Delay Reduction via Lagrange Multipliers in Stochastic Network Optimization

Longbo Huang, Michael J. Neely

***Abstract*—In this paper, we consider the problem of reducing network delay in stochastic network utility optimization problems. We start by studying the recently proposed quadratic Lyapunov function based algorithms (QLA). We show that for every stochastic problem, there is a corresponding *deterministic* problem, whose dual optimal solution "exponentially attracts" the network backlog process under QLA. In particular, the probability that the backlog vector under QLA deviates from the attractor is exponentially decreasing in their Euclidean distance. This suggests that one can roughly "subtract out" a Lagrange multiplier from the system induced by QLA. We thus develop a family of *Fast Quadratic Lyapunov based Algorithms* (FQLA) that achieve an $[O(1/V), O(\log^2(V))]$ performance-delay tradeoff.**

These results highlight the "network gravity" role of Lagrange Multipliers in network scheduling. This role can be viewed as the counterpart of the "shadow price" role of Lagrange Multipliers in flow regulation for classic flow-based network problems.

***Index Terms*—Queueing, Dynamic Control, Lyapunov analysis, Stochastic Optimization**

I. Introduction

In this paper, we consider the problem of reducing network delay in the following general framework of the stochastic network utility optimization problem. We are given a time slotted stochastic network. The network state, such as the network channel condition, is time varying according to some probability law. A network controller performs some action based on the observed network state at every time slot. The chosen action incurs a cost (since cost minimization is mathematically equivalent to utility maximization, below we will use cost and utility interchangeably), but also serves some amount of traffic and possibly generates new traffic for the network. This traffic causes congestion, and thus leads to backlogs at nodes in the network. The goal of the controller is to minimize its time average cost subject to the constraint that the time average total backlog in the network is finite.

This setting is very general, and many existing works fall into this category. Further, many techniques have been used to study this problem (see [1] for a survey). In this paper, we focus on algorithms that are built upon quadratic Lyapunov functions (called QLA in the following), e.g., [2], [3], [4], [5], [6], [7]. These QLA algorithms are easy to implement, greedy in nature, and are parameterized by a scalar control variable V. It has been shown that when the network state is i.i.d., QLA algorithms can achieve a time average utility that is within $O(1/V)$ to the optimal. Therefore, as V grows large, the time average utility can be pushed arbitrarily close to the optimal. However, such close-to-optimal utility is usually at the expense of large network delay. In fact, in [3], [4], [7], it is shown that an $O(V)$ network delay is incurred when an $O(1/V)$ close-to-optimal utility is achieved. Two recent papers [8] and [9], which show that it is possible to achieve within $O(1/V)$ of optimal utility with only $O(\log(V))$ delay, use a more sophisticated algorithm design approach based on exponential Lyapunov functions. Therefore, it seems that though being simple in implementation, QLA algorithms have undesired delay performance.

However, we note that the delay results of QLA are usually given in terms of long term upper bounds of the average network backlog e.g., [7]. Thus they do not examine the possibility that the actual backlog vector (or its time average) converges to some fixed value. Work in [10] considers drift properties towards an "invariant" backlog vector, derived in the special case when the problem exhibits a unique optimal Lagrange multiplier. An upper bound on the long term deviation of the actual backlog and the Lagrange multiplier vector is obtained. While this suggests Lagrange multipliers are "gravitational attractors," the bounds in [10] do not show that the the actual backlog is very unlikely to deviate significantly from the attractor.

In this paper, we focus on obtaining stronger probability results of the steady state backlog process behavior under QLA. We first show that under QLA, even though the backlog can grow linearly in V, it "typically" stays close to an "attractor," which is the dual optimal solution of a *deterministic* optimization problem. In particular, the probability that the backlog vector deviates from the attractor is exponentially decreasing in distance, which significantly tightens the attractor analysis in [10]. This implies that a large amount of the data is kept in the network simply for maintaining the backlog at the "right" level. Therefore, even if we replace these data with some fake data (denoted as *place-holder bits* [11]), the performance of QLA will not be heavily affected. Based on this finding, we propose a family of *Fast Quadratic Lyapunov based Algorithms* (FQLA), which intuitively speaking, can be viewed as subtracting out a Lagrange multiplier from the system induced by QLA. We show that when the network state is i.i.d., FQLA is able to achieve within $O(1/V)$ of optimal utility with an $O(\log^2(V))$ delay guarantee. The development of FQLA also provides us with additional insights into QLA algorithms and the role of Lagrange multipliers in stochastic network optimization. We now summarize the main contributions of

Longbo Huang (email: longbohu@usc.edu) and Michael J. Neely (web: http://www-rcf.usc.edu/~mjneely) are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA.

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.

this paper in the following:

- This paper proves that in steady state, the backlog process under QLA is “exponentially attracted” to an attractor.
- This paper proposes a family of *Fast Quadratic Lyapunov based Algorithms (FQLA)*, which are usually easy to implement, and can achieve an $[O(1/V), O(\log^2(V))]$ performance-delay tradeoff for general stochastic optimization problems.
- This paper highlights a new functionality of Lagrange multipliers: the “network gravity” in network scheduling.

The paper is organized as follows: In Section II, we set up our notations. In Section III, we state our network model. We then review the QLA algorithm and define the *deterministic problem* in Section IV. In Section V, we show that the backlog process under QLA always stays close to an attractor. In Section VI, we propose the FQLA algorithm. Section VII provides simulation results. We discuss the “gravity” role of Lagrange multipliers and relate QLA to the randomized incremental subgradient method (RISM) [12] in Section VIII.

II. Notations

- $\mathbb{R}$: the set of real numbers
- $\mathbb{R}_+$ (or $\mathbb{R}_-$): the set of nonnegative (or non-positive) real numbers
- $\mathbb{R}^n$ (or $\mathbb{R}^n_+$): the set of n dimensional *column* vectors, with each element being in $\mathbb{R}$ (or $\mathbb{R}_+$)
- **bold** symbols $\boldsymbol{x}$ and $\boldsymbol{x}^T$: *column* vector and its transpose
- $\boldsymbol{x} \succeq \boldsymbol{y}$: vector $\boldsymbol{x}$ is entrywise no less than vector $\boldsymbol{y}$
- $\mathbf{0}$: column vector with all elements being 0

III. System Model

In this section, we specify the general network model we use. We consider a network controller that operates a network with the goal of minimizing the time average cost, subject to the queue stability constraint. The network is assumed to operate in slotted time, i.e., $t \in \{0, 1, 2, ...\}$. We assume there are $r \geq 1$ queues in the network.

A. Network State

We assume there are a total of M different random network states, and define $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$ as the set of possible states. Each particular state s_i indicates the current network parameters, such as a vector of channel conditions for each link, or a collection of other relevant information about the current network channels and arrivals. Let $S(t)$ denote the network state at time t. We assume that $S(t)$ is i.i.d. every time slot, and let p_{s_i} denote its probability of being in state s_i, i.e., $p_{s_i} = Pr\{S(t) = s_i\}$. We assume the network controller can observe $S(t)$ at the beginning of every slot t, but the p_{s_i} probabilities are not necessarily known.

B. The Cost, Traffic and Service

At each time t, after observing $S(t) = s_i$, the controller chooses an action $x(t)$ from a set $\mathcal{X}^{(s_i)}$, i.e., $x(t) = x^{(s_i)}$ for some $x^{(s_i)} \in \mathcal{X}^{(s_i)}$. The set $\mathcal{X}^{(s_i)}$ is called the feasible action set for network state s_i and is assumed to be time-invariant and compact for all $s_i \in \mathcal{S}$. The cost, traffic and service generated by the chosen action $x(t) = x^{(s_i)}$ are as follows:

(a) The chosen action has an associated cost given by the cost function $f(t) = f(s_i, x^{(s_i)}) : \mathcal{X}^{(s_i)} \mapsto \mathbb{R}_+$ (or $\mathcal{X}^{(s_i)} \mapsto \mathbb{R}_-$ in the case of reward maximization problems);
(b) The amount of traffic generated by the action to queue j is determined by the traffic function $A_j(t) = g_j(s_i, x^{(s_i)}) : \mathcal{X}^{(s_i)} \mapsto \mathbb{R}_+$, in units of packets;
(c) The amount of service allocated to queue j is given by the rate function $\mu_j(t) = b_j(s_i, x^{(s_i)}) : \mathcal{X}^{(s_i)} \mapsto \mathbb{R}_+$, in units of packets;

Note that $A_j(t)$ includes both the exogenous arrivals from outside the network to queue j, and the endogenous arrivals from other queues, i.e., the transmitted packets from other queues, to queue j (See Section III-C and III-D for further explanations). We assume the functions $f(s_i, \cdot)$, $g_j(s_i, \cdot)$ and $b_j(s_i, \cdot)$ are time-invariant, their magnitudes are uniformly upper bounded by some constant $\delta_{max} \in (0, \infty)$ for all s_i, j, and they are known to the network operator. We also assume that there exists a set of actions $\{x^{(s_i)k}\}_{i=1,...,M}^{k=1,...,r+2}$ with $x^{(s_i)k} \in \mathcal{X}^{(s_i)}$ such that $\sum_{s_i} p_{s_i} \big\{ \sum_k \vartheta_k^{(s_i)} [g_j(s_i, x^{(s_i)k}) - b_j(s_i, x^{(s_i)k})] \big\} \leq -\epsilon$ for some $\epsilon > 0$ for all j, with $\sum_j \vartheta_k^{(s_i)} = 1$ and $\vartheta_k^{(s_i)} \geq 0$ for all s_i and k. That is, the constraints are feasible with ϵ slackness. Thus, there exists a stationary randomized policy that stabilizes all queues (where $\vartheta_k^{(s_i)}$ represents the probability of choosing action $x^{(s_i)k}$ when $S(t) = s_i$). In the following, we use:

$$\boldsymbol{A}(t) = (A_1(t), A_2(t), ..., A_r(t))^T, \tag{1}$$
$$\boldsymbol{\mu}(t) = (\mu_1(t), \mu_2(t), ..., \mu_r(t))^T, \tag{2}$$

to denote the arrival and service vectors at time t. It is easy to see from above that if we define:

$$B = \sqrt{r}\delta_{max}, \tag{3}$$

then $\|\boldsymbol{A}(t) - \boldsymbol{\mu}(t)\| \leq B$ for all t.

C. Queueing, Average Cost and the Stochastic Problem

Let $\boldsymbol{U}(t) = (U_1(t), ..., U_r(t))^T \in \mathbb{R}^r_+$, $t = 0, 1, 2, ...$ be the queue backlog vector process of the network, in units of packets. We assume the following queueing dynamics:

$$U_j(t+1) = \max\big[U_j(t) - \mu_j(t), 0\big] + A_j(t) \quad \forall j, \tag{4}$$

and $\boldsymbol{U}(0) = \mathbf{0}$. Note that by using (4), we assume that when a queue does not have enough packets to send, null packets are transmitted. In this paper, we adopt the following notion of queue stability:

$$\mathbb{E}\big\{\sum_{j=1}^r U_j\big\} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{j=1}^r \mathbb{E}\big\{U_j(\tau)\big\} < \infty. \tag{5}$$

We also use f^π_{av} to denote the time average cost induced by an action-seeking policy π, defined as:

$$f^\pi_{av} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{f^\pi(\tau)\big\}, \tag{6}$$

where $f^\pi(\tau)$ is the cost incurred at time τ by policy π. We call an action-seeking policy under which (5) holds a *stable* policy, and use f_{av}^* to denote the optimal time average cost over all stable policies. Every slot, the network controller observes the current network state and chooses a control action, with the goal of minimizing time average cost subject to network stability. This goal can be mathematically stated as:

$$\min: \ f_{av}, \quad s.t. \ (5).$$

In the rest of the paper, we will refer to this problem as *the stochastic problem*. This stochastic problem framework can be used to model many network utility problems, such as the energy minimization problem [3] and the access point pricing problem [5]. We note that a similar network model with stochastic penalties is treated in [13] using a fluid model and a primal-dual approach that achieves optimality in a limiting sense. The framework is also treated in [7] using a quadratic Lyapunov based algorithm (QLA) that provides an explicit $[O(1/V), O(V)]$ performance-delay tradeoff when the network state is i.i.d..

D. An Example of the Model

Here we provide an example to illustrate our model. Consider the 2-queue network in Fig.1. Every slot, the network operator makes a decision on whether or not to allocate one unit power to serve packets at each queue, so as to support all arriving traffic, i.e., maintain queue stability, with minimum energy expenditure. Every slot, the number of arrival packets $R(t)$, is i.i.d., being either 2 or 0 with probabilities $5/8$ and $3/8$ respectively. The channel states $S_1(t), S_2(t)$ are also i.i.d. being either "G=good" or "B=bad" with equal probabilities. One unit of power can serve 2 packets in a good channel but can only serve one in a bad channel. Both channels can be activated simultaneously without affecting each other.

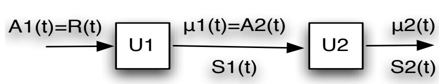

Fig. 1. A 2-queue system

In this case, a network state $S(t)$ is a $(R(t), S_1(t), S_2(t))$ tuple and $S(t)$ is i.i.d.. There are eight possible network states. At each state s_i, the action $x^{(s_i)}$ is a pair (x_1, x_2), with x_i being the amount of energy spent at queue i, and $(x_1, x_2) \in \mathcal{X}^{(s_i)} = \{0/1, 0/1\}$. The cost function is always $f(s_i, x^{(s_i)}) = x_1 + x_2$ for all s_i. The network states, the traffic functions and service rate functions are summarized in Fig. 2. Note here $A_1(t) = R(t)$ is part of $S(t)$ and thus is independent of $x^{(s_i)}$; while $A_2(t) = \mu_1(t)$ hence depends on $x^{(s_i)}$. Also note that $A_2(t)$ equals $\mu_1(t)$ instead of $\min[\mu_1(t), U_1(t)]$ due to our idle fill assumption in Section III-C.

IV. QLA AND THE DETERMINISTIC PROBLEM

In this section, we first review the quadratic Lyapunov functions based algorithms (the QLA algorithm) [7] for solving the stochastic problem. Then we define the *deterministic problem* and its dual. We then describe the ordinary subgradient method (OSM) that can be used to solve the dual. The dual problem and OSM will also be used later for our analysis of the steady state backlog behavior under QLA.

S(t)	$R(t)$	$S_1(t)$	$S_2(t)$	$A_1(t)$	$A_2(t)$	$\mu_1(t)$	$\mu_2(t)$
s_1	0	B	B	0	x_1	x_1	x_2
s_2	0	B	G	0	x_1	x_1	$2x_2$
s_3	0	G	B	0	$2x_1$	$2x_1$	x_2
s_4	0	G	G	0	$2x_1$	$2x_1$	$2x_2$
s_5	2	B	B	2	x_1	x_1	x_2
s_6	2	B	G	2	x_1	x_1	$2x_2$
s_7	2	G	B	2	$2x_1$	$2x_1$	x_2
s_8	2	G	G	2	$2x_1$	$2x_1$	$2x_2$

Fig. 2. Network state, Traffic and Rate functions

A. The QLA algorithm

To solve the stochastic problem using QLA, we first define a quadratic Lyapunov function $L(\boldsymbol{U}(t)) = \frac{1}{2}\sum_{j=1}^{r} U_j^2(t)$. We then define the one-slot conditional Lyapunov drift: $\Delta(\boldsymbol{U}(t)) = \mathbb{E}\big\{L(\boldsymbol{U}(t+1)) - L(\boldsymbol{U}(t)) \,|\, \boldsymbol{U}(t)\big\}$. From (4), we obtain the following drift expression:

$$\Delta(\boldsymbol{U}(t)) \leq C - \mathbb{E}\big\{\sum_{j=1}^{r} U_j(t)\big[\mu_j(t) - A_j(t)\big] \,|\, \boldsymbol{U}(t)\big\},$$

where $C = r\delta_{max}^2$. Now add to both sides the term $V\mathbb{E}\big\{f(t) \,|\, \boldsymbol{U}(t)\big\}$, where $V \geq 1$ is a scalar control variable, we obtain:

$$\Delta(\boldsymbol{U}(t)) + V\mathbb{E}\big\{f(t) \,|\, \boldsymbol{U}(t)\big\} \leq C - \mathbb{E}\Big\{ - Vf(t) + \sum_{j=1}^{r} U_j(t)\big[\mu_j(t) - A_j(t)\big] \,|\, \boldsymbol{U}(t)\Big\}. \quad (7)$$

The QLA algorithm is then obtained by choosing an action x at every time slot t to *minimize the right hand side of (7)* given $\boldsymbol{U}(t)$. Specifically, the QLA algorithm works as follows:

QLA: At every time slot t, observe the current network state $S(t)$ and the backlog $\boldsymbol{U}(t)$. If $S(t) = s_i$, choose $x^{(s_i)} \in \mathcal{X}^{(s_i)}$ that solves the following:

$$\begin{aligned} \max \quad & -Vf(s_i, x) + \sum_{j=1}^{r} U_j(t)\big[b_j(s_i, x) - g_j(s_i, x)\big] \quad (8) \\ s.t. \quad & x \in \mathcal{X}^{(s_i)}. \end{aligned}$$

Depending on the problem structure, (8) can usually be decomposed into separate parts that are easier to solve, e.g., [3], [5]. Also, it can be shown, as in [7] that,

$$f_{av}^{QLA} = f_{av}^* + O(1/V), \quad \overline{U}^{QLA} = O(V), \quad (9)$$

where f_{av}^{QLA} is the average cost under QLA and $\overline{U}^{QLA}$ is the time average network backlog size under QLA.

B. The Deterministic Problem

Consider *the deterministic problem* as follows:

$$\begin{aligned} \min \quad & \mathcal{F}(\boldsymbol{x}) \triangleq V\sum_{s_i} p_{s_i} f(s_i, x^{(s_i)}) \quad (10) \\ s.t. \quad & \mathcal{G}_j(\boldsymbol{x}) \triangleq \sum_{s_i} p_{s_i} g_j(s_i, x^{(s_i)}) \\ & \qquad \leq \mathcal{B}_j(\boldsymbol{x}) \triangleq \sum_{s_i} p_{s_i} b_j(s_i, x^{(s_i)}) \quad \forall j \\ & x^{(s_i)} \in \mathcal{X}^{(s_i)} \quad \forall i = 1, 2, ..., M, \end{aligned}$$

where p_{s_i} corresponds to the probability of $S(t) = s_i$ and $\boldsymbol{x} = (x^{(s_1)}, ..., x^{(s_M)})^T$. The dual problem of (10) can be obtained as follows:

$$\begin{aligned} \max \quad & q(\boldsymbol{U}) \\ s.t. \quad & \boldsymbol{U} \succeq \boldsymbol{0}, \end{aligned} \tag{11}$$

where $q(\boldsymbol{U})$ is called the dual function and is defined as:

$$q(\boldsymbol{U}) = \inf_{x^{(s_i)} \in \mathcal{X}^{(s_i)}} \bigg\{ V \sum_{s_i} p_{s_i} f(s_i, x^{(s_i)}) \tag{12}$$
$$+ \sum_j U_j \big[\sum_{s_i} p_{s_i} g_j(s_i, x^{(s_i)}) - \sum_{s_i} p_{s_i} b_j(s_i, x^{(s_i)}) \big] \bigg\}.$$

By rearranging the terms, we note that $q(\boldsymbol{U})$ can also be written in the following separable form, which is more useful for our later analysis.

$$q(\boldsymbol{U}) = \inf_{x^{(s_i)} \in \mathcal{X}^{(s_i)}} \sum_{s_i} p_{s_i} \bigg\{ V f(s_i, x^{(s_i)}) \tag{13}$$
$$+ \sum_j U_j \big[g_j(s_i, x^{(s_i)}) - b_j(s_i, x^{(s_i)}) \big] \bigg\}.$$

Here $\boldsymbol{U} = (U_1, ..., U_r)^T$ is the *Lagrange multiplier* of (10). It is well known that $q(\boldsymbol{U})$ in (12) is concave in the vector $\boldsymbol{U}$, and hence the problem (11) can usually be solved efficiently, particularly when cost functions and rate functions are separable over different different network components. It is also well known that in many situations, the optimal value of (11) is the same as the optimal value of (10) and in this case we say that there is no duality gap [12].

We note that the deterministic problem (10) is not necessarily convex as the sets $\mathcal{X}^{(s_i)}$ are not necessarily convex, and the functions $f(s_i, \cdot)$, $g_j(s_i, \cdot)$ and $b_j(s_i, \cdot)$ are not necessarily convex. Therefore, there may be a duality gap between the deterministic problem (10) and its dual (11). Furthermore, solving the deterministic problem (10) may not solve the stochastic problem. This is so since at every network state, the stochastic problem may require time sharing over more than one action, but the solution to the deterministic problem gives only a fixed operating point per network state. However, one can show, by using an argument similar to showing the existence of an optimal stationary randomized algorithm in [5], that the dual problem (11) gives the exact value of $V f^*_{av}$, where f^*_{av} is the optimal time average cost, even if (10) is non-convex.

Among the many algorithms that can be used to solve (11), the following algorithm is the most common one (for performance see [12]), we denote it as the *ordinary subgradient method* (OSM):

OSM: Initialize $\boldsymbol{U}(0)$; at every iteration t, observe $\boldsymbol{U}(t)$,

1) Find $x^{(s_i)}_{\boldsymbol{U}} \in \mathcal{X}^{(s_i)}$ for $i \in \{1, ..., M\}$ that achieves the infimum of the right hand side of (12).
2) Using the $\boldsymbol{x}_{\boldsymbol{U}} = (x^{(s_1)}_{\boldsymbol{U}}, x^{(s)}_{\boldsymbol{U}}, ..., x^{(s_M)}_{\boldsymbol{U}})^T$ found, update:

$$U_j(t+1) = \max \bigg[U_j(t) - \alpha^t \sum_{s_i} p_{s_i} \big[b_j(s_i, x^{(s_i)}_{\boldsymbol{U}}) \tag{14}$$
$$- g_j(s_i, x^{(s_i)}_{\boldsymbol{U}}) \big], 0 \bigg].$$

We use $x^{(s_i)}_{\boldsymbol{U}}$ to highlight its dependency on $\boldsymbol{U}(t)$. The term $\alpha^t > 0$ is called the *step size* at iteration t. In the following, we will always assume $\alpha^t = 1$ when referring to OSM. Note that if there is only one network state, QLA and OSM will choose the same action given the same $\boldsymbol{U}$, and they differ only by (4) and (14). The term $\boldsymbol{G}_{\boldsymbol{U}} = (G_{\boldsymbol{U},1}, G_{\boldsymbol{U},2}, ..., G_{\boldsymbol{U},r})^T$, with:

$$\begin{aligned} G_{\boldsymbol{U},j} &= \mathcal{G}_j(\boldsymbol{x}_{\boldsymbol{U}}) - \mathcal{B}_j(\boldsymbol{x}_{\boldsymbol{U}}) \\ &= \sum_{s_i} p_{s_i} \big[- b_j(s_i, x^{(s_i)}_{\boldsymbol{U}}) + g_j(s_i, x^{(s_i)}_{\boldsymbol{U}}) \big], \end{aligned} \tag{15}$$

is called the *subgradient* of $q(\boldsymbol{U})$ at $\boldsymbol{U}(t)$. It is well known that for any other $\hat{\boldsymbol{U}} \in \mathbb{R}^r$, we have:

$$(\hat{\boldsymbol{U}} - \boldsymbol{U}(t))^T \boldsymbol{G}_{\boldsymbol{U}} \geq q(\hat{\boldsymbol{U}}) - q(\boldsymbol{U}(t)). \tag{16}$$

Using $\|\boldsymbol{G}_{\boldsymbol{U}}\| \leq B$, we note that (16) also implies:

$$q(\hat{\boldsymbol{U}}) - q(\boldsymbol{U}(t)) \leq B \|\hat{\boldsymbol{U}} - \boldsymbol{U}(t)\| \quad \forall \hat{\boldsymbol{U}}, \boldsymbol{U} \in \mathbb{R}^r \tag{17}$$

We are now ready to study the steady state behavior of $\boldsymbol{U}(t)$ under QLA. To simplify notations and highlight the scaling effect of the scalar V in QLA, we use the following notations:

1) We use $q_0(\boldsymbol{U})$ and $\boldsymbol{U}^*_0$ to denote the dual objective function and an optimal solution of (11) when $V = 1$; and use $q(\boldsymbol{U})$ and $\boldsymbol{U}^*_V$ (also called the optimal Lagrange multiplier) for their counterparts with general $V \geq 1$;
2) We use $x^{(s_i)}_{\boldsymbol{U}}$ to denote an action chosen by QLA for a given $\boldsymbol{U}(t)$ and $S(t) = s_i$; and use $\boldsymbol{x}_{\boldsymbol{U}} = (x^{(s_1)}_{\boldsymbol{U}}, ..., x^{(s_M)}_{\boldsymbol{U}})^T$ to denote a solution chosen by OSM for a given $\boldsymbol{U}(t)$.

To simplify analysis, we assume the following throughout:

***Assumption** 1:* $\boldsymbol{U}^*_V = (U^*_{V1}, ..., U^*_{Vr})^T$ is unique for all $V \geq 1$.

Note that Assumption 1 is not very restrictive. In fact, it holds in many network utility optimization problems, e.g., [10]. In many cases, we also have $\boldsymbol{U}^*_V \neq \boldsymbol{0}$. Moreover, for the assumption to hold for all $V \geq 1$, it suffices to have just $\boldsymbol{U}^*_0$ being unique. This is shown in the following lemma regarding the scaling effect of the parameter V on the optimal Lagrange multiplier.

***Lemma** 1:* $\boldsymbol{U}^*_V = V \boldsymbol{U}^*_0$.

Proof: From (13) we see that:

$$q(\boldsymbol{U})/V = \inf_{x^{(s_i)} \in \mathcal{X}^{(s_i)}} \sum_{s_i} p_{s_i} \bigg\{ f(s_i, x^{(s_i)})$$
$$+ \sum_j \hat{U}_j \big[g_j(s_i, x^{(s_i)}) - b_j(s_i, x^{(s_i)}) \big] \bigg\},$$

where $\hat{U}_j = \frac{U_j}{V}$. However, the right hand side is exactly $q_0(\hat{\boldsymbol{U}})$, and thus is maximized at $\hat{\boldsymbol{U}} = \boldsymbol{U}^*_0$. Hence $q(\boldsymbol{U})$ is maximized at $V \boldsymbol{U}^*_0$. ■

V. Backlog vector behavior under QLA

In this section we study the backlog vector behavior under QLA of the stochastic problem. The following theorem summarizes the main results. Recall that B is defined in (3) as the upper bound of the magnitude change of $\boldsymbol{U}$ in a slot.

***Theorem** 1:* If the dual function $q_0(\boldsymbol{U})$ satisfies:

$$q_0(\boldsymbol{U}_0^*) \geq q_0(\boldsymbol{U}) + L\|\boldsymbol{U}_0^* - \boldsymbol{U}\| \qquad \forall\, \boldsymbol{U} \succeq \boldsymbol{0}, \tag{18}$$

for some constant $L > 0$ independent of V, then under QLA,

(a) There exist constants $D \geq \eta > 0$, both independent of V, such that whenever $\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| \geq D$, we have:

$$\mathbb{E}\big\{\|\boldsymbol{U}(t+1) - \boldsymbol{U}_V^*\| \mid \boldsymbol{U}(t)\big\} \leq \|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| - \eta. \tag{19}$$

In particular, the constants D and η that satisfy (19) can be chosen as follows: Choose η as any value such that $0 < \eta < L$, independent of V. Then, choose D as: [1]

$$D = \max\left[\frac{2B^2 - \eta^2}{2(L-\eta)}, \eta\right]. \tag{20}$$

(b) For given constants D, η in (a), there exist some constants $c^*, \beta^* > 0$, independent of V, such that:

$$\mathcal{P}(D, m) \leq c^* e^{-\beta^* m}, \tag{21}$$

where $\mathcal{P}(D, m)$ is defined as:

$$\mathcal{P}(D, m) \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} Pr\{\|\boldsymbol{U}(\tau) - \boldsymbol{U}_V^*\| > D + m\}. \tag{22}$$

Note that if $m = \frac{\log(V)}{\beta^*}$, by (21) we have $\mathcal{P}(D, m) \leq \frac{c^*}{V}$. Also if a steady state distribution of $\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\|$ exists under QLA, i.e., the limit of $\frac{1}{t}\sum_{\tau=0}^{t-1} Pr\{\|\boldsymbol{U}(\tau) - \boldsymbol{U}_V^*\| > D + m\}$ exists as $t \to \infty$, then one can replace $\mathcal{P}(D, m)$ with the steady state probability that $\boldsymbol{U}(t)$ deviates from $\boldsymbol{U}_V^*$ by an amount of $D + m$, i.e., $Pr\{\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| > D + m\}$. Therefore Theorem 1 can be viewed as showing that when (18) is satisfied, for a large V, the backlog $\boldsymbol{U}(t)$ under QLA will mostly be within $O(\log(V))$ distance from $\boldsymbol{U}_V^*$. This implies that the average backlog will roughly be $\sum U_{Vj}^*$, which is typically $\Theta(V)$ by Lemma 1. However, this fact will also allow us to build FQLA upon QLA to "subtract out" roughly $\sum U_{Vj}^*$ data from the network and reduce network delay. Theorem 1 also highlights a deep connection between the steady state behavior of the network backlog process $\boldsymbol{U}(t)$ and the structure of the dual function $q_0(\boldsymbol{U})$. We note that (18) is not very restrictive. In fact, if $q_0(\boldsymbol{U})$ is polyhedral (e.g., $\mathcal{X}^{(s_i)}$ is finite for all s_i), with a unique optimal solution $\boldsymbol{U}_0^* \succeq \boldsymbol{0}$, then (18) can be satisfied (see Section VII for an example). To prove the theorem, we need the following lemma.

***Lemma** 2:* Under QLA, we have for all t,

$$\begin{aligned}\mathbb{E}\big\{\|\boldsymbol{U}(t+1) - \boldsymbol{U}_V^*\|^2 \mid \boldsymbol{U}(t)\big\} \leq \|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\|^2 + 2B^2 \\ -2(q(\boldsymbol{U}_V^*) - q(\boldsymbol{U}(t))).\end{aligned} \tag{23}$$

Proof: See [14]. ∎

We now use Lemma 2 to prove Theorem 1.

Proof: (Theorem 1) Part (a): We first show that if (18) holds for $q_0(\boldsymbol{U})$ with L, then it also holds for $q(\boldsymbol{U})$ with the same L. To this end, suppose (18) holds for $q_0(\boldsymbol{U})$ for all $\boldsymbol{U} \succeq \boldsymbol{0}$. Multiplying both sides of (18) by V, we get:

$$Vq_0(\boldsymbol{U}_0^*) \geq Vq_0(\boldsymbol{U}) + LV\|\boldsymbol{U}_0^* - \boldsymbol{U}\|.$$

[1]It can be seen from (17) that $B \geq L$. Thus $B > \eta$.

Now using $\boldsymbol{U}_V^* = V\boldsymbol{U}_0^*$ and $q(\boldsymbol{U}) = Vq_0(\boldsymbol{U}/V)$ in the above inequality, we have for all $\boldsymbol{U} \succeq \boldsymbol{0}$:

$$q(\boldsymbol{U}_V^*) \geq q(V\boldsymbol{U}) + L\|\boldsymbol{U}_V^* - V\boldsymbol{U}\|.$$

Thus for any $\boldsymbol{U} \succeq \boldsymbol{0}$, we have:

$$q(\boldsymbol{U}_V^*) \geq q(\boldsymbol{U}) + L\|\boldsymbol{U}_V^* - \boldsymbol{U}\|. \tag{24}$$

Now for a given $\eta > 0$, if:

$$2B^2 - 2\big(q(\boldsymbol{U}_V^*) - q(\boldsymbol{U}(t))\big) \leq \eta^2 - 2\eta\|\boldsymbol{U}_V^* - \boldsymbol{U}(t)\|, \tag{25}$$

then by (23), we have:

$$\mathbb{E}\big\{\|\boldsymbol{U}(t+1) - \boldsymbol{U}_V^*\|^2 \mid \boldsymbol{U}(t)\big\} \leq (\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| - \eta)^2,$$

which then by Jensen's inequality implies:

$$(\mathbb{E}\big\{\|\boldsymbol{U}(t+1) - \boldsymbol{U}_V^*\| \mid \boldsymbol{U}(t)\big\})^2 \leq (\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| - \eta)^2.$$

Thus (19) follows whenever (25) holds and $\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| \geq \eta$. It suffices to choose D and η such that $D \geq \eta$ and that (25) holds whenever $\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| \geq D$. Now note that (25) can be rewritten as the following inequalty:

$$q(\boldsymbol{U}_V^*) \geq q(\boldsymbol{U}(t)) + \eta\|\boldsymbol{U}_V^* - \boldsymbol{U}(t)\| + \mathcal{Y} \tag{26}$$

where $\mathcal{Y} = \frac{2B^2 - \eta^2}{2}$. Choose $\eta \in (0, L)$ independent of V. By (24), if:

$$L\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| \geq \eta\|\boldsymbol{U}_V^* - \boldsymbol{U}(t)\| + \mathcal{Y} \tag{27}$$

then (26) holds. Now choose D as defined in (20), we see that if $\|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\| \geq D$, then (27) holds, which implies (26), and equivalently (25). We also have $D \geq \eta$, hence (19) holds.

Part (b): Now we show that (19) implies (21). Choose constants D and η that are independent of V in (a). Denote $Y(t) = \|\boldsymbol{U}(t) - \boldsymbol{U}_V^*\|$, we see then whenever $Y(t) \geq D$, we have $\mathbb{E}\big\{Y(t+1) - Y(t) \mid \boldsymbol{U}(t)\big\} \leq -\eta$. It is also easy to see that $|Y(t+1) - Y(t)| \leq B$, as B is defined in (3) as the upper bound of the magnitude change of $\boldsymbol{U}$ in a slot. Define $\tilde{Y}(t) = \max\big[Y(t) - D, 0\big]$. We see that whenever $\tilde{Y}(t) \geq B$, we have:

$$\begin{aligned}&\mathbb{E}\big\{\tilde{Y}(t+1) - \tilde{Y}(t) \mid \boldsymbol{U}(t)\big\} \\ &\qquad = \mathbb{E}\big\{Y(t+1) - Y(t) \mid \boldsymbol{U}(t)\big\} \leq -\eta.\end{aligned} \tag{28}$$

Now define a Lyapunov function of $\tilde{Y}(t)$ to be $L(\tilde{Y}(t)) = e^{w\tilde{Y}(t)}$ with some $w > 0$, and define the one-slot conditional *drift* to be:

$$\begin{aligned}\Delta(\tilde{Y}(t)) &\triangleq \mathbb{E}\big\{L(\tilde{Y}(t+1)) - L(\tilde{Y}(t)) \mid \boldsymbol{U}(t)\big\} \\ &= \mathbb{E}\big\{e^{w\tilde{Y}(t+1)} - e^{w\tilde{Y}(t)} \mid \boldsymbol{U}(t)\big\}.\end{aligned} \tag{29}$$

It is shown in Appendix A that by choosing $w = \frac{\eta}{B^2 + B\eta/3}$, we have for all $\tilde{Y}(t) \geq 0$:

$$\Delta(\tilde{Y}(t)) \leq e^{2wB} - \frac{w\eta}{2}e^{w\tilde{Y}(t)}. \tag{30}$$

Taking expectation on both sides and carrying out a telescoping sum, we obtain:

$$\sum_{\tau=0}^{t-1}\frac{w\eta}{2}\mathbb{E}\big\{e^{w\tilde{Y}(\tau)}\big\} \leq te^{2wB} + \mathbb{E}\big\{e^{w\tilde{Y}(0)}\big\}. \tag{31}$$

Divide both sides by t and take the limsup as t goes to infinity, we obtain:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\frac{w\eta}{2}\mathbb{E}\big\{e^{w\tilde{Y}(\tau)}\big\} \quad \leq \quad e^{2wB}. \tag{32}$$

Using that $\mathbb{E}\big\{e^{w\tilde{Y}(\tau)}\big\} \geq e^{wm}Pr\{\tilde{Y}(\tau) > m\}$, we obtain:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\frac{w\eta}{2}e^{wm}Pr\{\tilde{Y}(\tau) > m\} \quad \leq \quad e^{2wB}. \tag{33}$$

Plug in $w = \frac{\eta}{B^2+B\eta/3}$ and use the definition of $\tilde{Y}(t)$, we get:

$$\begin{aligned}\mathcal{P}(D,m) \quad &\leq \quad \frac{2e^{2wB}}{w\eta}e^{-wm} \\ &= \quad \frac{2(B^2+B\eta/3)e^{\frac{2\eta}{B+\eta/3}}}{\eta^2}e^{-\frac{\eta m}{B^2+B\eta/3}},\end{aligned} \tag{34}$$

where $\mathcal{P}(D,m)$ is defined in (22). Therefore (21) holds with:

$$c^* = \frac{2(B^2+B\eta/3)e^{\frac{2\eta}{B+\eta/3}}}{\eta^2}, \qquad \beta^* = \frac{\eta}{B^2+B\eta/3}. \tag{35}$$

It is easy to see that c^* and β^* are independent of V. ∎

Note from (31) and (32) that Theorem 1 indeed holds for any finite $\boldsymbol{U}(0)$. We will later use this fact to prove the performance of FQLA. The following theorem is a special case of Theorem 1 and gives a more direct illustration of Theorem 1. Recall that $\mathcal{P}(D,m)$ is defined in (22). Define:

$$\begin{aligned}&\mathcal{P}^{(r)}(D,m) \\ &\quad\triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}Pr\{\exists\, j, |U_j(\tau)-U^*_{Vj}| > D+m\}.\end{aligned} \tag{36}$$

***Theorem** 2:* If the condition in Theorem 1 holds, then under QLA, for any $c > 0$:

$$\mathcal{P}(D_1, cK_1\log(V)) \quad \leq \quad \frac{c_1^*}{V^c}, \tag{37}$$

$$\mathcal{P}^{(r)}(D_1, cK_1\log(V)) \quad \leq \quad \frac{c_1^*}{V^c}. \tag{38}$$

where $D_1 = \frac{2B^2}{L} + \frac{L}{4}$, $K_1 = \frac{B^2+BL/6}{L/2}$ and $c_1^* = \frac{8(B^2+BL/6)e^{\frac{L}{B+L/6}}}{L^2}$.

Proof: Choose $\eta = L/2$, then we see from (20) that

$$D = \max\left[\frac{2B^2-L^2/4}{L}, \frac{L}{2}\right] \leq \frac{2B^2}{L}+\frac{L}{4}.$$

Now by (35) we see that (21) holds with $c^* = c_1^*$ and $\beta^* = \frac{L/2}{B^2+BL/6}$. Thus by taking $D_1 = \frac{2B^2}{L}+\frac{L}{4}$, we have:

$$\begin{aligned}\mathcal{P}(D_1, cK_1\log(V)) \quad &\leq \quad c^*e^{-cK_1\beta^*\log(V)} \\ &= \quad c_1^*e^{-c\log(V)},\end{aligned}$$

where the last step follows since $\beta^*K_1 = 1$. Thus (37) follows. Equation (38) follows from (37) by using the fact that for any constant ζ, the events $\mathcal{E}_1 = \{\exists\, j, |U_j(\tau)-U^*_{Vj}| > \zeta\}$ and $\mathcal{E}_2 = \{\|\boldsymbol{U}(\tau)-\boldsymbol{U}^*_V\| > \zeta\}$ satisfy $\mathcal{E}_1 \subset \mathcal{E}_2$. Thus: $Pr\{\exists\, j, |U_j(\tau)-U^*_{Vj}| > \zeta\} \leq Pr\{\|\boldsymbol{U}(\tau)-\boldsymbol{U}^*_V\| > \zeta\}$. ∎

Theorem 2 can be viewed as showing that for a large V, the probability for $U_j(t)$ to deviate from the j^{th} component of $\boldsymbol{U}^*_V$ is exponentially decreasing in the distance. Thus it rarely deviates from U^*_{Vj} by more than $\Theta(\log(V))$ distance. Note that one can similarly prove the following theorem for OSM:

***Theorem** 3:* If the condition in Theorem 1 holds, then there exist positive constants $D = \Theta(1)$ and $\eta = \Theta(1)$, i.e, independent of V, such that, under OSM, if $\|\boldsymbol{U}(t)-\boldsymbol{U}^*_V\| \geq D$,

$$\|\boldsymbol{U}(t+1)-\boldsymbol{U}^*_V\| \leq \|\boldsymbol{U}(t)-\boldsymbol{U}^*_V\| - \eta. \tag{39}$$

Proof: It is easy to show that under OSM, Lemma 2 holds without the expectation. Thus the theorem follows by the same argument as in Theorem 1. ∎

Therefore, when there is a single network state, we see that given (18), the backlog process converges to a ball of size $\Theta(1)$ around $\boldsymbol{U}^*_V$.

VI. The FQLA Algorithm

In this section, we propose a family of *Fast Quadratic Lyapunov based Algorithms* (FQLA) for general stochastic network optimization problems. We first provide an example to illustrate the idea of FQLA. We then describe FQLA with known $\boldsymbol{U}^*_V$, called FQLA-Ideal, and study its performance. After that, we describe the more general FQLA without such knowledge, called FQLA-General.

A. FQLA: a Single Queue Example

To illustrate the idea of FQLA, we first look at an example. Figure 3 shows a 10^4-slot sample backlog process under QLA.[2] We see that after roughly 1500 slots, $U(t)$ always stays very close to U^*_V, which is a $\Theta(V)$ scalar in this case. To reduce delay, we can first find $\mathcal{W} \in (0, U^*_V)$ such that: under QLA, there exists a time t_0 so that $U(t_0) \geq \mathcal{W}$ and once $U(t) \geq \mathcal{W}$, it remains so for all time (the solid line in Fig. 3 shows one for these 10^4 slots). We then place $\mathcal{W}$ fake bits (called *place-holder bits* [11]) in the queue at time 0, i.e., initialize $U(0) = \mathcal{W}$, and run QLA. It is easy to show that the utility performance of QLA will remain the same with this change, and the average backlog is now reduced by $\mathcal{W}$. However, such a $\mathcal{W}$ may require $\mathcal{W} = U^*_V - \Theta(V)$, thus the average backlog may still be $\Theta(V)$.

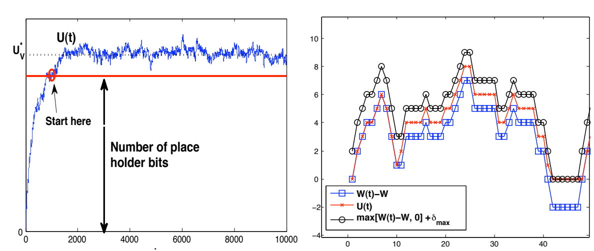

Fig. 3. Left: A sample backlog process; Right: An Example of $W(t)$ and $U(t)$.

FQLA instead finds a $\mathcal{W}$ such that in steady state, the backlog process under QLA *rarely* goes below it, and places $\mathcal{W}$ place-holder bits in the queue at time 0. FQLA then uses an auxiliary process $W(t)$, called the *virtual backlog process*, to keep track of the backlog process that should have

[2] This sample backlog process is one sample backlog process of queue 1 of the system considered in Section VII, under QLA with $V = 50$.

been generated if QLA is used. Specifically, FQLA initializes $W(0) = \mathcal{W}$. Then at every slot, QLA is run using $W(t)$ as the queue size, and $W(t)$ is updated according to QLA. With $W(t)$ and $\mathcal{W}$, FQLA works as follows: At time t, if $W(t) \geq \mathcal{W}$, FQLA performs QLA's action (obtained based on $S(t)$ and $W(t)$); else if $W(t) < \mathcal{W}$, FQLA carefully modifies QLA's action so as to maintain $U(t) \approx \max[W(t) - \mathcal{W}, 0]$ for all t (see Fig.3 for an example). Similar as above, this roughly reduces the average backlog by $\mathcal{W}$. The difference is that now we can show that $\mathcal{W} = \max[U_V^* - \log^2(V), 0]$ meets the requirement. Thus it is possible to bring the average backlog down to $O(\log^2(V))$. Also, since $W(t)$ can be viewed as a backlog process generated by QLA, it rarely goes below $\mathcal{W}$ in steady state. Hence FQLA is almost always the same as QLA, thus is able to achieve an $O(1/V)$ close-to-optimal utility performance.

B. The FQLA-Ideal Algorithm

In this section, we present the FQLA-Ideal algorithm. We assume the value $\boldsymbol{U}_V^* = (U_{V1}^*, ..., U_{Vr}^*)^T$ is known a-priori.

FQLA-Ideal:

(I) Determining place-holder bits: For each j, define:

$$\mathcal{W}_j = \max\left[U_{Vj}^* - \log^2(V), 0\right], \tag{40}$$

as the number of *place-holder bits* of queue j.

(II) Place-holder-bit based action: Initialize

$$U_j(0) = 0, \quad W_j(0) = \mathcal{W}_j, \quad \forall j.$$

For $t \geq 1$, observe the network state $S(t)$, solve (8) with $\boldsymbol{W}(t)$ in place of $\boldsymbol{U}(t)$. Perform the chosen action with the following modification: Let $\boldsymbol{A}(t)$ and $\boldsymbol{\mu}(t)$ be the arrival and service rate vectors generated by the action. For each queue j, do (Idle fill whenever needed):

a) If $W_j(t) \geq \mathcal{W}_j$: admit $A_j(t)$ arrivals, serve $\mu_j(t)$ data, i.e., update the backlog by:

$$U_j(t+1) = \max\left[U_j(t) - \mu_j(t), 0\right] + A_j(t).$$

b) If $W_j(t) < \mathcal{W}_j$: admit $\tilde{A}_j(t) = \max\left[A_j(t) - \mathcal{W}_j + W_j(t), 0\right]$ arrivals, serve $\mu_j(t)$ data, i.e., update the backlog by:

$$U_j(t+1) = \max\left[U_j(t) - \mu_j(t), 0\right] + \tilde{A}_j(t).$$

c) Update $W_j(t)$ by:

$$W_j(t+1) = \max\left[W_j(t) - \mu_j(t), 0\right] + A_j(t).$$

From above we see that FQLA-Ideal is the same as QLA based on $\boldsymbol{W}(t)$ when $W_j(t) \geq \mathcal{W}_j$ for all j. When $W_j(t) < \mathcal{W}_j$ for some queue j, FQLA-Ideal admits roughly the *excessive* packets after $W_j(t)$ is brought back to be above $\mathcal{W}_j$ for the queue. Thus for problems where QLA admits an easy implementation, e.g., [3], [5], it is also easy to implement FQLA. However, we also notice two different features of FQLA: (1) By (40), $\mathcal{W}_j$ can be 0. However, when V is large, this happens only when $U_{0j}^* = U_{Vj}^* = 0$ according to Lemma 1. In this case $\mathcal{W}_j = U_{Vj}^* = 0$, and queue j indeed needs zero place-holder bits. (2) Packets may be dropped in Step II-(b) upon their arrivals, or after they are admitted into the network in a multihop problem. Such packet dropping is natural in many flow control problems and does not change the nature of these problems. In other problems where such option is not available, the packet dropping option is introduced to achieve desired delay performance, and it can be shown that the fraction of packets dropped can be made arbitrarily small. Note that packet dropping here is to compensate for the deviation from the desired Lagrange multiplier, thus is different from that in [15], where packet dropping is used for drift steering.

C. Performance of FQLA-Ideal

We look at the performance of FQLA-Ideal in this section. We first have the following lemma that shows the relationship between $\boldsymbol{U}(t)$ and $\boldsymbol{W}(t)$. We will use it later to prove the delay bound of FQLA. Note that the lemma also holds for FQLA-General described later, as FQLA-Ideal/General differ only in the way of determining $\boldsymbol{\mathcal{W}} = (\mathcal{W}_1, ..., \mathcal{W}_r)^T$.

Lemma 3: Under FQLA-Ideal/General, we have $\forall\ j, t$:

$$\max\left[W_j(t) - \mathcal{W}_j, 0\right] \leq U_j(t) \leq \max\left[W_j(t) - \mathcal{W}_j, 0\right] + \delta_{max} \tag{41}$$

where δ_{max} is defined in Section III-B to be the upper bound of the number of arriving or departing packets of a queue.

Proof: See [14]. ■

The following theorem summarizes the main performance results of FQLA-Ideal. Recall that for a given policy π, f_{av}^π denotes its average cost defined in (6) and $f^\pi(t)$ denotes the cost induced by π at time t.

Theorem 4: If the condition in Theorem 1 holds and a steady state distribution exists for the backlog process generated by QLA, then with a sufficiently large V, we have under FQLA-Ideal that,

$$\overline{U} = O(\log^2(V)), \tag{42}$$

$$f_{av}^{FI} = f_{av}^* + O(1/V), \tag{43}$$

$$P_{drop} = O(1/V^{c_0 \log(V)}), \tag{44}$$

where $c_0 = \Theta(1)$, $\overline{U}$ is the time average backlog, f_{av}^{FI} is the time average cost of FQLA-Ideal, f_{av}^* is the optimal time average cost and P_{drop} is the time average fraction of packets that are dropped in Step-II (b).

Proof: Since a steady state distribution exists for the backlog process generated by QLA, we see that $\mathcal{P}(D, m)$ in (22) represents the steady state probability of the event that the backlog vector deviates from $\boldsymbol{U}_V^*$ by distance $D + m$. Now since $\boldsymbol{W}(t)$ can be viewed as a backlog process generated by QLA, with $\boldsymbol{W}(0) = \boldsymbol{\mathcal{W}}$ instead of $\boldsymbol{0}$, we see from the proof of Theorem 1 that Theorem 1 and 2 hold for $\boldsymbol{W}(t)$, and by [7], QLA based on $\boldsymbol{W}(t)$ achieves an average cost of $f_{av}^* + O(1/V)$. Hence by Theorem 2, there exist constants $D_1, K_1, c_1^* = \Theta(1)$ so that: $\mathcal{P}^{(r)}(D_1, cK_1\log(V)) \leq \frac{c_1^*}{V^c}$. By the definition of $\mathcal{P}^{(r)}(D_1, cK_1\log(V))$, this implies that in steady state:

$$Pr\{W_j(t) > U_{Vj}^* + D_1 + m\} \leq c_1^* e^{-\frac{m}{K_1}},$$

Now let: $Q_j(t) = \max[W_j(t) - U_{Vj}^* - D_1, 0]$. We see that $Pr\{Q_j(t) > m\} \leq c_1^* e^{-\frac{m}{K_1}}$, $\forall m \geq 0$. We thus have $\overline{Q_j} = O(1)$, where $\overline{Q_j}$ is the time average value of $Q_j(t)$. Now it is easy to see by (40) and (41) that $U_j(t) \leq Q_j(t) + \log^2(V) + D_1 + \delta_{max}$ for all t. Thus (42) follows since for a large V:

$$\overline{U_j} \leq \overline{Q_j} + \log^2(V) + D_1 + \delta_{max} = \Theta(\log^2(V)).$$

Now consider the average cost. To save space, we use FI for FQLA-Ideal. From above, we see that QLA based on $\boldsymbol{W}(t)$ achieves an average cost of $f^*_{av} + O(1/V)$. Thus it suffices to show that FQLA-Ideal performs almost the same as QLA based on $\boldsymbol{W}(t)$. First we have for all $t \geq 1$ that:

$$\frac{1}{t}\sum_{\tau=0}^{t-1} f^{FI}(\tau) = \frac{1}{t}\sum_{\tau=0}^{t-1} f^{FI}(\tau)1_{E(\tau)} + \frac{1}{t}\sum_{\tau=0}^{t-1} f^{FI}(\tau)1_{E^c(\tau)},$$

where $1_{E(\tau)}$ is the indicator function of the event $E(\tau)$, $E(\tau)$ is the event that FQLA-Ideal performs the same action as QLA at time τ, and $1_{E^c(\tau)} = 1 - 1_{E(\tau)}$. Taking expectation on both sides and using the fact that when FQLA-Ideal takes the same action as QLA, $f^{FI}(\tau) = f^{QLA}(\tau)$, we have:

$$\begin{aligned}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{f^{FI}(\tau)\big\} \quad &\leq \quad \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{f^{QLA}(\tau)1_{E(\tau)}\big\} \\ &\quad + \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{\delta_{max}1_{E^c(\tau)}\big\}.\end{aligned}$$

Taking the limit as t goes to infinity on both sides and using $f^{QLA}(\tau)1_{E(\tau)} \leq f^{QLA}(\tau)$, we get:

$$\begin{aligned} f^{FI}_{av} \quad &\leq \quad f^{QLA}_{av} + \delta_{max}\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{1_{E^c(\tau)}\big\} \\ &= \quad f^{QLA}_{av} + \delta_{max}\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}Pr\{E^c(\tau)\}. \qquad (45)\end{aligned}$$

However, $E^c(\tau)$ is included in the event that there exists a j such that $W_j(\tau) < \mathcal{W}_j$. Therefore by (38) in Theorem 2, for a large V such that $\frac{1}{2}\log^2(V) \geq D_1$ and $\log(V) \geq 8K_1$,

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}Pr\{E^c(\tau)\} \quad &\leq \quad \mathcal{P}^{(r)}(D_1, \log^2(V) - D_1) \\ &= \quad O(c^*_1/V^{\frac{1}{2K_1}\log(V)}) \\ &= \quad O(1/V^4). \qquad (46)\end{aligned}$$

Using this fact in (45), we obtain:

$$f^{FI}_{av} \quad = \quad f^{QLA}_{av} + O(\delta_{max}/V^4) = f^*_{av} + O(1/V),$$

where the last equality holds since $f^{QLA}_{av} = f^*_{av} + O(1/V)$. This proves (43). (44) follows since packets are dropped at time τ only if $E^c(\tau)$ happens, thus by (46), the fraction of time when packet dropping happens is $O(1/V^{c_0\log(V)})$ with $c_0 = \frac{1}{2K_1} = \Theta(1)$, and each time no more than $\sqrt{r}B$ packets can be dropped. ■

D. The FQLA-General algorithm

Now we describe the FQLA algorithm without any a-priori knowledge of $\boldsymbol{U}^*_V$, called FQLA-General. FQLA-General first runs the system for a long enough time T, such that the system enters its steady state. Then it chooses a sample of the queue vector value to estimate $\boldsymbol{U}^*_V$ and uses that to decide the number of place holder bits.

FQLA-General:

(I) *Determining place-holder bits:*

a) Choose a large time T (See Section VI-E for the size of T) and initialize $\boldsymbol{W}(0) = \boldsymbol{0}$. Run the QLA algorithm with parameter V, at every time slot t, update $\boldsymbol{W}(t)$ according to the QLA algorithm and obtain $\boldsymbol{W}(T)$.

b) For each queue j, define:

$$\mathcal{W}_j = \max\big[W_j(T) - \log^2(V), 0\big], \qquad (47)$$

as the number of *place-holder bits*.

(II) *Place-holder-bit based action:* same as FQLA-Ideal.

The performance of FQLA-General is summarized as follows:

***Theorem** 5:* Assume the conditions in Theorem 4 hold and the system is in steady state at time T, then under FQLA-General with a sufficiently large V, with probability $1 - O(\frac{1}{V^4})$: (a) $\overline{U} = O(\log^2(V))$, (b) $f^{FG}_{av} = f^*_{av} + O(1/V)$, and (c) $P_{drop} = O(1/V^{c_0\log(V)})$, where $c_0 = \Theta(1)$ and f^{FG}_{av} is the time average cost of FQLA-General.

Proof: We will show that with probability of $1 - O(\frac{1}{V^4})$, $\mathcal{W}_j$ is close to $\max[U^*_{Vj} - \log^2(V), 0]$. The rest can then be proven similarly as in the proof of Theorem 4.

For each queue j, define:

$$v^+_j = U^*_{Vj} + \frac{1}{2}\log^2(V), \quad v^-_j = \max\big[U^*_{Vj} - \frac{1}{2}\log^2(V), 0\big].$$

Note that v^-_j is defined with a $\max[\,]$ operator. This is due to the fact that U^*_{Vj} can be zero. As in (46), we see that by Theorem 2, there exists $D_1 = \Theta(1), K_1 = \Theta(1)$ such that if V is such that $\frac{1}{4}\log^2(V) \geq D_1$ and $\log(V) \geq 16K_1$, then:

$$\begin{aligned} Pr\big\{\exists\, j,\, W_j(T) \notin [v^-_j, v^+_j]\big\} \quad &\leq \quad \mathcal{P}^{(r)}(D_1, \frac{1}{2}\log^2(V) - D_1) \\ &= \quad O(1/V^4)\end{aligned}$$

Thus we see that $Pr\big\{W_j(T) \in [v^-_j, v^+_j]\,\forall j\big\} = 1 - O(1/V^4)$, which implies:

$$Pr\big\{\mathcal{W}_j \in [\hat{v}^-_j, \hat{v}^+_j] \quad \forall j\big\} = 1 - O(1/V^4).$$

where $\hat{v}^+_j = \max\big[U^*_{Vj} - \frac{1}{2}\log^2(V), 0\big]$ and $\hat{v}^-_j = \max\big[U^*_{Vj} - \frac{3}{2}\log^2(V), 0\big]$. Hence for a large V, with probability $1 - O(\frac{1}{V^4})$: if $U^*_{Vj} > 0$, we have $U^*_{Vj} - \frac{3}{2}\log^2(V) \leq \mathcal{W}_j \leq U^*_{Vj} - \frac{1}{2}\log^2(V)$; else if $U^*_{Vj} = 0$, we have $\mathcal{W}_j = U^*_{Vj}$. The rest of the proof is similar to the proof of Theorem 4. ■

E. Practical Issues

From Lemma 1 we see that the magnitude of $\boldsymbol{U}^*_V$ can be $\Theta(V)$. This means that T in FQLA-General may need to be $\Omega(V)$, which is not very desirable when V is large. We can instead use the following heuristic method to accelerate the process of determining $\boldsymbol{\mathcal{W}}$: For every queue j, guess a very large $\mathcal{W}_j$. Then start with this $\boldsymbol{\mathcal{W}}$ and run the QLA algorithm for some T_1, say $\sqrt{V}$ slots. Observe the resulting backlog process. Modify the guess for each queue j using a bisection algorithm until a proper $\boldsymbol{\mathcal{W}}$ is found, i.e. when running QLA from that value, we observe fluctuations of $W_j(t)$ around $\mathcal{W}_j$ instead of a nearly constant increase or decrease for all j. Then let $\mathcal{W}_j = \max[\mathcal{W}_j - \log^2(V), 0]$ be the number of place-holder bits of queue j. To further reduce the error probability, one can repeat Step-I (a) multiple times and use the average value as $\boldsymbol{W}(T)$.

Note that even though results in Theorem 4 and 5 assume a large V, in practice, the V value may not have to be very large (See Section VII for an example).

VII. Simulation

In this section we provide simulation results for the FQLA algorithms. We consider a five queue system that extends the example in Section III-D. In this case $r = 5$. The system is shown in Fig. 4. The goal is to perform power allocation at each node so as to support the arrival with minimum energy expenditure.

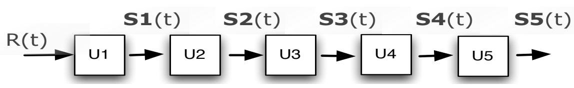

Fig. 4. A five queue system

In this example, the random network state $S(t)$ is the vector containing the random arrivals $R(t)$ and the channel states $S_i(t)$, $i = 1, ..., 5$. Similar as in Section III-D, we have:

$$\begin{aligned}\boldsymbol{A}(t) &= (R(t), \mu_1(t), \mu_2(t), \mu_3(t), \mu_4(t))^T,\\ \boldsymbol{\mu}(t) &= (\mu_1(t), \mu_2(t), \mu_3(t), \mu_4(t), \mu_5(t))^T,\end{aligned}$$

i.e., $A_1(t) = R(t)$, $A_i(t) = \mu_{i-1}(t)$ for $i \geq 2$, where $\mu_i(t)$ is the service rate obtained by queue i at time t. $R(t)$ is 0 or 2 with probabilities $\frac{3}{8}$ and $\frac{5}{8}$, respectively. $S_i(t)$ can be "Good" or "Bad" with equal probabilities for $1 \leq i \leq 5$. When the channel is good, one unit of power can serve two packets; otherwise one unit of power can serve only one packet. We assume all channels can be activated at the same time without affecting others. It can be verified that $\boldsymbol{U}_V^* = (5V, 4V, 3V, 2V, V)^T$ is unique. In this example, the backlog vector process evolves as a Markov chain with countably many states. Thus one can show that there exists a stationary distribution for the backlog vector under QLA.

We simulate FQLA-Ideal and FQLA-General with $V = 50, 100, 200, 500, 1000$ and 2000. We run each case for 5×10^6 slots under both algorithms. For FQLA-General, we use $T = 50V$ in Step-I and repeat Step-I 100 times and use their average as $\boldsymbol{W}(T)$. It is easy to see from the left plot in Fig. 5 that the average queue sizes under both FQLAs are always close to the value $5\log^2(V)$ ($r = 5$). From the middle plot we also see that the percentage of packets dropped decreases rapidly and gets below 10^{-4} when $V \geq 500$ under both FQLAs. These plots show that in practice, V may not have to be very large for Theorem 4 and 5 to hold. The right plot shows a sample $(W_1(t), W_2(t))$ process for a 10^5-slot interval under FQLA-Ideal with $V = 1000$, considering only the first two queues of Fig. 4 for this example. We see that during this interval, $(W_1(t), W_2(t))$ always remains close to $(U_{V1}^*, U_{V2}^*) = (5V, 4V)$, and $W_1(t) \geq \mathcal{W}_1 = 4952$, $W_2(t) \geq \mathcal{W}_2 = 3952$. For all V values, the average power expenditure is very close to 3.75, which is the optimal energy expenditure, and the average of $\sum W_j(t)$ is very close to $15V$ (plots omitted for brevity).

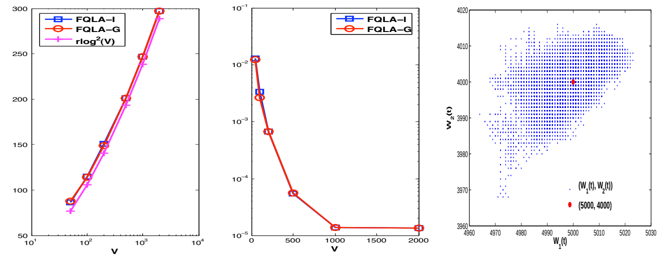

Fig. 5. FQLA-Ideal performance: Left - Average queue size; Middle - Percentage of packets dropped; Right - Sample $(W_1(t), W_2(t))$ process for $t \in [10000, 110000]$ and $V = 1000$ under FQLA-Ideal.

VIII. Lagrange Multiplier: "Shadow Price" and "Network Gravity"

It is well known that Lagrange Multipliers can play the role of "shadow prices" to regulate flows in many flow-based problems with different objectives, e.g., [16]. This important feature has enabled the development of many distributed algorithms in resource allocation problems, e.g., [17]. However, a problem of this type typically requires data transmissions to be represented as flows. Thus in a network that is discrete in nature, e.g., time slotted or packetized transmission, a rate allocation solution obtained by solving such a flow-based problem does not immediately specify a scheduling policy.

Recently, several Lyapunov algorithms have been proposed to solve utility optimization problems under discrete network settings. In these algorithms, backlog vectors act as the "gravity" of the network and allow optimal scheduling to be built upon them. It is also revealed in [18] that QLA is closely related to the dual subgradient method and backlogs play the same role as Lagrange multipliers in a time invariant network. Now we see by Theorem 1 that the backlogs indeed play the same role as Lagrange multipliers even under a more general stochastic network.

In fact, the backlog process under QLA can be closely related to a sequence of updated Lagrange multipliers under a subgradient method. Consider the following important variant of OSM, called the randomized incremental subgradient method (RISM) [12], which makes use of the separable nature of (13) and solves the dual problem (11) as follows:

RISM: Initialize $\boldsymbol{U}(0)$; at iteration t, observe $\boldsymbol{U}(t)$, choose a random state $S(t) \in \mathcal{S}$ according to some probability law. (1) If $S(t) = s_i$, find $x_{\boldsymbol{U}}^{(s_i)} \in \mathcal{X}^{(s_i)}$ that solves the following:

$$\begin{aligned}\min \quad & Vf(s_i, x) + \sum_j U_j(t)\big[g_j(s_i, x) - b_j(s_i, x)\big]\\ s.t. \quad & x \in \mathcal{X}^{(s_i)}.\end{aligned} \tag{48}$$

(2) Using the $x_{\boldsymbol{U}}^{(s_i)}$ found, update $\boldsymbol{U}(t)$ according to: [3]

$$U_j(t+1) = \max\left[U_j(t) - \alpha^t b_j(s_i, x_{\boldsymbol{U}}^{(s_i)}), 0\right] + \alpha^t g_j(s_i, x_{\boldsymbol{U}}^{(s_i)}).$$

As an example, $S(t)$ can be chosen by independently choosing $S(t) = s_i$ with probability p_{s_i} every time slot. In this

[3]Note that this update rule is different from RISM's usual rule, i.e., $U_j(t+1) = \max\left[U_j(t) - \alpha^t b_j(s_i, x) + \alpha^t g_j(s_i, x), 0\right]$, but it almost does not affect the performance of RISM.

case $S(t)$ will be i.i.d.. Note that in the stochastic problem, a network state s_i is chosen randomly by nature as the physical system state at time t; while here a state is chosen artificially by RISM according some probability law. Now we see from (8) and (48) that: given the same $\boldsymbol{U}(t)$ and s_i, *QLA and RISM choose an action in the same way*. If also $\alpha^t = 1$ for all t, and that $S(t)$ under RISM evolves according to the same probability law as $S(t)$ of the physical system, we see that *applying QLA to the network is indeed equivalent to applying RISM to the dual problem of (10), with the network state being chosen by nature, and the network backlog being the Lagrange multiplier.* Therefore, Lagrange Multipliers under such stochastic discrete network settings act as the "network gravity," thus allow scheduling to be done optimally and adaptively based on them. This "network gravity" functionality of Lagrange Multipliers in discrete network problems can thus be viewed as the counterpart of their "shadow price" functionality in the flow-based problems. Further more, the "network gravity" property of Lagrange Multipliers enables the use of place holder bits to reduce network delay in network utility optimization problems. This is a unique feature not possessed by its "price" counterpart.

Appendix A – Proof of (30)

Here we prove that for $\tilde{Y}(t)$ defined in the proof of part (b) of Theorem 1, we have for all $\tilde{Y}(t) \geq 0$:

$$\Delta(\tilde{Y}(t)) \quad \leq \quad e^{2wB} - \frac{w\eta}{2} e^{w\tilde{Y}(t)}.$$

Proof: If $\tilde{Y}(t) > B$, denote $\delta(t) = \tilde{Y}(t+1) - \tilde{Y}(t)$. It is easy to see that $|\delta(t)| \leq B$. Rewrite (29) as:

$$\Delta(\tilde{Y}(t)) \quad = \quad e^{w\tilde{Y}(t)} \mathbb{E}\big\{\big(e^{w\delta(t)} - 1\big) \mid \boldsymbol{U}(t)\big\}. \tag{49}$$

By a Taylor expansion, we have that:

$$e^{w\delta(t)} \quad = \quad 1 + w\delta(t) + \frac{w^2\delta^2(t)}{2} g(w\delta(t)), \tag{50}$$

where $g(y) = 2\sum_{k=2}^{\infty} \frac{y^{k-2}}{k!} = \frac{2(e^y - 1 - y)}{y^2}$ [19] has the following properties:

1) $g(0) = 1$; $g(y) \leq 1$ for $y < 0$; $g(y)$ is monotone increasing for $y \geq 0$;
2) For $y < 3$,

$$g(y) = 2\sum_{k=2}^{\infty} \frac{y^{k-2}}{k!} \leq \sum_{k=2}^{\infty} \frac{y^{k-2}}{3^{k-2}} = \frac{1}{1 - y/3}.$$

Thus by (50) we have:

$$e^{w\delta(t)} \quad \leq \quad 1 + w\delta(t) + \frac{w^2B^2}{2} g(wB). \tag{51}$$

Plug this into (49) and note that $\tilde{Y}(t) > B$, so by (28) we have: $\mathbb{E}\big\{\delta(t) \mid \boldsymbol{U}(t)\big\} \leq -\eta$. Hence:

$$\Delta(\tilde{Y}(t)) \quad \leq \quad e^{w\tilde{Y}(t)}\big(- w\eta + \frac{w^2B^2}{2} g(wB)\big). \tag{52}$$

Choosing $w = \frac{\eta}{B^2 + B\eta/3}$, we see that $wB < 3$, thus:

$$\frac{w^2B^2}{2} g(wB) \leq \frac{w^2B^2}{2} \frac{1}{1 - wB/3} = \frac{w\eta}{2}.$$

The last equality follows since:

$$\begin{aligned} w = \frac{\eta}{B^2 + B\eta/3} \quad &\Rightarrow \quad w(B^2 + B\eta/3) = \eta \\ &\Rightarrow \quad wB^2 = \eta - wB\eta/3 \\ &\Rightarrow \quad wB^2 \frac{1}{1 - wB/3} = \eta. \end{aligned}$$

Therefore (52) becomes:

$$\Delta(\tilde{Y}(t)) \leq -\frac{w\eta}{2} e^{w\tilde{Y}(t)} \leq e^{2wB} - \frac{w\eta}{2} e^{w\tilde{Y}(t)}. \tag{53}$$

Now if $\tilde{Y}(t) \leq B$, it is easy to see that $\Delta(\tilde{Y}(t)) \leq e^{2wB} - e^{w\tilde{Y}(t)} \leq e^{2wB} - \frac{w\eta}{2} e^{w\tilde{Y}(t)}$, since $\tilde{Y}(t+1) \leq B + \tilde{Y}(t) \leq 2B$ and $\frac{w\eta}{2} \leq 1$. Thus for all $\tilde{Y}(t) \geq 0$, (30) holds. ■

References

[1] Y.Yi and M.Chiang. Stochastic network utility maximization: A tribute to kelly's paper published in this journal a decade ago. *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 421-442, June 2008.

[2] L.Tassiulas and A.Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936-1949*, Dec. 1992.

[3] M.J.Neely. Energy optimal control for time-varying wireless networks. *IEEE Transactions on Information Theory 52(7): 2915-2934*, July 2006.

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

[5] L.Huang and M.J.Neely. The optimality of two prices: Maximizing revenue in a stochastic network. *Proc. of 45th Annual Allerton Conference on Communication, Control, and Computing (invited paper)*, Sept. 2007.

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

[7] 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. 1-144, 2006.

[8] 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.

[9] M.J.Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Nonlinear Optimization of Communication Systems*, 24(8), Aug. 2006.

[10] A.Eryilmaz and R.Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Trans. Netw.*, 15(6):1333–1344, 2007.

[11] M.J.Neely and R.Urgaonkar. Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage. *Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA (Invited paper)*, Oct. 2008.

[12] D.P.Bertsekas, A.Nedic, and A.E.Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.

[13] A.L.Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, Vol. 54, No.3, pp.203-220, 2006.

[14] L.Huang and M.J.Neely. Delay reduction via lagrange multipliers in stochastic network optimization. *arXiv:0904.3795v1*, April 2009.

[15] M.J.Neely. Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks. *Proc. of the 4th Int. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, April 2006.

[16] F.Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, vol. 8, pp. 33-37, 1997.

[17] C.Curescu and S.Nadjm-Tehrani. Price/utility-based optimized resource allocation in wireless ad hoc networks. *IEEE SECON*, 85- 95, 2005.

[18] 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*, January 2005.

[19] F.Chung and L.Lu. Concentration inequalities and martingale inequalities a survey. *Internet Math.*, 3 (2006-2007), 79–127.