsymbol{\Theta}\} = \mathbb{E}_{S|\boldsymbol{\Theta}}\{\mathbb{E}\{X \mid \boldsymbol{\Theta}, S\}\}$ for any random variables X, $\boldsymbol{\Theta}$, and S. If $X = X(I, S)$ (i.e., a random function that depends on S and on an additional, possibly random, input I), then $\mathbb{E}\{X(I,S) \mid \boldsymbol{\Theta}\}$ is minimized by the control input that chooses I as the function of S and $\boldsymbol{\Theta}$ that minimizes $\mathbb{E}\{X(I,S) \mid \boldsymbol{\Theta}, S\}$.

[4]The $\hat{R}_q^{in}(\cdot)$ function may include an uncontrollable exogenous arrival rate term λ_q that is not needed to minimize (9), or may include a flow control term that can be minimized with knowledge of the new arrivals $A_q(t)$ observed on slot t, without requiring the distribution of $A_q(t)$, see [1]. In the latter case, $A_q(t)$ can be viewed as a part of the current network state $S(t)$.

We are implicitly assuming the above time average constraints are *feasible*, and hence p^* is the infimum time average penalty over all feasible S-only algorithms. While our Generalized Max-Weight Policy is not an S-only policy, we shall measure its optimality with respect to p^*. For many network objectives, such as throughput, average power, fairness, etc., optimality can be *attained* over the class of S-only algorithms, and hence p^* is the optimal performance over all control algorithms [14] [18]. For simplicity, we assume that the infimum p^* is achievable, in the sense of satisfying the constraints of (10), by a particular S-only policy $I^*(t)$.[5] We must further make the following *slackness assumption*: There exists a constant $\epsilon > 0$ together with an S-only policy $I^s(t)$ such that:[6]

$$\mathbb{E}\{x_k(I^s(t), S(t))\} + \epsilon \leq x_{k,av}\ ,\ \forall k \in \{1, \ldots, K\}$$
$$\mathbb{E}\left\{R_q^{out}(I^s(t), S(t))\right\} \geq \epsilon + \mathbb{E}\left\{R_q^{in}(I^s(t), S(t))\right\}\ ,\ \forall q \in \mathcal{Q}$$

Theorem 3: (Performance Theorem [1] [18]) Suppose there is a value $p_{max} < \infty$ so that $0 \leq p(I(t), S(t)) \leq p_{max}$ always. Under the Generalized Max-Weight Policy with any control parameter $V \geq 0$, the network is stable, the constraints $\overline{\boldsymbol{x}} \leq \boldsymbol{x}_{av}$ are satisfied, the $\limsup$ time average penalty $\overline{p}$ satisfies:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{p(I(\tau), S(\tau))\} \leq p^* + (B_Z + B_Q)/V \quad (11)$$

and the time average queue backlogs satisfy:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{q\in\mathcal{Q}} \mathbb{E}\{Q_q(\tau)\} + \sum_{k=1}^{K}\mathbb{E}\{Z_k(\tau)\}\right] \leq (B_Z + B_Q + Vp_{max})/\epsilon \quad (12)$$

Thus, the V parameter can be chosen as desired to affect the $[O(1/V), O(V)]$ performance-congestion tradeoff.

Proof: Let $I^*(t)$ be the S-only policy that achieves all feasibility constraints (10) and yields $\mathbb{E}\{p(I^*(t), S(t))\} = p^*$. Let $I^{MW}(t)$ denote the Max-Weight policy. Let $\Delta^{MW}(\boldsymbol{\Theta}(t))$ denote the Lyapunov drift under the Max-Weight policy $I^{MW}(t)$ (as defined in (4)), and let $\overline{\Delta}_Z^{MW}(\boldsymbol{\Theta}(t))$ and $\overline{\Delta}_Q^{MW}(\boldsymbol{\Theta}(t))$ denote the corresponding drift bounds as defined in Section III-C (equations (5) and (6)). Let $\overline{\Delta}_Z^*(\boldsymbol{\Theta}(t))$ and $\overline{\Delta}_Q^*(\boldsymbol{\Theta}(t))$ be the corresponding drift bounds for the $I^*(t)$ policy. Because $I^{MW}(t)$ (by definition) minimizes (8), we have:

$$\Delta^{MW}(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{p(I^{MW}(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \leq$$
$$\overline{\Delta}_Z^*(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q^*(\boldsymbol{\Theta}(t)) + V\mathbb{E}\{p(I^*(t), S(t)) \mid \boldsymbol{\Theta}(t)\} \quad (13)$$

Now note that because $I^*(t)$ is S-only and $S(t)$ is i.i.d. over slots, $I^*(t)$ is independent of queue backlogs $\boldsymbol{\Theta}(t)$, and so:

$$\mathbb{E}\{x_k(I^*(t), S(t)) \mid \boldsymbol{\Theta}(t)\} = \mathbb{E}\{x_k(I^*(t), S(t))\} \leq x_{k,av}$$

where the final inequality above is due to the inequality constraints of (10). Thus, plugging the inequality constraints

[5]Else, we can prove the same performance results with a limiting argument using a sequence of policies $I_n^*(t)$ that converge to the performance p^*.

[6]This single ϵ-slackness assumption that holds for all constraints simultaneously can be relaxed by replacing it with multiple ϵ-slackness assumptions that hold for each constraint individually while all other constraints are satisfied with possible 0-slackness.

of (10) into the definitions of $\overline{\Delta}_Z^*(\boldsymbol{\Theta}(t))$ and $\overline{\Delta}_Q^*(\boldsymbol{\Theta}(t))$ given in (5) and (6), it is a simple matter to verify:

$$\overline{\Delta}_Z^*(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q^*(\boldsymbol{\Theta}(t)) \leq B_Z + B_Q$$

Define $B \triangleq B_Z + B_Q$. Using the above inequality in (13) yields:

$$\Delta^{MW}(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left\{p(I^{MW}(t), S(t)) \mid \boldsymbol{\Theta}(t)\right\} \leq B + Vp^*$$

This is in the exact form for using the Lyapunov Optimization Theorem (Theorem 2) with $\epsilon = 0$, $x(t) = p(I^{MW}(t), S(t))$, $x^* = p^*$, which proves the performance bound (11).

Similarly, to derive the time average bound on queue backlog, we can use the S-only policy $I^s(t)$ that yields the slackness constraints. Note that the inequality (13) holds equally if we plug the $I^s(t)$ policy into each of the three terms on the right hand side, rather than the $I^*(t)$ policy (as $I^{MW}(t)$ minimizes the right hand side over any other control action). It is again a simple matter to verify from the slackness constraints and the definitions of $\overline{\Delta}_Z^s(\boldsymbol{\Theta}(t))$ and $\overline{\Delta}_Q^s(\boldsymbol{\Theta}(t))$ (representing drift bounds under the $I^s(t)$ policy) that:

$$\overline{\Delta}_Z^s(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q^s(\boldsymbol{\Theta}(t)) \leq B - \epsilon\left[\sum_{k=1}^K Z_k(t) + \sum_{q\in\mathcal{Q}} Q_q(t)\right]$$

which can be plugged into the right hand side of (13) to yield the resulting bound (12) upon application of Theorem 2. □

A. On Distributed Implementation and Queue Groupings

We note that distributed implementation of the Generalized Max-Weight policy is possible when channels are orthogonal and/or transmission options are randomized [14] [18] [23]. Constant-factor throughput and performance results, where stability regions are reduced by a constant factor, are guaranteed when a distributed algorithm comes within a constant factor of maximizing (9). This is shown in [15] for stability, in [1] [17] for joint stability and throughput-utility maximization, and in [3] for stability and power minimization.

A different approach to stable scheduling via *maximal matchings* can also be shown to yield constant-factor results [33] [34] [35] [36]. Maximal scheduling is often analyzed using queue-grouped Lyapunov functions that can provide tighter delay guarantees for reduced throughput regions [37] [38]. Queue groupings can also be used to prove order-optimal delay for the full-throughput max-weight scheduling algorithm in wireless downlinks with ON/OFF channels [39].

B. On Delay Reduction and Worst Case Backlog

We note that network congestion and delay can often be improved by directly minimizing the drift expression (7), rather than its upper bound that uses $\Delta(\boldsymbol{\Theta}(t)) \leq \overline{\Delta}_Z(\boldsymbol{\Theta}(t)) + \overline{\Delta}_Q(\boldsymbol{\Theta}(t))$. The performance bounds of Theorem 3 hold also in this case (the proof is exactly the same). However, congestion is typically better because the drift expression (7) uses the actual amount of backlog transferred (often being a minimum of the offered transfer rate and the actual backlog available), while the drift bound uses only the offered transfer rate (the offered rate and the actual amount transferred are the same when queue backlog is sufficiently large).

Alternatively, one can minimize any expression that differs by only an additive constant from either the drift expression (7) or its upper bound. This simply adds an additive constant to the congestion bound of Theorem 3. Such a scenario arises if backlog updates are delayed so that actual queue backlogs differ by a constant from those used in the control decision [1]. Queue backlogs can also be augmented by constants to achieve additional desired behavior while maintaining stability. For example, delay in multi-hop networks can often be improved by adding a shortest path bias term to each queue backlog. This is used in the Enhanced Dynamic Routing and Power Allocation (EDRPC) algorithm of [14] and the Enhanced Diversity Backpressure Routing (E-DIVBAR) algorithm of [3], where simulations demonstrate considerable delay improvement (see also Section VI). We further note that in systems with flow control, the max-weight policy (or modifications that add a constant to the backpressure value) often allow worst case queue backlog guarantees [18] [23] [21].

V. Place Holder Bits for Delay Improvement

Note that the initial condition of the queues does not affect the time average penalty or queue backlog bounds of Theorem 2 or Theorem 3. Suppose now that the network dynamics under the Max-Weight policy (9) satisfy the following property:

Property 1: There is a non-negative queue backlog vector $\boldsymbol{\Theta}_0 \triangleq (\boldsymbol{Q}_0; \boldsymbol{Z}_0)$ such that if $\boldsymbol{\Theta}(0) \geq \boldsymbol{\Theta}_0$, then $\boldsymbol{\Theta}(t) \geq \boldsymbol{\Theta}_0$ for all $t \geq 0$, where inequality is taken entrywise. □

If Property 1 holds for a non-zero vector $\boldsymbol{\Theta}_0 = (\boldsymbol{Q}_0, \boldsymbol{Z}_0)$, we initialize the virtual queues to $\boldsymbol{Z}(0) = \boldsymbol{Z}_0$, and we initialize the actual queues so that there is no *actual* data initially in these queues, but there is an amount $\boldsymbol{Q}_0$ of *fake data* or *place holder data*. Specifically, suppose the actual queue backlog is given by $\hat{\boldsymbol{Q}}(t)$, and define:

$$\boldsymbol{Q}(t) \triangleq \hat{\boldsymbol{Q}}(t) + \boldsymbol{Q}_0$$

The Max-Weight algorithm uses the $\boldsymbol{Q}(t)$ values in its decisions (9). However, whenever it makes a transmission decision, it transmits only *actual data* (not fake data), whenever possible. The resulting sample path of $\boldsymbol{Q}(t)$ is equivalent to that of a system with initial condition $\Theta(0) = (\boldsymbol{Q}_0; \boldsymbol{Z}_0)$, and hence yields the exact same time average bounds on penalty $\overline{p}$ and queue backlog $\overline{Q}_q$ as specified in Theorem 3. Further, because Property 1 holds, we will have $(\boldsymbol{Q}(t); \boldsymbol{Z}(t)) \geq (\boldsymbol{Q}_0; \boldsymbol{Z}_0)$ for all t, and so the actual queue backlog will *never* drop low enough to require transmission of fake data. It follows that the *actual* backlog vector $\hat{\boldsymbol{Q}}(t)$ will be exactly an amount $\boldsymbol{Q}_0$ lower than $\boldsymbol{Q}(t)$ for every instant of time t, which reduces the average queue backlog by exactly this amount, without any change in the time average penalty $\overline{p}$. Thus, this fake data is simply acting as a *place holder* to properly affect the performance optimization.

We recently developed this concept of *place holder bits* in [2] for a single-hop system (the reader is referred to more specific examples given there), although it applies equally well in this potentially multi-hop setting. For cost minimization problems, the Max-Weight algorithm of (9) typically *does* satisfy Property 1 for a non-zero vector $\boldsymbol{\Theta}_0$. Indeed, from

(9) it can be seen that a queueing node q will typically not transmit any data out unless the amount of backlog it currently has, multiplied by the amount it will transmit out, exceeds a threshold equal to the penalty incurred by this transmission multiplied by V. The backlog reduction of place holder data can be quite significant, and can be successfully applied to energy minimization algorithms [18] [3] and other related cost minimization algorithms to improve delay.

VI. Multi-Receiver Diversity and DIVBAR

This stochastic optimization technique is applied in our prior work [3] to minimize average power in a multi-hop wireless mobile network with multi-receiver diversity. Specifically, a packet transmission at a given node can potentially be received by a set of neighbors, with heterogeneous success probabilities that depend on the current topology state $S(t)$. All successful receivers provide ACK feedback, and the transmitting node sends a final message to indicate which one (if any) should take responsibility for the packet. The detailed network control variables are given in [3], where the Max-Weight policy is applied to create a Diversity Backpressure Routing Algorithm (DIVBAR) that is throughput and energy optimal.

In the case of a single commodity, orthogonal channels, and no power optimization (i.e., $V = 0$), DIVBAR reduces to having each node transmit whenever possible, and shifting forwarding responsibility to the successful receiver with the largest *differential backlog* (being the difference in queue backlog at the transmitter and receiver), retaining the packet at the transmitter if no receiver has a positive differential backlog. If power minimization is included ($V > 0$), the algorithm is more complex and requires instantaneous success probabilities of neighbor nodes under a given topology state, but does not require knowledge of traffic rates or of how the topology changes from one slot to the next. If at most one packet can be transmitted on a slot, and each transmission requires 1 unit of power, it can be shown that DIVBAR satisfies Property 1 with $\boldsymbol{Q}_0 = (\max[\lfloor V \rfloor - 1, 0])_{q \in \mathcal{Q}}$ (in units of packets).

Figure 1 illustrates a cell-partitioned mobile ad-hoc network, from “Example 2” of [3]. There are 9 source nodes: 3 stationary, 3 locally mobile, and 3 fully mobile. The locally mobile nodes are restricted to move in the shaded cells while the fully mobile nodes can move anywhere in the network. There are 2 stationary sinks and packets can be delivered to either of them (thus, this is a single commodity scenario). The mobile nodes perform a Markovian random walk over their respective regions (choosing each slot to either remain in their current cell or to move to an adjacent cell). We assume at most one packet can be transmitted per cell, but transmissions in different cells are orthogonal. Figure 2 illustrates the performance of DIVBAR for this example (see the detailed model description in [3]). This figure also includes new data that demonstrates the benefits of backlog reduction achieved by using place holder packets, at no additional energy cost (as shown in Figure 3).

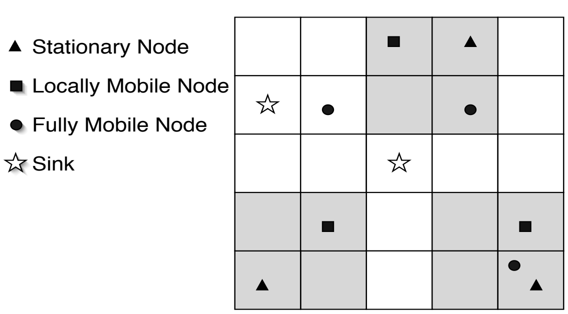

Fig. 1. A mobile network with two sinks and heterogeneous mobility [3].

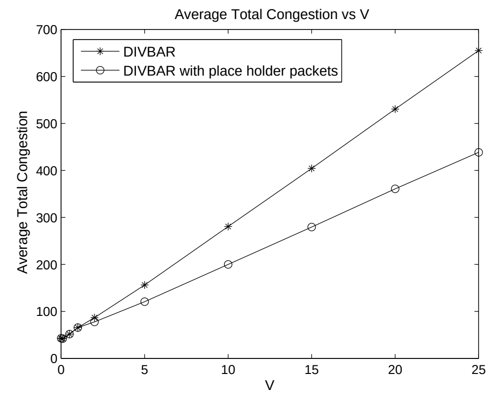

Fig. 2. Average total congestion (i.e., total packets summed over all queues in the network) versus V under DIVBAR with and without place holder packets on the example mobile network of Fig. 1.

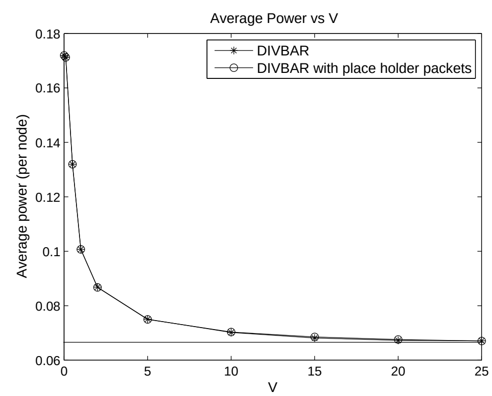

Fig. 3. Average power versus V under DIVBAR with and without place holder packets on the example mobile network of Fig. 1. The two curves are almost identical, demonstrating no loss of energy efficiency with place holder packets.

Figure 4 treats a different network: the 2-commodity static network from “Example 1” in [3] (figure omitted for brevity, see [3]). In the multi-commodity case, DIVBAR selects the optimal commodity to transmit via a rank-ordering that depends on the current backlog and reception success probabilities of neighboring nodes. Such a commodity distinction is required for optimality, and is quite different from other diversity algorithms (such as ExOR) that do not optimize over commodity selection. We consider only stability here, without power optimization (so that $V = 0$), and compare to the ExOR algorithm of [5] that uses the heuristic of shifting packets to the successful receiver that is “closest” to the destination, where proximity is measured with respect to a shortest path metric.

Unlike DIVBAR, the ExOR algorithm is not throughput-optimal, and in this example it becomes unstable at rates that are a little more than half the distance to the capacity region boundary. However, its shortest-path properties allow ExOR to have lower delay in the low rate region. A combined strategy E-DIVBAR that uses backpressure and shortest path in the forwarding decision is also illustrated (this is related to the mixed backpressure/shortest path enhancement in [14]). It is shown in [3] that E-DIVBAR is also throughput optimal, and we see in this example that it has lower delay than both DIVBAR and ExOR across all input rates.

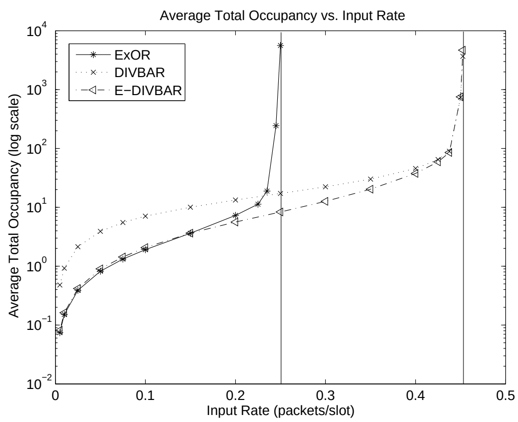

Fig. 4. A comparison of DIVBAR, ExOR, and E-DIVBAR for the 2-commodity static network example of [3].

REFERENCES

[1] 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.

[2] M. J. Neely and A. Sharma. Dynamic data compression with distortion constraints for wireless transmission over a fading channel. *arXiv:0807.3768v1*, July 24, 2008.

[3] M. J. Neely and R. Urgaonkar. Optimal backpressure routing in wireless networks with multi-receiver diversity. *Ad Hoc Networks (Elsevier)*, 2008, doi:10.1016/j.adhoc.2008.07.009.

[4] M. J. Neely. Optimal backpressure routing for wireless networks with multi-receiver diversity. *Proc. of Conf. on Information Sciences and Systems (CISS), Invited paper on Optimization of Communication Networks*, March 2006.

[5] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. *Proc. of Sigcomm*, 2005.

[6] 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.

[7] 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.

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

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

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

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

[12] L. Tassiulas. Scheduling and performance limits of networks with constantly changing topology. *IEEE Trans. on Inf. Theory*, May 1997.

[13] 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.

[14] 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.

[15] 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.

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

[17] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396-409, April 2008.

[18] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, July 2006.

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

[20] R. Urgaonkar and M. J. Neely. Opportunistic scheduling with reliability guarantees in cognitive radio networks. *Proc. IEEE INFOCOM*, April 2008.

[21] M. J. Neely. Optimal pricing in a free market wireless network. *Proc. IEEE INFOCOM*, May 2007.

[22] L. Huang and M. J. Neely. The optimality of two prices: Maximizing revenue in a stochastic network. *Proc. 45th Allerton Conf. on Communication, Control, and Computing*, Sept. 2007.

[23] M. J. Neely and R. Urgaonkar. Cross layer adaptive control for wireless mesh networks. *Ad Hoc Networks (Elsevier)*, vol. 5, no. 6, pp. 719-743, August 2007.

[24] M. J. Neely. Distributed and secure computation of convex programs over a network of connected processors. *DCDIS Conf., Guelph, Ontario*, July 2005.

[25] M. Andrews, K. Jung, and A. Stolyar. Stability of the max-weight routing and scheduling protocol in dynamic networks and at critical loads. *Proc. STOC*, June 2007.

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

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

[28] 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.

[29] 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.

[30] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *IEEE Transactions on Information Theory*, vol. 53, no. 9, pp. 3095-3113, Sept. 2007.

[31] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1489-1501, Aug. 2006.

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

[33] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *Proc. IEEE INFOCOM*, 2005.

[34] X. Wu, R. Srikant, and J. R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Transactions on Mobile Computing*, June 2007.

[35] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Trans. on Information Theory*, vol. 54, no. 2, pp. 572-594, Feb. 2008.

[36] L. Lin, X. Lin, and N. B. Shroff. Low-complexity and distributed energy minimization in multi-hop wireless networks. *Proc. IEEE INFOCOM*, 2007.

[37] S. Deb, D. Shah, and S. Shakkottai. Fast matching algorithms for repetitive optimization: An application to switch scheduling. *Proc. of 40th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ*, March 2006.

[38] M. J. Neely. Delay analysis for maximal scheduling in wireless networks with bursty traffic. *Proc. IEEE INFOCOM*, April 2008.

[39] M. J. Neely. Delay analysis for max weight opportunistic scheduling in wireless systems. *Proc. 46th Annual Allerton Conf. on Communication, Control, and Computing*, Sept. 2008.