SIMPLE, OPTIMAL, AND ROBUST RANDOM ACCESS

A. Hamed Mohsenian-Rad[†], Jianwei Huang[‡], Mung Chiang[#], Vincent W.S. Wong[†]

[†]Electrical and Computer Engineering Department, University of British Columbia, Vancouver, Canada
[‡] Information Engineering Department, Chinese University of Hong Kong, Hong Kong, China
[#]Electrical Engineering Department, Princeton University, Princeton, USA
E-mails: {hamed, vincentw}@ece.ubc.ca, jwhuang@ie.cuhk.edu.hk, chiangm@princeton.edu

ABSTRACT

In this paper, we propose two distributed contention-based medium access control (MAC) algorithms to solve a network utility maximization (NUM) problem in wireless ad hoc networks. Most of the previous NUM-based random access algorithms have one or more of the following performance bottlenecks: (1) extensive signaling among nodes, (2) synchronous updates of contention probabilities, (3) small update stepsizes to ensure convergence but with typically slow speed, and (4) supporting a limited range of utility functions under which the NUM is shown to be convex. Our algorithms overcome these bottlenecks in all four aspects. First, only limited message passing among nodes is required. Second, fully asynchronous updates of contention probabilities are allowed. Furthermore, our algorithms are robust to arbitrary large message passing delays and message loss. Third, we do not utilize any stepsize during updates, thus our algorithms can achieve faster convergence. Finally, our algorithms have provable convergence, optimality, and robustness properties under a wider range of utility functions, even if the NUM problem is non-convex.

I. INTRODUCTION

There are two major types of wireless medium access control (MAC) protocols: scheduling-based and contention-based. In general, the contention-based protocols are more scalable and inherently more flexible, but they typically have poor performance due to insufficient feedback. For example, in IEEE 802.11 distributed coordination function [1], a node updates its transmission probability based on the *binary* feedback of its data transmission: *success* or *failure*. This leads to low throughput, unfair resource allocation, and unstable equilibrium [2].

In this paper, we design distributed contention-based random MAC algorithms through the framework of network utility maximization (NUM), where nodes *randomly* and *distributively* access the shared channel with certain transmission probabilities. Several related algorithms that are also proposed based on the same NUM

framework include [3], [4]. They have various performance bottlenecks due to one or more of the following: (1) extensive message passing among nodes, (2) synchronous updates of contention probabilities that require homogeneous computational capabilities among nodes, (3) small update stepsizes to guarantee convergence with typically slow speed, and (4) supporting only a limited range of utility functions due to non-convexity.

Our proposed algorithms overcome the performance bottlenecks of previous proposed NUM-based random access algorithms in all four aspects. First, they only require limited message passing (i.e., signalling) among nodes. Based on the messages from other nodes, each node updates its persistent probabilities by solving a *local* and *myopic* optimization problem in an attempt to maximize the *total* network utility. Compared to the NUM-based random access algorithm in [3], our algorithms can reduce the total signalling overhead by more than a factor of 10. Second, our algorithms allow fully asynchronous updates of messages and contention probabilities. They can tolerate arbitrary large and finite asynchronism and message delays and are also robust to message losses. For example, even when the packet loss rate of the underlying communication channel is down to 0.5 (i.e., on average, half of the messages are lost), our algorithms can still achieve the optimal performance within a short time. Third, in our algorithms, nodes update their contention probabilities through *best response* updates, thus no small stepsizes are needed. This enables our algorithms to achieve a much faster convergence compared with the previously proposed subgradient-based update methods (e.g., in [3], [4]). Finally, our algorithms have provable convergence property under a wider range of utility functions, even if the NUM cannot be transformed into a convex optimization problem.

The rest of this paper is organized as follows. The system model is described in Section II. Our proposed algorithms are presented in Section III. The convergence, optimality, and robustness of our algorithms are analytically proved in Section IV. Simulation results are shown in Section V. Conclusions are discussed in Section VI.
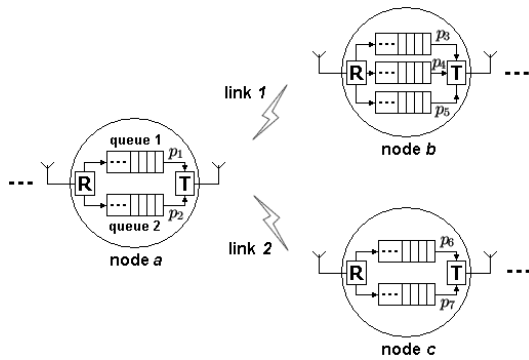
Fig. 1. A sample wireless ad-hoc network. In node a, those packets which are assigned to be sent to node b (over link 1) and node c (over link 2) are enqueued in queue 1 and queue 2, respectively. R and T boxes represent receiver and transmitter units, respectively.

II. SYSTEM MODEL

Consider a wireless ad-hoc network. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the set of nodes and $\mathcal{L} = \{1, \ldots, L\}$ denote the set of unidirectional wireless links. For each node $n \in \mathcal{N}$, we denote the set of its *outgoing* links by $\mathcal{L}_n \subset \mathcal{L}$, with size $L_n = |\mathcal{L}_n|$. Each node n has L_n separate queues, each queue holds the packets for one of its outgoing links of node n (see Fig. 1). Time is divided into equal-length slots. At each time slot, node n may choose to transmit on one of its outgoing links $i \in \mathcal{L}_n$ with a *persistent probability* p_i. The probabilities need to satisfy $\sum_{i \in \mathcal{L}_n} p_i \leq P_n^{\max} < 1$, where $P_n^{\max}$ denotes the maximum *total* persistent probability. Thus, node n may remain silent in some slots. In Fig. 1, node a has $L_a = 2$ outgoing links. In this node, those packets which are destined to node b are enqueued in queue 1. Similarly, the packets which are destined to node c are enqueued in queue 2. At each time slot, a packet from queue 1 is sent over link 1 with probability p_1, and a packet from queue 2 is sent over link 2 with probability p_2.

For each node $n \in \mathcal{N}$, if the *receiver* node of link $i \in \mathcal{L}_n$ is within the *interference range* of another node $s \in \mathcal{N} \setminus \{n\}$, then any transmission by node s (i.e., transmission on any link $j \in \mathcal{L}_s$) interferes with transmissions of link i. Those nodes which interfere with transmissions of link i are denoted by set $\mathcal{N}_i$. For each node $n \in \mathcal{N}$, let r_i denote the average data rate for link $i \in \mathcal{L}_n$, which is a function of the persistent probability vector $\boldsymbol{p} = (p_i, \forall i \in \mathcal{L})$ of all links [5]:

$$r_i(\boldsymbol{p}) = \gamma_i p_i \prod_{s \in \mathcal{N}_i} (1 - \sum_{j \in \mathcal{L}_s} p_j). \tag{1}$$

Here γ_i denotes the peak data rate for link i (i.e., the rate achieved by link i if no node in $\mathcal{N}_i$ is active). To ensure that no link is starved, for any node n and any link $i \in \mathcal{L}_n$, we require $p_i \geq P_n^{\min} > 0$ and $L_n P_n^{\min} \leq P_n^{\max}$. We define $P^{\min} = \min_{n \in \mathcal{N}} P_n^{\min}$, $P^{\max} = \max_{n \in \mathcal{N}} P_n^{\max}$, $\gamma^{\min} = \min_{i \in \mathcal{L}} \gamma_i$, and $\gamma^{\max} = \max_{i \in \mathcal{L}} \gamma_i$.

Each link $i \in \mathcal{L}$ has a *utility* which is an increasing and concave function of r_i and indicates link i's degree of satisfaction on its average rate. The utility of link i is denoted by $u(r_i(\boldsymbol{p}))$, which is also a function of $\boldsymbol{p}$. We are want to find the value of $\boldsymbol{p}$ that solves the following *network utility maximization* (NUM) problem [6]:

$$\max_{\boldsymbol{p} \in \mathcal{P}} \sum_{i \in \mathcal{L}} u(r_i(\boldsymbol{p})), \tag{2}$$

where the feasible persistent probability region is

$$\mathcal{P} = \left\{ \boldsymbol{p} : p_i \geq P_n^{\min}, \sum_{j \in \mathcal{L}_n} p_j \leq P_n^{\max}, \forall n \in \mathcal{N}, i \in \mathcal{L}_n \right\}$$

and the utility function is α-fair [7]:

$$u(r_i) = \begin{cases} (1-\alpha)^{-1} r_i^{1-\alpha}, & \text{if } \alpha \in (0,1) \cup (1,\infty), \\ \log r_i, & \text{if } \alpha = 1, \end{cases} \quad \forall i \in \mathcal{L}. \tag{3}$$

Using (3), a wide range of efficient and fair allocations can be modeled. In fact, Problem (2) reduces to throughput maximization with $\alpha \to 0$, to proportional fair allocation with $\alpha = 1$, to harmonic mean fair allocation with $\alpha = 2$, and to max-min fairness with $\alpha \to \infty$.

Although the objective function in Problem (2) is concave in link rates $\boldsymbol{r} = (r_i, \forall i \in \mathcal{L})$, it is not concave in persistent probabilities $\boldsymbol{p}$ due to the product form of the data rate in (1). Thus, finding the global optimal solution of this *non-convex* and *constrained* optimization problem is quite difficult even in a centralized fashion.

III. ALGORITHMS

In this section, we propose two distributed algorithms to solve Problem (2), one for *single-cell* topologies in Section III-A and another one for *general* topologies in Section III-B. In both algorithms, each node n performs a *myopic* and local optimization, i.e., optimizing the *total* network utility by choosing the persistent probabilities of its own outgoing links, assuming others do not change theirs. Despite the complexity of the problem, we obtained the solution of this local optimization problem in closed-form, facilitated by limited signalling among nodes and a simple local sorting procedure. Various properties of the algorithms, including convergence, optimality, and robustness, will be proved in Section IV.

A. Single-Cell Topology

We begin by considering a *single-cell* topology, where all links interfere with each other. That is, for each $n \in \mathcal{N}$ and any $i \in \mathcal{L}_n$, the interference node set $\mathcal{N}_i = \mathcal{N} \setminus \{n\}$. This models some practical networks including *wireless personal area networks* where multiple wireless devices interact with each other over short distances, as well as indoor *wireless local area networks* where several wireless devices communicate with an access point and each other (e.g., in a large conference room).

1) Node n's Local Optimization Problem: For each node n, let $\boldsymbol{p}_n = (p_i, \forall i \in \mathcal{L}_n)$ denote the persistent probabilities of its outgoing links. Also let $\boldsymbol{p}_{-n} = (p_j, \forall j \in \mathcal{L} \backslash \mathcal{L}_n)$ denote the persistent probabilities of all links *other than* the outgoing links of node n. Consider the following *local* and *myopic* optimization problem:

$$\max_{\boldsymbol{p}_n \in \mathcal{P}_n} \sum_{i \in \mathcal{L}} u(r_i(\boldsymbol{p}_n, \boldsymbol{p}_{-n})), \tag{4}$$

where for each node $n \in \mathcal{N}$ we have:

$$\mathcal{P}_n = \left\{ \boldsymbol{p}_n : \sum_{i \in \mathcal{L}_n} p_i \le P_n^{\max}, p_i \ge P_n^{\min}, \forall i \in \mathcal{L}_n \right\}, \tag{5}$$

By solving Problem (4), node n can select $\boldsymbol{p}_n$ such that the *total* network utility is maximized *assuming* that $\boldsymbol{p}_{-n}$ is fixed (i.e., none of the other nodes change their persistent probabilities). It is clear that nodes are *not* selfish in this case, and they *cooperate* with each other. This is necessary for achieving the optimal performance in a distributed fashion.

We can convert Problem (4) to an equivalent instructive representation. In particular, its objective function in the single-cell case can be written as:

$$\sum_{i \in \mathcal{L}} u(r_i(\boldsymbol{p}_n, \boldsymbol{p}_{-n})) = \left(\prod_{c \in \mathcal{N} \backslash \{n\}} (1 - \sum_{l \in \mathcal{L}_c} p_l) \right)^{1-\alpha}$$
$$(1-\alpha) \Big[\sum_{i \in \mathcal{L}_n} (\gamma_i p_i)^{1-\alpha} + (1 - \sum_{i \in \mathcal{L}_n} p_i)^{1-\alpha}$$
$$\sum_{s \in \mathcal{N} \backslash \{n\}} \sum_{j \in \mathcal{L}_s} (\gamma_j p_j)^{1-\alpha} / (1 - \sum_{l \in \mathcal{L}_s} p_l)^{1-\alpha} \Big].$$

Since the multiplicative term $(\prod_{c \in \mathcal{N} \backslash \{n\}} (1 - \sum_{l \in \mathcal{L}_c} p_l))^{1-\alpha}$ does not depend on $\boldsymbol{p}_{-n}$, Problem (4) can be *equivalently* written as:

$$\max_{\boldsymbol{p}_n \in \mathcal{P}_n} \left(\sum_{i \in \mathcal{L}_n} u(\gamma_i p_i) + v_n(\boldsymbol{p}_{-n}) u \left(1 - \sum_{i \in \mathcal{L}_n} p_i \right) \right), \tag{6}$$

where
$$v_n(\boldsymbol{p}_{-n}) =$$
$$\sum_{s \in \mathcal{N} \backslash \{n\}} \left(1 - \sum_{j \in \mathcal{L}_s} p_j \right)^{\alpha - 1} \left(\sum_{j \in \mathcal{L}_s} (\gamma_j p_j)^{1-\alpha} \right). \tag{7}$$

Since $\sum_{i \in \mathcal{L}_n} u(\gamma_i p_i)$ and $u(1 - \sum_{i \in \mathcal{L}_n} p_i)$ are strictly concave with respect to $\boldsymbol{p}_n$ and $v_n(\boldsymbol{p}_{-n})$ is independent of $\boldsymbol{p}_n$, Problem (6) is strictly concave in local variable $\boldsymbol{p}_n$. In other words, there exists a unique optimal solution of Problem (6) and thus Problem (4).

2) Closed-Form Solution of Problem (4): Next, we show how to obtain a *closed-form* optimal solution for Problem (6). Consider a node $n \in \mathcal{N}$ and the set of its outgoing links $\mathcal{L}_n$. We define a permutation, $i_1, \ldots, i_{L_n}$, of the link indices in set $\mathcal{L}_n$ such that for any j and l that satisfy $1 \le j \le l \le L_n$, we have $\sqrt[\alpha]{\gamma_{i_j}^{\alpha-1}} \le \sqrt[\alpha]{\gamma_{i_l}^{\alpha-1}}$. Thus, in case of $\alpha \ge 1$, we have $\gamma_{i_1} \le \ldots \le \gamma_{i_{L_n}}$, and in case of $\alpha \in (0, 1)$, we have $\gamma_{i_1} \ge \ldots \ge \gamma_{i_{L_n}}$. For example, let $\mathcal{L}_n = \{4, 7, 12\}$, $\gamma_4 = 18$ Mbps, $\gamma_7 = 24$ Mbps, and $\gamma_{12} = 6$ Mbps. If $\alpha \ge 1$, then: $i_1 = 12$, $i_2 = 4$, and $i_3 = 7$. If $\alpha \in (0, 1)$, then we have $i_1 = 7$, $i_2 = 4$, and $i_3 = 12$.

Define σ as the *smallest* number in set $\{0, \ldots, L_n - 1\}$ such that we have:

$$\frac{1}{P_n^{\min}} - L_n + \sigma \le \sum_{l=1}^{\sigma} \sqrt[\alpha]{\left(\frac{\gamma_{i_{\sigma+1}}}{\gamma_{i_l}} \right)^{\alpha-1}} + \sqrt[\alpha]{\gamma_{i_{\sigma+1}}^{\alpha-1} \, v_n(\boldsymbol{p}_{-n})}. \tag{8}$$

We can show that if condition (8) holds for σ, then it also holds for $\sigma + 1$. We define the set $\mathcal{B}_n = \{i_{\sigma+1}, \ldots, i_{L_n}\}$, with its size $B_n = |\mathcal{B}_n| = L_n - \sigma$. Notice that if condition (8) does not hold for any $\sigma \in \{0, \ldots, L_n - 1\}$, then we set $\mathcal{B}_n = \{\}$ and $B_n = 0$. Similarly, let ς denote the *smallest* number in the set $\{0, \ldots, L_n - 1\}$ such that:

$$P_n^{\max} / P_n^{\min} - L_n + \varsigma \le \sum_{l=1}^{\varsigma} \sqrt[\alpha]{\left(\gamma_{i_{\varsigma+1}} / \gamma_{i_l} \right)^{\alpha-1}}. \tag{9}$$

Again, we can show that if condition (9) holds for ς, then it also holds for $\varsigma + 1$. We define $\mathcal{C}_n = \{i_{\varsigma+1}, \ldots, i_{L_n}\}$, and its size $C_n = |\mathcal{C}_n| = L_n - \varsigma$. If condition (9) does not hold for any $\varsigma \in \{0, \ldots, L_n - 1\}$, then we set $\mathcal{C}_n = \{\}$ with $C_n = 0$. We also define $\mathcal{A}_n = \mathcal{B}_n \cup \mathcal{C}_n$ with size $A_n = |\mathcal{A}_n| = L_n - \kappa$ where $\kappa = \min\{\sigma, \varsigma\}$. In fact, $\mathcal{A}_n = \{i_{\kappa+1}, \ldots, i_{L_n}\}$. Depending on the value of $v_n(\boldsymbol{p}_{-n})$, either $\mathcal{A}_n = \mathcal{B}_n$ or $\mathcal{A}_n = \mathcal{C}_n$. Using $\mathcal{A}_n$, the closed-form solution of Problem (6) can be obtained as follows[1].

Theorem 1: For each node $n \in \mathcal{N}$, the unique optimal solution of Problem (6) is $\boldsymbol{p}_n^*(\boldsymbol{p}_{-n}) = \boldsymbol{f}_n(\boldsymbol{p}_{-n}) = (f_i(\boldsymbol{p}_{-n}), \forall i \in \mathcal{L}_n)$, where for each link $i \in \mathcal{L}_n$, we have:

$$f_i(\boldsymbol{p}_{-n}) = \begin{cases} P_n^{\min}, & \text{if } i \in \mathcal{A}_n, \\ \left[\frac{1}{\sqrt[\alpha]{\gamma_i^{\alpha-1}}} \frac{1 - A_n P_n^{\min}}{w_n + \sqrt[\alpha]{v_n(\boldsymbol{p}_{-n})}} \right]_{P_n^{\min}}^{\frac{P_n^{\max} - A_n P_n^{\min}}{\sqrt[\alpha]{\gamma_i^{\alpha-1} w_n}}}, & \text{if } i \in \mathcal{L}_n \backslash \mathcal{A}_n, \end{cases} \tag{10}$$

where $[x]_b^a = \max[\min[x, a], b]$ and

$$w_n = \sum_{j \in \mathcal{L}_n \backslash \mathcal{A}_n} \sqrt[\alpha]{(1/\gamma_j)^{\alpha-1}}. \tag{11}$$

The key to prove Theorem 1 is to show that $\boldsymbol{f}_n(\boldsymbol{p}_{-n})$ satisfies the *necessary* and *sufficient* Karush-Kuhn-Tucker (KKT) optimality conditions [9, pp. 244]. Since Problems (4) and (6) are equivalent, $\boldsymbol{p}_n^*(\boldsymbol{p}_{-n})$ is also the unique global optimal solution of Problem (4).

It is now clear that if node n wants to compute (10), the only information it needs from other nodes is $v_n(\boldsymbol{p}_{-n})$. If each node s announces a message m_s where:

$$m_s = \left(1 - \sum_{j \in \mathcal{L}_s} p_j \right)^{\alpha-1} \left(\sum_{j \in \mathcal{L}_s} (\gamma_j p_j)^{1-\alpha} \right), \forall s \in \mathcal{N} \backslash \{n\},$$

node n can compute $v_n(\boldsymbol{p}_{-n}) = \sum_{s \in \mathcal{N} \backslash \{n\}} m_s$. This motivates us to propose our first algorithm.

3) A Distributed MAC Algorithm: Our distributed random MAC algorithm is given in Algorithm 1, where each node $n \in \mathcal{N}$, regardless of how many links it has, announces only a *single* message m_n. All nodes choose the persistent probabilities of their outgoing links based on the received messages from other nodes. The probabilities and messages are *asynchronously* updated.

---

[1] All proofs are available in the longer version of this paper in [8].

Let $T_{n,p}$ and $T_{n,m}$ be two unbounded sets of time slots at which node n updates $\boldsymbol{p}_n$ and m_n, respectively. We assume that there is a finite constant H such that:

$$\forall t_1 \in T_{n,p} \ , \ \exists t_2 \in T_{n,p} \ , \ t_2 - t_1 \leq H, \tag{12}$$

$$\forall t_3 \in T_{n,m} \ , \ \exists t_4 \in T_{n,m} \ , \ (t_4 - t_3) + D \leq H, \tag{13}$$

where D denotes an upper bound on *communication delay*. From (12), each node updates the persistent probabilities of its outgoing links at least once every H time slots. From (13), the information used by each node is outdated by at most H time slots. We notice that H can be arbitrarily large as long as it is bounded.

Algorithm 1 - *For each node $n \in \mathcal{N}$ in a single-cell.*

1: Allocate memory for messages $\boldsymbol{m} = (m_1, \cdots, m_N)$.
2: Randomly choose transmission probabilities $p_i \geq P_n^{\min}$ for all link $i \in \mathcal{L}_n$ so that $\sum_{i \in \mathcal{L}_n} p_i \leq P_n^{\max}$.
3: Randomly choose $m_s > 0$ for all $s \in \mathcal{N}$.
4: **repeat**
5:     Transmit on link $i \in \mathcal{L}_n$ with probability p_i.
6:     **if** $t \in T_{n,p}$ **then**
7:         Set $\mathcal{A}_n = getA(n, v_n(\boldsymbol{p}_{-n}), L_n, \gamma_1, \ldots, \gamma_{L_n})$.
8:         Set $p_i = P_n^{\min}$ for all $i \in \mathcal{A}_n$.
9:         Set $p_i = \left[\frac{1}{\sqrt[\alpha]{\gamma_i^{\alpha-1}}} \frac{1 - A_n P_n^{\min}}{w_n + \sqrt[\alpha]{v_n(\boldsymbol{p}_{-n})}} \right]_{P_n^{\min}}^{\frac{P^{\max} - A_n P_n^{\min}}{\sqrt[\alpha]{\gamma_i^{\alpha-1}} w_n'}}$
10:         for all $i \in \mathcal{L}_n \backslash \mathcal{A}_n$.
11:     **if** $t \in T_{n,qm}$ **then**
12:         Set $m_n = \left(1 - \sum_{i \in \mathcal{L}_n} p_i \right)^{\alpha-1} \left(\sum_{i \in \mathcal{L}_n} \left(\frac{1}{\gamma_i p_i} \right)^{\alpha-1} \right)$.
13:         Broadcast m_n.
14:     **if** a message is received **then** Update $\boldsymbol{m}$.
15: **until** node n decides to leave the network.

1: **function** $getA(n, v_n(\boldsymbol{p}_{-n}), L_n, \gamma_1, \ldots, \gamma_{L_n})$
2:     Set $\mathcal{B}_n = \{\}$ and $\mathcal{C}_n = \{\}$.
3:     **if** $\alpha \geq 1$ **then** Set $i_1, \ldots, i_{L_m}$ so that $\gamma_{i_1} \leq \ldots \leq \gamma_{i_{L_n}}$.
4:         **else** Set $i_1, \ldots, i_{L_m}$ so that $\gamma_{i_1} \geq \ldots \geq \gamma_{i_{L_n}}$.
5:     **for** $\sigma = 0, \ldots, L_n - 1$ **do**
6:         **if** $1/P_n^{\min} - L_n + \sigma \leq \sum_{l=1}^{\sigma} \sqrt[\alpha]{\left(\gamma_{i_{k+1}} / \gamma_{i_l} \right)^{\alpha-1}} +$
7:         $\sqrt[\alpha]{\gamma_{i_{k+1}}^{\alpha-1} v_n(\boldsymbol{p}_{-n})}$ **then**
8:             Set $\mathcal{B}_n = \{i_{\sigma+1}, \ldots, i_{L_n}\}$. Break.
9:     **end for**
10:     **for** $\varsigma = 0, \ldots, L_n - 1$ **do**
11:         **if** $\frac{P_n^{\max}}{P_n^{\min}} - L_n + \varsigma \leq \sum_{l=1}^{\varsigma} \sqrt[\alpha]{\left(\gamma_{i_{k+1}} / \gamma_{i_l} \right)^{\alpha-1}}$ **then**
12:             Set $\mathcal{C}_n = \{i_{\varsigma+1}, \ldots, i_{L_n}\}$. Break.
13:     **end for**
14:     Set $\mathcal{A}_n = \mathcal{B}_n \cup \mathcal{C}_n$.
15: **return** $\mathcal{A}_n$.

Compared with the distributed MAC algorithms proposed in the literature, Algorithm 1 has several distinct features: (i) less explicit message passing is needed (e.g., in the subgradient algorithm proposed in [3], each node needs to announce *two* messages), (ii) asynchronous updates with arbitrarily finite delay, which minimizes the coordination overhead and allows maximum heterogeneity among nodes, and (iii) does not use any stepsizes, which avoids the slow convergence problem due to small stepsizes in the commonly used subgradient methods.

B. General Topology

Now consider the general case, where each node is within the interference range of an *arbitrary subset* of the other nodes. For each node $n \in \mathcal{N}$ and any of its outgoing links $i \in \mathcal{L}_n$, the set of nodes that interfere with link i is an arbitrary subset of all nodes, i.e., $\mathcal{N}_i \subseteq \mathcal{N} \backslash \{n\}$. In this case, Problem (4) can be *equivalently* written as:

$$\max_{\boldsymbol{p}_n \in \mathcal{P}_n} \left(\sum_{i \in \mathcal{L}_n} u\left(\gamma_i'(\boldsymbol{p}_{-n}) p_i \right) + v_n'(\boldsymbol{p}_{-n}) u\left(1 - \sum_{i \in \mathcal{L}_n} p_i \right) \right), \tag{14}$$

where for outgoing each link $i \in \mathcal{L}_n$, auxiliary term γ_i' is defined as $\gamma_i'(\boldsymbol{p}_{-n}) = \gamma_i \prod_{s \in \mathcal{N}_i} \left(1 - \sum_{l \in \mathcal{L}_s} p_l \right)$ and

$$v_n'(\boldsymbol{p}_{-n}) =$$
$$\sum_{s \in \mathcal{N} \backslash \{n\}} \sum_{j \in \mathcal{L}_i; n \in \mathcal{N}_j} (\gamma_j p_j \prod_{s \in \mathcal{N}_j \backslash \{n\}} (1 - \sum_{l \in \mathcal{L}_s} p_l))^{1-\alpha}.$$

Notice that $\gamma_i'(\boldsymbol{p}_{-n})$ does not represent the peak data rate of link i. The closed-form solution of Problem (14) can be obtained similarly as that of Problem (6) in the single-cell case. For each node $n \in \mathcal{N}$, we can define a permutation of link indices in set $\mathcal{L}_n$, $i_1, \ldots, i_{L_n}$, such that for any j and l that satisfy $1 \leq j \leq l \leq L_n$, we have $\sqrt[\alpha]{\gamma_{i_j}'(\boldsymbol{p}_{-n})^{\alpha-1}} \leq \sqrt[\alpha]{\gamma_{i_l}'(\boldsymbol{p}_{-n})^{\alpha-1}}$. If $\alpha \geq 1$, we have $\gamma_{i_1}' \leq \ldots \leq \gamma_{i_{L_n}}'$. If $\alpha \in (0, 1)$, we have $\gamma_{i_1}' \geq \ldots \geq \gamma_{i_{L_n}}'$. Let σ' denote the *smallest* value in $\{0, \ldots, L_n - 1\}$ so that:

$$\frac{1}{P_n^{\min}} - L_n + \sigma' \leq \sum_{l=1}^{\sigma'} \sqrt[\alpha]{(\gamma_{i_{\sigma'+1}}'(\boldsymbol{p}_{-n}) / \gamma_{i_l}'(\boldsymbol{p}_{-n})^{\alpha-1}}$$
$$+ \sqrt[\alpha]{\gamma_{i_{\sigma'+1}}'^{\alpha-1}(\boldsymbol{p}_{-n}) v_n'(\boldsymbol{p}_{-n})}. \tag{15}$$

Also let ς' denote the *smallest* value in set $\{0, \ldots, L_n - 1\}$ such that we have:

$$\frac{P_n^{\max}}{P_n^{\min}} - L_n + \varsigma' \leq \sum_{l=1}^{\varsigma'} \sqrt[\alpha]{\left(\gamma_{i_{\varsigma'+1}}'(\boldsymbol{p}_{-n}) / \gamma_{i_l}'(\boldsymbol{p}_{-n}) \right)^{\alpha-1}}. \tag{16}$$

We define $\mathcal{B}_n' = \{i_{\sigma'+1}, \ldots, i_{L_n}\}$, with its size $B_n' = |\mathcal{B}_n'| = L_n - \sigma'$. If (15) does not hold for any $\sigma' \in \{0, \ldots, L_n - 1\}$, then we set $\mathcal{B}_n' = \{\}$ with $B_n' = 0$. Similarly, we define $\mathcal{C}_n' = \{i_{\varsigma'+1}, \ldots, i_{L_n}\}$, with its size $C_n' = |\mathcal{C}_n'| = L_n - \varsigma'$. If condition (16) does not hold for any $\varsigma' \in \{0, \ldots, L_n - 1\}$, then we set $\mathcal{C}_n' = \{\}$ with $C_n' = 0$. Given $\mathcal{B}_n'$ and $\mathcal{C}_n'$, we define $\mathcal{A}_n' = \mathcal{B}_n' \cup \mathcal{C}_n'$ with its size $A_n' = |\mathcal{A}_n'| = L_n - \kappa'$ where $\kappa' = \min\{\sigma', \varsigma'\}$.

Theorem 2: For each node $n \in \mathcal{N}$, the unique glonal optimal solution of Problem (14) is $\boldsymbol{p}_n^*(\boldsymbol{p}_{-n}) = \boldsymbol{f}_n'(\boldsymbol{p}_{-n}) = (f_i'(\boldsymbol{p}_{-n}), \ \forall i \in \mathcal{L}_n)$, where for each link $i \in \mathcal{L}_i$, function $f_i(\boldsymbol{p}_{-n})$ is defined as:

$$f_i'(\boldsymbol{p}_{-n}) =$$
$$\begin{cases} P_n^{\min}, & \text{if } i \in \mathcal{A}_n', \\ \left[\frac{1}{\sqrt[\alpha]{\gamma_i'(\boldsymbol{p}_{-n})^{\alpha-1}}} \frac{1 - A_n' P_n^{\min}}{w_n'(\boldsymbol{p}_{-n}) + \sqrt[\alpha]{v_n'(\boldsymbol{p}_{-n})}} \right]_{P_n^{\min}}^{\frac{P_n^{\max} - A_n' P_n^{\min}}{\sqrt[\alpha]{\gamma_i'(\boldsymbol{p}_{-n})^{\alpha-1} w_n'(\boldsymbol{p}_{-n})}}}, & \text{otherwise,} \end{cases}$$
(17)

with $w_n'(\boldsymbol{p}_{-n}) = \sum_{j \in \mathcal{L}_n \setminus \mathcal{A}_n'} \sqrt[\alpha]{\left(1/\gamma_j'(\boldsymbol{p}_{-n}) \right)^{\alpha-1}}$.

Eq. (17) provides optimal solution for Problem (4) in the general topology case, which includes the single-cell case as a special case. We define node s' *messages* as:

$$q_s = 1 - \sum_{j \in \mathcal{L}_j} p_j, \tag{18}$$

and

$$m_{s,n} = \sum_{j \in \mathcal{L}_s : n \in \mathcal{N}_j} 1/(\gamma_j p_j \prod_{c \in \mathcal{N}_j \setminus \{n\}} q_c)^{\alpha-1}. \tag{19}$$

Then $v_n'(\boldsymbol{p}_{-n}) = \sum_{s \in \mathcal{N} \setminus \{n\}} m_{s,n}$ and $\gamma_i'(\boldsymbol{p}_{-n}) = \gamma_i \prod_{s \in \mathcal{N}_i} q_s$ for all $i \in \mathcal{L}_n$. Message q_s denotes the probability that node s remains silent at a time slot. Note that for each node $n \neq s$, if there does not exist any $j \in \mathcal{L}_s$ such that $n \in \mathcal{N}_j$, then $m_{s,n} = 0$ (i.e., node n does not cause interference to any outgoing link of node s).

Algorithm 2 - *For each node $n \in \mathcal{N}$ in a general (i.e., arbitrary) topology.*

1: Allocate memory for messages $\boldsymbol{m} = (m_{1,n}, \dots, m_{N,n})$ and $\boldsymbol{q} = (q_1, \dots, q_N)$.
2: Randomly choose $p_i \geq P_n^{\min}$ for all links $i \in \mathcal{L}_n$ so that $\sum_{i \in \mathcal{L}_n} p_i \leq P_n^{\max}$.
3: Randomly choose $m_{s,n} > 0$ and $q_s \in (0,1)$ for $s \in \mathcal{N}$.
4: **repeat**
5:     Transmit on link $i \in \mathcal{L}_n$ with probability p_i.
6:     **if** $t \in T_{n,p}$ **then**
7:         Set $\mathcal{A}_n' = getA(n, v_n'(\boldsymbol{p}_{-n}), L_n, \gamma_1', \dots, \gamma_{L_n}')$.
8:         Set $p_i = P_n^{\min}$ for all $i \in \mathcal{A}_n'$.
9:         Set $p_i = \left[\frac{1}{\sqrt[\alpha]{\gamma_i'^{\alpha-1}}} \frac{1 - A_n' P_n^{\min}}{w_n' + \sqrt[\alpha]{v_n'(\boldsymbol{p}_{-n})}} \right]_{P_n^{\min}}^{\frac{P_n^{\max} - A_n' P_n^{\min}}{\sqrt[\alpha]{\gamma_i'^{\alpha-1} w_n'}}}$
10:         for all $i \in \mathcal{L}_n \setminus \mathcal{A}_n'$.
11:     **if** $t \in T_{n,m}$ **then**
12:         Set $q_n = 1 - \sum_{i \in \mathcal{L}_n} p_i$.
13:         Set $m_{s,n} = \sum_{j \in \mathcal{L}_n : n \in \mathcal{N}_j} \frac{1}{(\gamma_j p_j \prod_{s \in \mathcal{N}_j \setminus \{n\}} q_s)^{\alpha-1}}$
14:         for all $s \in \mathcal{N} \setminus \{n\}$.
15:         Inform $m_{n,s}$ to all $s \in \cup_{i \in \mathcal{L}_n} \mathcal{N}_i$.
16:         Inform q_n to all $s \neq n$ if $\exists j \in \mathcal{L}_s$ with $n \in \mathcal{N}_j$.
17:     **if** a message is received **then** Update $\boldsymbol{m}$ and $\boldsymbol{q}$.
18: **until** node n decides to leave the network.

Our second proposed algorithm works for any general topology and is shown in Algorithm 2. In this algorithm, each node $n \in \mathcal{N}$ informs $m_{n,s}$ to all nodes s whose transmissions interfere with transmissions of at least one of the outgoing links of node n. It also informs q_n to all nodes s whose outgoing transmissions is interfered by transmissions from node n. All nodes then choose the persistent probabilities of their outgoing links based on the received messages from other nodes. In Algorithm 2, $T_{n,p}$ and $T_{n,qm}$ are two unbounded sets of time slots at which node n updates $\boldsymbol{p}_n$ and announces q_n, and $m_{n,s}$ for all $s \neq n$, respectively. We will show in Section IV-B that for *any* topology, the fixed point of Algorithm 2 also corresponds to the global optimal solution of the *non-convex* Problem (2) under proper technical conditions.

IV. CONVERGENCE, OPTIMALITY, AND ROBUSTNESS

A. Algorithm 1: Single-Cell Topology

Here we first study Algorithm 1 which was proposed to solve Problem (4) in a single-cell topology. We first define $\boldsymbol{f}(\boldsymbol{p}) = (\boldsymbol{f}_n(\boldsymbol{p}), \forall\, n \in \mathcal{N})$, where $\boldsymbol{f}_n(\boldsymbol{p})$ is as in (10) for each node n. Recall that a fixed point of mapping $\boldsymbol{f}(\boldsymbol{p})$ is also a fixed point of Algorithm 1.

Theorem 3: Assume that $\boldsymbol{f}(\boldsymbol{p})$ has a unique fixed point $\boldsymbol{p}^\star$. Starting from any initial point $\boldsymbol{p} \in \mathcal{P}$, Algorithm 1 globally converges to $\boldsymbol{p}^\star$.

The key to prove Theorem 3 is to show that mapping $\boldsymbol{f}(\boldsymbol{p})$ is a *monotone mapping*, thus, *asynchronous convergence theorem* [10, pp. 431] is applicable. Next, we show that not only Algorithm 1 has a unique fixed point under mild technical conditions, the fixed point is the global optimal solution of Problem (2).

Let $\mathcal{F}$ denote the set of fixed points of Algorithm 1. For each $\boldsymbol{p}^\star \in \mathcal{F}$ and any link $i \in \mathcal{L}_n$, $p_i^\star = f_i(\boldsymbol{p}_{-n}^\star)$. We also let $\mathcal{S}$ denote the set of *stationary points* [11, pp. 194] of Problem (2). Note that all local (and global) optimal solutions of Problem (2) belong to set $\mathcal{S}$.

Theorem 4: $\mathcal{F} = \mathcal{S}$.

From Theorems 3 and 4, we have:

Corollary 1: If either $\mathcal{S}$ or $\mathcal{F}$ is a *singleton* set (i.e., it has only one element), then Algorithm 1 asynchronously and globally converges to the unique global optimal solution of non-convex optimization problem in (2).

In [3], it has been shown that the set of stationary points $\mathcal{S}$ is a singleton set for all $\alpha \geq 1$. They used logarithmic mapping and transformed Problem (2) to an equivalent convex problem and showed that it has a unique stationary point. However, this transformation does not work if $\alpha \in (0, 1)$. That is the reason the algorithm in [3] does not support the α-fair utilities with $\alpha \in (0, 1)$. Here we are able to show the following:

Theorem 5: Consider the case where $\alpha \in (0,1)$. Set $\mathcal{F}$ is a singleton set if following holds:

$$\left(\frac{1-\alpha}{\alpha}\Psi\Phi(V^{\min},V^{\max})\right)^2\left(\frac{\gamma^{\max}}{\gamma^{\min}}\Gamma\right)^{1-\alpha}\left(\Omega-\frac{1}{L/L^{\min}-1}\right)<1,$$
(20)

where $L^{\min} = \min_{n\in\mathcal{N}} L_n$, $L^{\max} = \max_{n\in\mathcal{N}} L_n$, $\Gamma = \left(P^{\max}(1-P^{\min})\right)/\left(P^{\min}(1-P^{\max})\right)$, $\Psi = L^{\max}/(1-P^{\max})+1/P^{\min}$, $\Omega = \sum_{n\in\mathcal{N}} 1/(L/L_n-1)$, $V^{\min} = (N-1)\left(\gamma^{\max}(1/P^{\min}-1)/\gamma^{\min}\right)^{\alpha-1}$, $V^{\max} = (N-1)\left(\gamma^{\min}(1/P^{\min}-1)/\gamma^{\max}\right)^{\alpha-1}$, and we have:

$$\Phi(V^{\min},V^{\max})=\begin{cases}\frac{(V^{\max})^{1/\alpha}}{(1+(V^{\max})^{1/\alpha})2}, & \text{if } V^{\max}\leq 1,\\ \frac{(V^{\min})^{1/\alpha}}{(1+(V^{\min})^{1/\alpha})2}, & \text{if } V^{\min}\geq 1, \\ 0.25, & \text{otherwise,}\end{cases}$$
(21)

The key to prove Theorem 5 is to show that if (20) holds, then $\boldsymbol{f}$ is not only a monotone mapping, but also an l_2-norm *contraction mapping*. A contraction mapping always has unique fixed point [10, pp. 183].

Theorems 3 to 5 together show that Algorithm 1 asynchronously converges to the unique global optimal solution of the Problem (2) when either $\alpha \in (0,1)$ (under condition (20)) or $\alpha \geq 1$ (with any system parameters).

B. Algorithm 2: General Topology

To analyze Algorithm 2, we first defined $\boldsymbol{f}'(\boldsymbol{p}) = (\boldsymbol{f}'_n(\boldsymbol{p}), \forall i \in \mathcal{N})$, where $\boldsymbol{f}'_n(\boldsymbol{p})$ is as in (17). We denote the set of fixed points of mapping $\boldsymbol{f}'(\boldsymbol{p})$ by $\mathcal{F}'$, which is the set of fixed points of Algorithm 2. Let $\mathcal{S}'$ denote the set of stationary points of Problem (2) in this case.

Theorem 6: $\mathcal{F}' = \mathcal{S}'$.

If $\alpha \geq 1$, then from [3, Lemma 1] we know that stationary point set $\mathcal{S}'$ is a singleton. Thus, if $\alpha \geq 1$ and Algorithm 2 converges, then it converges to the unique global optimal solution of Problem (2). On the other hand, if $\alpha \in (0,1)$, we can use the same idea of Theorem 5 and obtain sufficient conditions to assure that the stationary point set $\mathcal{S}'$ is a singleton set. However, unlike $\boldsymbol{f}(\boldsymbol{p})$, $\boldsymbol{f}'(\boldsymbol{p})$ may not be always a monotone mapping.

To prove the asynchronous convergence of Algorithm 2, we first notice that since not all links interfere with each other, for each node $n \in \mathcal{N}$ and any link $i \in \mathcal{L}_n$, function f'_i may only depend on a small subset of entries in vector $\boldsymbol{p}_{-n}$. We define set $\mathcal{X}_i$ as the *dependency set* for link i. That is, for any $i, j \in \mathcal{L}$, we have $j \in \mathcal{X}_i$ if and only if p_j appears in the formulation of f'_i. Let $X_i = |\mathcal{X}_i|$ denote the size of set $\mathcal{X}_i$. We define $X^{\max} = \max_{i\in\mathcal{L}} X_i$.

Theorem 7: For *any* general topology, Algorithm 2 globally and asynchronously converges to the unique global optimal solution of Problem (2) if we have:

$$\frac{|1-\alpha|}{\alpha} \sqrt{N} X^{\max}\Lambda \Phi(Z^{\min}, Z^{\max}) < 1,$$
(22)

where Φ is as in (21), $\Lambda = 1/P^{\min} + 1/(1 - P^{\max})$, $Z^{\min} = \left(\gamma'^{\min}/\gamma'^{\max}\right)^{1-\alpha}(L^{\min} - 1)^{\alpha}$, $Z^{\max} = \left(\gamma'^{\max}/\gamma'^{\min}\right)^{1-\alpha}((L^{\min} - 1) + \sqrt[\alpha]{(L-L^{\min})P^{\max}})^{\alpha}$, $\gamma'^{\min} = \gamma^{\min}(1 - P^{\max})^{N-1}$, $\gamma'^{\max} = \gamma^{\max}$.

The key to prove Theorem 7 is to show that $\boldsymbol{f}'(\boldsymbol{p})$ is a *weighted maximum norm contraction mapping* with *unit* weights, thus, not only the fixed point set is a singleton set, but also the *asynchronous convergence theorem* [10, pp. 431] is applicable. Note that condition (22) is a *sufficient* (but not *necessary*) condition for asynchronous convergence. Simulation results verify that Algorithm 2 converges under a wide range of system parameters.

V. SIMULATION RESULTS

In this section, we assess the performance of our algorithms. In particular, we show their advantages compared with the subgradient-based algorithm [3][2].

A. Signalling Overhead

High signalling overhead is a critical problem for algorithms which require cooperation among nodes in a wireless ad-hoc network. In this section, we compare the signalling overhead in our proposed algorithms with the subgradient-based algorithm [3]. In the simulation model, the peak transmission rates (i.e., γ_i for all $i \in \mathcal{L}$) are selected *randomly* between 6 Mbps to 54 Mbps. Utility parameter α is set to 2 which models harmonic mean fair allocation. We assume that each message value requires two bytes. Thus, the signalling overhead for each algorithm is defined as the total required message exchange (in KBytes) that the algorithm needs before it reaches the optimal solution of Problem (2). Results for single-cell and general topologies are shown in Fig. 2(a) and Fig. 2(b), respectively. We see that increasing N increases the signalling overhead. However, Algorithm 1 and 2 manage to reach the optimal solutions via much less signalling. Compared to the subgradient-based algorithm and when $N = 30$, Algorithm 1 and Algorithm 2 reduce the signalling overhead by 1120% (from 55.2 KByte to 4.5 KByte) and 810% (from 111.3 KByte to 10.8 KByte), respectively. Notice that one reason for the superiority of our algorithms is their faster convergence. In addition, Algorithm 1 reduces the message size by half, which also helps to reduce the signalling overhead.

B. Robustness

Since the communication channels are not ideal, transmitted messages by MAC protocols may be delayed or

---

[2]Further simulation results, including results on convergence and optimality as well as performance comparison with IEEE 802.11 distributed coordination function, are available in [8].
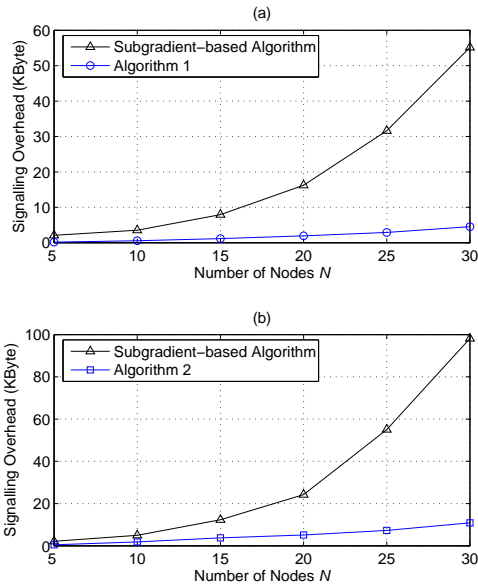
Fig. 2. Comparison between Algorithms 1 and 2 and the subgradient-based algorithm [3] in term of signalling overhead when the number of nodes varies from 5 to 30. Each point shows the average results from simulating 10 topologies. (a) Using Algorithm 1 for *single-cell* topologies. (b) Using Algorithm 2 for *general* topologies.

lost. In this section, we show that our algorithms are robust with respect to both message delay and loss. We only consider general topologies with $N = 30$ nodes. Results for single-cell topologies are similar.

First, we assume that the communication delay varies from 10 to 50 time slots. Results are shown in Fig. 3(a). We see that by increasing delay up to 50 time slots, the subgradient-based algorithm leads to 8.4% optimality loss while Algorithm 2 can always find the exact optimal solutions. Next, we consider the effect of message delay when the packet error rate varies from 0.1 (i.e. 10% of the messages are lost) to 0.5. From the results in Fig. 3(b), we see that Algorithm 2 is robust to message loss.

VI. CONCLUSION

In this paper, we designed two distributed contention-based MAC algorithms to solve network utility maximization (NUM) problem in wireless ad hoc networks. Both algorithms globally and asynchronously converge to the global optimal solution of the NUM problem under mild technical conditions on the system parameters. Besides supporting a wider range of utility functions, our proposed algorithms have several other advantages over previously proposed algorithms, including less message passing, fully asynchronous updates, robustness to message delays and losses, and faster convergence.
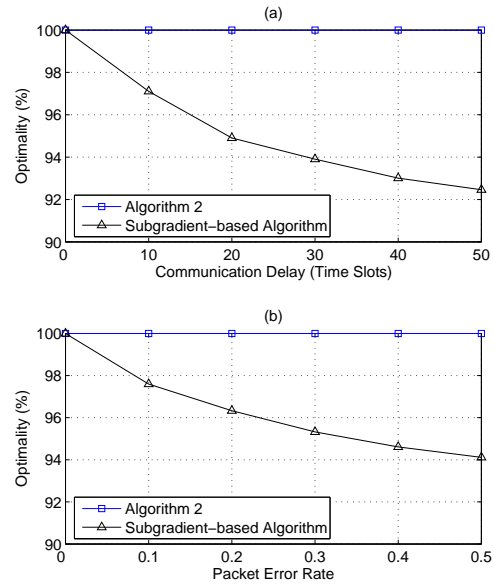
Fig. 3. Comparison between Algorithm 2 and the subgradient-based algorithm [3] in term of robustness to communication delay and message loss. (a) Optimality in percentage when maximum communication delay varies from 10 to 50 time slots, (b) Optimality in percentage when packet error rate varies from 0.1 to 0.5.

REFERENCES

[1] "IEEE 802.11a," http://standards.ieee.org/getieee802/download/802.11-1999.pdf.

[2] J. Lee, A. Tang, J. Huang, M. Chiang, and A. Calderbank, "Reverse engineering MAC: A game-theoretic model," *IEEE J. on Sel. Areas in Comm.*, vol. 6, pp. 2741–2751, Jul. 2007.

[3] J. Lee, M. Chiang, and R. Calderbank, "Utility-optimal random-access control," *IEEE Trans. on Wireless Communications*, vol. 25, pp. 1135–1147, Aug. 2007.

[4] L. Chen, S. Low, and J. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *Proc. of IEEE INFOCOM*, Miami, FL, Mar. 2005.

[5] D. P. Bertsekas and R. Gallager, *Data Communications*, 2nd ed. Prentice Hall, 1992.

[6] F. Kelly, "Charging and rate control for elastic traffic," *European Trans. on Telecommunication*, vol. 8, pp. 33–37, 1997.

[7] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. on Net.*, vol. 8, 2000.

[8] A. H. Mohsenian Rad, J. Huang, M. Chiang, and V. W. S. Wong, "Utility-optimal random access," *Technical Report, University of British Columbia,* http://www.ece.ubc.ca/~hamed/uora.pdf.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[11] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Sci., 2004.