# A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks

Libin Jiang and Jean Walrand, *Fellow, IEEE*

***Abstract*—In multihop wireless networks, designing distributed scheduling algorithms to achieve the maximal throughput is a challenging problem because of the complex interference constraints among different links. Traditional maximal-weight scheduling (MWS), although throughput-optimal, is difficult to implement in distributed networks. On the other hand, a distributed greedy protocol similar to IEEE 802.11 does not guarantee the maximal throughput. In this paper, we introduce an adaptive carrier sense multiple access (CSMA) scheduling algorithm that can achieve the maximal throughput distributively. Some of the major advantages of the algorithm are that it applies to a very general interference model and that it is simple, distributed, and asynchronous. Furthermore, the algorithm is combined with congestion control to achieve the optimal utility and fairness of competing flows. Simulations verify the effectiveness of the algorithm. Also, the adaptive CSMA scheduling is a modular MAC-layer algorithm that can be combined with various protocols in the transport layer and network layer. Finally, the paper explores some implementation issues in the setting of 802.11 networks.**

***Index Terms*—Cross-layer optimization, carrier sense multiple access (CSMA), joint scheduling and congestion control, maximal throughput.**

## I. Introduction

In MULTIHOP wireless networks, it is important to efficiently utilize the network resources and provide fairness to competing data flows. These objectives require the cooperation of different network layers. The transport layer needs to inject the right amount of traffic into the network based on the congestion level, and the MAC layer needs to serve the traffic efficiently to achieve high throughput. Through a utility optimization framework [1], this problem can be naturally decomposed into congestion control at the transport layer and scheduling at the MAC layer.

It turns out that MAC-layer scheduling is the bottleneck of the problem [1]. In particular, it is not easy to achieve the maximal throughput through distributed scheduling, which in turn prevents full utilization of the wireless network. Scheduling is challenging since the conflicting relationships between different links can be complicated.

It is well known that maximal-weight scheduling (MWS) [22] is *throughput-optimal*. That is, that scheduling can support any incoming rates within the capacity region. In MWS, time is assumed to be slotted. In each slot, a set of nonconflicting links (called an "independent set," or "IS") that have the maximal weight are scheduled, where the "weight" of a set of links is the summation of their queue lengths. (This algorithm has also been applied to achieve 100% throughput in input-queued switches [23].) However, finding such a maximal-weighted IS is NP-complete in general and is hard even for centralized algorithms. Therefore, its distributed implementation is not trivial in wireless networks.

A few recent works proposed throughput-optimal algorithms for certain interference models. For example, Eryilmaz *et al.* [3] proposed a polynomial-complexity algorithm for the "two-hop interference model".[1] Modiano *et al.* [4] introduced a gossip algorithm for the "node-exclusive model".[2] The extensions to more general interference models, as discussed in [3] and [4], involve extra challenges. Sanghavi *et al.* [5] introduced an algorithm that can approach the throughput capacity (with increasing overhead) for the node-exclusive model.

On the other hand, a number of low-complexity but suboptimal scheduling algorithms have been proposed in the literature. By using a distributed greedy protocol similar to IEEE 802.11, [8] shows that only a fraction of the throughput region can be achieved (after ignoring collisions). The fraction depends on the network topology and interference relationships. The algorithm is related to Maximal Scheduling [9], which chooses a maximal schedule among the nonempty queues in each slot. Different from Maximal Scheduling, the Longest-Queue-First (LQF) algorithm [10]–[13] takes into account the queue lengths of the nonempty queues. It shows good throughput performance in simulations. In fact, LQF is proven to be throughput-optimal if the network topology satisfies a "local pooling" condition [10], [12] or if the network is small [13]. In general topologies, however, LQF is not throughput-optimal, and the achievable fraction of the capacity region can be characterized as in [11]. Reference [14] studied the impact of such imperfect scheduling on utility maximization in wireless networks. In [16], Proutiere *et al.* developed asynchronous random-access-based scheduling algorithms that can achieve throughput performance similar to that of the Maximum Size scheduling algorithm.

Our first contribution in this paper is to introduce a *distributed* adaptive carrier sense multiple access (CSMA) algorithm for a

Manuscript received October 09, 2008; revised March 23, 2009; approved by IEEE/ACM Transactions on Networking Editor E. Modiano. This work was supported by MURI Grant BAA 07-036.18. A preliminary version of this paper was published in the Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing [2].

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: ljiang@eecs.berkeley.edu; wlr@eecs.berkeley.edu.).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2009.2035046

[1] In this model, a transmission over a link from node $m$ to node $n$ is successful iff none of the one-hop neighbors of $m$ and $n$ is in any conversation at the time.

[2] In this model, a transmission over a link from node $m$ to node $n$ is successful iff neither $m$ nor $n$ is in another conversation at the time.

general interference model. It is inspired by CSMA, but may be applied to more general resource sharing problems (i.e., not limited to wireless networks). We show that if packet collisions are ignored (as in some of the mentioned references), the algorithm can achieve maximal throughput. The optimality in the presence of collisions is studied in [30] and [31] (and also in [35] with a different algorithm). The algorithm may not be directly comparable to those throughput-optimal algorithms we have mentioned since it utilizes the carrier-sensing capability. However, it does have a few distinct features:

- Each node only uses its local information (e.g., its backlog). No explicit control messages are required among the nodes.
- It is based on CSMA random access, which is similar to the IEEE 802.11 protocol and is easy to implement.
- Time is not divided into synchronous slots. Thus, no synchronization of transmissions is needed.

In a related work, Marbach *et al.* [15] studied a model of CSMA with collisions. It was shown that under the "node-exclusive" interference model, CSMA can be made asymptotically throughput-optimal in the limiting regime of large networks with a small sensing delay. In [17], Rajagopalan and Shah independently proposed a randomized algorithm similar to ours in the context of optical networks. However, there are some notable differences (e.g., the use of Theorem 1 here). Also, utility maximization (discussed below) was not considered in [17].

Our second contribution is to combine the proposed scheduling algorithm with congestion control using a novel technique to achieve fairness among competing flows as well as maximal throughput (Sections III and IV). The performance is evaluated by simulations (Section VI). We show that the proposed CSMA scheduling is a modular MAC-layer algorithm and demonstrate its combination with optimal routing, anycast, and multicast with network coding [40]. Finally, we considered some practical issues (e.g., packet collisions) in the setting of IEEE 802.11 networks (Section VII).

There is extensive research in joint MAC and transport-layer optimization, for example [6] and [7]. Their studies have assumed the slotted-Aloha random access protocol in the MAC layer instead of the CSMA protocol we consider here. Slotted-Aloha does not need to consume power in carrier sensing. On the other hand, CSMA has a larger capacity region. (In this paper, we are primarily interested in the throughput performance.) Other related works assume physical-layer models which are quite different from ours. For example, [18] considered the CDMA interference model, and [19] focused on time-varying wireless channels.

## II. Adaptive CSMA for Maximal Throughput

### A. Interference Model

First, we describe the general interference model we will consider in this paper. Assume there are $K$ *links* in the network, where each *link* is an (ordered) transmitter–receiver pair. The network is associated with a conflict graph (or "CG") $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of vertices (each of them represents a link) and $\mathcal{E}$ is the set of edges. Two links cannot transmit at the same time (i.e., "conflict") iff there is an edge between them. Note that this framework includes the "node-exclusive model" and "two-hop interference model" mentioned as two special cases.

Assume that $G$ has $N$ independent sets ("IS," not confined to "maximal independent sets"). Denote the $i$th IS by a vector $x^i \in \{0,1\}^K$. The $k$th element of $x^i$, $x^i_k = 1$ if link $k$ is transmitting in the IS, and $x^i_k = 0$ otherwise. We also refer to $x^i$ as a *transmission state*, and $x^i_k$ as the *transmission state of link* $k$.

### B. An Idealized CSMA Protocol and the Average Throughput

We use an idealized model of CSMA as in [25]–[27]. This model makes two simplifying assumptions. First, it assumes that if two links conflict, because their simultaneous transmissions would result in incorrectly received packets, then each of the two links hears when the other one transmits. Second, the model assumes that this sensing is instantaneous. Consequently, collisions can be avoided, as we will further explain. The first assumption implies that there are no hidden nodes (HN). This is possible if the range of carrier-sensing is large enough [29].[3] The second assumption is violated in actual systems because of the finite speed of light and of the time needed to detect a received power.

There are two reasons for using this model in our context, although it makes the above simplifying assumptions about collisions and the HN problem: 1) The model is simple, tractable, and captures the essence of CSMA/CA. It is also an easier starting point before analyzing the case with collisions. Indeed, in [30], [31], we have developed a more general model that explicitly considers collisions in wireless network and extended the distributed algorithms in this paper to that case to achieve throughput-optimality. This will be further discussed in Section VII. 2) The algorithms we propose here were inspired by CSMA, but they can be applied to more general resource-sharing problems[4] that does not have the issues of collisions and HN (i.e., not limited to wireless networks).

In this subsection, assume that the links are always backlogged. If the transmitter of link $k$ senses the transmission of any conflicting link (i.e., any link $m$ such that $(k,m) \in \mathcal{E}$), then it keeps silent. If none of its conflicting links is transmitting, then the transmitter of link $k$ waits (or backs off) for a random period of time that is exponentially distributed with mean $1/R_k$ and

[3] A related problem that affects the performance of wireless networks is the exposed-node (EN) problem. EN occurs when two links could transmit together without interference, but they can sense the transmission of each other. As a result, their simultaneous transmissions are unnecessarily forbidden by CSMA. [29] proposed a protocol to address HN and EN problems in a systematic way. We assume in this paper that HN and EN are negligible with the use of such a protocol. Note that, however, although EN problem may reduce the capacity region, it does not affect the applicability of our model here since we can define an edge between two links in the CG as long as they can sense the transmission of each other, even if this results in EN.

[4] An example is the "task processing" problem described as follows. There are $K$ different types of tasks and a finite set of resources $\mathcal{B}$. To perform a type-$k$ task, one needs a subset $\mathcal{B}_k \subseteq \mathcal{B}$ of resources, and these resources are then monopolized by the task while it is being performed. Note that two tasks can be performed simultaneously iff they use disjoint subsets of resources. Clearly, this can be accommodated in our model in Section II-A by associating each type of task to a "link."

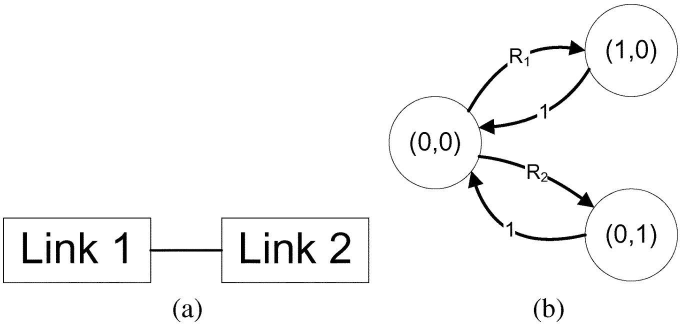


Fig. 1. Example: A conflict graph and the corresponding CSMA Markov chain. (a) Conflict graph. (b) CSMA Markov chain.

then starts its transmission.[5] If some conflicting link starts transmitting during the backoff, then link $k$ suspends its backoff and resumes it after the conflicting transmission is over. The transmission time of link $k$ is exponentially distributed with mean 1. (The assumption on the exponential distribution can be relaxed [27].) Assuming that the sensing time is negligible, given the continuous distribution of the backoff times, the probability for two conflicting links to start transmission at the same time is zero. Therefore, in the model of [25]–[27], collisions do not occur. (In Section VII, we will address the case with collisions.)

It is not difficult to see that the transitions of the transmission states form a continuous-time Markov chain, which is called the *CSMA Markov chain*. Denote link $k$'s neighboring set by $\mathcal{N}(k) := \{m : (k,m) \in \mathcal{E}\}$. If in state $x^i$, link $k$ is not active ($x_k^i = 0$) and all of its conflicting links are not active (i.e., $x_m^i = 0, \forall m \in \mathcal{N}(k)$), then state $x^i$ transits to state $x^i + \mathbf{e}_k$ with rate $R_k$, where $\mathbf{e}_k$ is the $K$-dimension vector whose $k$th element is 1 and all other elements are 0's. Similarly, state $x^i + \mathbf{e}_k$ transits to state $x^i$ with rate 1. However, if in state $x^i$, any link in its neighboring set $\mathcal{N}(k)$ is active, then state $x^i + \mathbf{e}_k$ does not exist.

Fig. 1 gives an example network whose CG is shown in (a). There are two links with an edge between them, which means that they cannot transmit together. Fig. 1(b) shows the corresponding CSMA Markov chain. State (0,0) means that no link is transmitting, state (1,0) means that only link 1 is transmitting, and (0,1) means that only link 2 is transmitting. The state (1,1) is not feasible.

Let $r_k = \log(R_k)$. We call $r_k$ the "transmission aggressiveness" (TA) of link $k$. For a given positive vector $\mathbf{r} = \{r_k, k = 1, \ldots, K\}$, the CSMA Markov chain is irreducible. Designate the stationary distribution of its feasible states $x^i$ by $p(x^i; \mathbf{r})$. We have the following result.

*Lemma 1:* ([25]–[27]) The stationary distribution of the CSMA Markov chain has the following product-form:

$$p(x^i; \mathbf{r}) = \frac{\exp\left(\sum_{k=1}^{K} x_k^i r_k\right)}{C(\mathbf{r})} \tag{1}$$

where

$$C(\mathbf{r}) = \sum_j \exp\left(\sum_{k=1}^{K} x_k^j r_k\right). \tag{2}$$

Note that the summation $\sum_j$ is over all feasible states $x^j$.

*Proof:* We verify that the distribution (1) satisfies the detailed balance equations [24]. Consider states $x^i$ and $x^i + \mathbf{e}_k$, where $x_k^i = 0$ and $x_m^i = 0, \forall m \in \mathcal{N}(k)$. From (1), we have

$$\frac{p(x^i + \mathbf{e}_k; \mathbf{r})}{p(x^i; \mathbf{r})} = \exp(r_k) = R_k$$

which is exactly the detailed balance equation between state $x^i$ and $x^i + \mathbf{e}_k$. Such relations hold for any two states that differ in only one element, which are the only pairs that correspond to nonzero transition rates. Thus, the distribution is invariant. ■

Since the detailed balance equations hold, the CSMA Markov chain is time-reversible. In fact, the Markov chain is a reversible "spatial process," and its stationary distribution (1) is a Markov random field ([24, p. 189]; [28]). (That is, the state of every link $k$ is conditionally independent of all other links, given the transmission states of its conflicting links.)

Later, we also write $p(x^i; \mathbf{r})$ as $p_i(\mathbf{r})$. These notations are interchangeable. Let $\mathbf{p}(\mathbf{r}) \in \mathcal{R}_+^N$ be the vector of all $p_i(\mathbf{r})$'s. In Fig. 1, for example, the probabilities of states (0,0), (1,0) and (0,1) are $1/(1 + R_1 + R_2)$, $R_1/(1 + R_1 + R_2)$, and $R_2/(1 + R_1 + R_2)$ in the stationary distribution.

It follows from Lemma 1 that $s_k(\mathbf{r})$, the probability that link $k$ transmits, is given by

$$s_k(\mathbf{r}) = \sum_i [x_k^i \cdot p(x^i; \mathbf{r})]. \tag{3}$$

Without loss of generality, assume that each link $k$ has a capacity of 1. That is, if link $k$ transmits data all the time (without contention from other links), then its service rate is 1 (unit of data per unit time). Then, $s_k(\mathbf{r})$ is also the *normalized* throughput (or service rate) with respect to the link capacity.

Even if the waiting time and transmission time are not exponential distributed but have the same means $1/R_k$ and 1 (in fact, as long as the ratio of their means is $1/R_k$), [27] shows that the stationary distribution (1) still holds. That is, the stationary distribution is insensitive.

### C. Adaptive CSMA for Maximal Throughput

Assume i.i.d. traffic arrival at each link $k$ with arrival rate $\lambda_k$. $\lambda_k \leq 1$ is also *normalized* with respect to the link capacity 1 and can be viewed as the fraction of time when link $k$ needs to be active to serve the arrival traffic. Denote the vector of arrival rates by $\boldsymbol{\lambda} \in R_+^K$. Further assume that $\lambda_k > 0, \forall k$ without loss of generality since the link(s) with zero arrival rate can be removed from the problem. We say that $\boldsymbol{\lambda}$ is *feasible* if and only if $\boldsymbol{\lambda} = \sum_i \bar{p}_i x^i$ for some probability distribution $\bar{\mathbf{p}} \in \mathcal{R}_+^N$ satisfying $\bar{p}_i \geq 0$ and $\sum_i \bar{p}_i = 1$. That is, $\boldsymbol{\lambda}$ is a convex combination of the ISs such that it is possible to serve the arriving traffic with some transmission schedule. Denote the set of feasible $\boldsymbol{\lambda}$

[5] If more than one backlogged links share the same transmitter, the transmitter maintains independent backoff timers for these links.

by $\bar{\mathcal{C}}$. We say that $\boldsymbol{\lambda}$ is *strictly feasible* iff it can be written as $\boldsymbol{\lambda} = \sum_i \bar{p}_i x^i$, where $\bar{p}_i > 0$ and $\sum_i \bar{p}_i = 1$. Denote the set of strictly feasible $\boldsymbol{\lambda}$ by $\mathcal{C}$. It can be shown that $\mathcal{C}$ is exactly the interior of $\bar{\mathcal{C}}$ [40].

Define the following function (the "log-likelihood function" [33] if we estimate the parameter $\mathbf{r}$ from the observation $\bar{p}_i$):

$$\begin{aligned} F(\mathbf{r};\boldsymbol{\lambda}) &:= \sum_i \bar{p}_i \log(p_i(\mathbf{r})) \\ &= \sum_i \bar{p}_i \left[\sum_{k=1}^{K} x_k^i r_k - \log(C(\mathbf{r}))\right] \\ &= \sum_k \lambda_k r_k - \log\left(\sum_j \exp\left(\sum_{k=1}^{K} x_k^j r_k\right)\right) \end{aligned}$$

where $\lambda_k = \sum_i \bar{p}_i x_k^i$ is the arrival rate at link $k$.

Consider the following optimization problem:

$$\sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda}). \tag{4}$$

Since $\log(p_i(\mathbf{r})) \leq 0$, we have $F(\mathbf{r};\boldsymbol{\lambda}) \leq 0$. Therefore, $\sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda})$ exists. Also, $F(\mathbf{r};\boldsymbol{\lambda})$ is concave in $\mathbf{r}$ [32]. We show that the following proposition holds.

*Proposition 1:* If $\sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda})$ is attainable [i.e., there exists *finite* $\mathbf{r}^* \geq 0$ such that $F(\mathbf{r}^*;\boldsymbol{\lambda}) = \sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda})$], then $s_k(\mathbf{r}^*) \geq \lambda_k,\ \forall k$. That is, the service rate is not less than the arrival rate when $\mathbf{r} = \mathbf{r}^*$.

*Proof:* Let $\mathbf{d} \geq 0$ be a vector of dual variables associated with the constraints $\mathbf{r} \geq 0$ in problem (4), then the Lagrangian is $\mathcal{L}(\mathbf{r};\mathbf{d}) = F(\mathbf{r};\boldsymbol{\lambda}) + \mathbf{d}^T\mathbf{r}$. At the optimal solution $\mathbf{r}^*$

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{r}^*;\mathbf{d}^*)}{\partial r_k} &= \lambda_k - \frac{\sum_j x_k^j \exp\left(\sum_{k=1}^{K} x_k^j r_k^*\right)}{C(\mathbf{r}^*)} + d_k^* \\ &= \lambda_k - s_k(\mathbf{r}^*) + d_k^* = 0 \end{aligned} \tag{5}$$

where $s_k(\mathbf{r})$, according to (3), is the service rate (at stationary distribution) given $\mathbf{r}$. Since $d_k^* \geq 0$, one has $s_k(\mathbf{r}^*) \geq \lambda_k$. ■

Equivalently, problem (4) is the same as minimizing the Kullback–Leibler divergence (KL divergence) between the two distributions $\bar{\mathbf{p}}$ and $\mathbf{p}(\mathbf{r})$

$$\inf_{\mathbf{r}\geq \mathbf{0}} D_{KL}(\bar{\mathbf{p}}||\mathbf{p}(\mathbf{r})) \tag{6}$$

where the KL divergence is defined as follows:

$$\begin{aligned} D_{KL}(\bar{\mathbf{p}}||\mathbf{p}(\mathbf{r})) &:= \sum_i \left[\bar{p}_i \log\left(\bar{p}_i/p_i(\mathbf{r})\right)\right] \\ &= \sum_i [\bar{p}_i \log(\bar{p}_i)] - F(\mathbf{r};\boldsymbol{\lambda}). \end{aligned}$$

That is, we choose $\mathbf{r} \geq \mathbf{0}$ such that $\mathbf{p}(\mathbf{r})$ is the "closest" to $\bar{\mathbf{p}}$ in terms of the KL divergence.

The above result is related to the theory of Markov random fields [33] in that when we minimize the KL divergence between a given joint distribution $\mathbf{p}_I$ and a product-form joint distribution $\mathbf{p}_{II}$, then depending on the structure of $\mathbf{p}_{II}$, certain marginal distributions induced by the two joint distributions are equal. In our case, the time-reversible CSMA Markov chain gives the product-form distribution. Also, the arrival rate and service rate on link $k$ are viewed as two marginal probabilities. They are not always equal, but satisfy the desired inequality in Proposition 1 due to the constraint $\mathbf{r} \geq \mathbf{0}$, which is important in our design.

The following condition, proved in Appendix A, ensures that $\sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda})$ is attainable.

*Proposition 2:* If the arrival rate $\boldsymbol{\lambda}$ is strictly feasible (i.e., $\boldsymbol{\lambda} \in \mathcal{C}$), then $\sup_{\mathbf{r}\geq 0} F(\mathbf{r};\boldsymbol{\lambda})$ is attainable.

Combining Propositions 1 and 2, we have the following.

*Theorem 1:* For any $\boldsymbol{\lambda} \in \mathcal{C}$, there exists a finite $\mathbf{r}^*$ such that $s_k(\mathbf{r}^*) \geq \lambda_k,\ \forall k$.

Remark: To see why strict feasibility is necessary, consider the network in Fig. 1. If $\lambda_1 = \lambda_2 = 0.5$ (not strictly feasible), then the service rates $s_1(\mathbf{r}) = s_2(\mathbf{r}) \to 0.5$ when $r_1 = r_2 \to \infty$, but they cannot reach 0.5 for finite values of $\mathbf{r}$.

Since $\partial F(\mathbf{r};\boldsymbol{\lambda})/\partial r_k = \lambda_k - s_k(\mathbf{r})$, a simple gradient algorithm to solve (4) is

$$r_k(j+1) = [r_k(j) + \alpha(j) \cdot (\lambda_k - s_k(\mathbf{r}(j)))]_+,\ \forall k \tag{7}$$

where $j = 0, 1, 2, \ldots$, and $\alpha(j)$ is some (small) step size. The algorithm is easy for *distributed* implementation in wireless networks because link $k$ can adjust $r_k$ based on its *local information*: arrival rate $\lambda_k$ and service rate $s_k(\mathbf{r}(j))$. (If the arrival rate is larger than the service rate, then $r_k$ should be increased, and vice versa.) Note that, however, the arrival and service rates are generally random variables in actual networks, unlike in (7).

Let link $k$ adjust $r_k$ at time $t_j$, $j = 1, 2, \ldots$. Let $t_0 = 0$ and the *update interval* $T(j) := t_j - t_{j-1}, j = 1, 2, \ldots$. Define "period $j$" as the time between $t_{j-1}$ and $t_j$, and $\mathbf{r}(j)$ as the value of $\mathbf{r}$ set at time $t_j$. Let $\lambda'_k(j)$ and $s'_k(j)$ be, respectively, the empirical average arrival rate and service rate at link $k$ between time $t_j$ and $t_{j+1}$. That is, $s'_k(j) := \int_{t_j}^{t_{j+1}} x_k(\tau)d\tau/T(j+1)$, where $x_k(\tau) \in \{0, 1\}$ is the transmission state of link $k$ at time instance $\tau$. Note that $\lambda'_k(j)$ and $s'_k(j)$ are generally random variables. We design the following distributed algorithm.

*Algorithm 1: Adjusting the TA (Transmission Aggressiveness) in CSMA:* At time $t_{j+1}$ where $j = 0, 1, 2, \ldots$, let

$$r_k(j+1) = [r_k(j) + \alpha(j) \cdot (\lambda'_k(j) - s'_k(j))]_D,\ \forall k \tag{8}$$

where $\alpha(j) > 0$ is the step size, and $[\cdot]_D$ means the projection to the set $D := [0, r_{\max}]$ where $r_{\max} > 0$. We allow $r_{\max} = +\infty$, in which case the projection is the same as $[\cdot]_+$.[6] In the next section and in Appendix B, we will discuss the convergence and stability properties of Algorithm 1 under different settings of $\alpha(j), T(j)$ and $r_{\max}$.

## D. Convergence and Stability

Reference [38] provides some stability results of the following algorithm extended from Algorithm 1. The intuition is

[6] A subtle point: If during period $j+1$, the queue of link $k'$ becomes empty, then link $k'$ still transmits dummy packets with TA $r_{k'}(j)$ until $t_{j+1}$. This ensures that the (ideal) average service rate is still $s_k(\mathbf{r}(j))$ for all $k$. (The dummy packets are counted in the computation of $s'_{k'}(j)$.)

that one can make $\mathbf{r}$ change slowly (i.e., "quasi-static") to allow the CSMA Markov chain to approach its stationary distribution (and thus obtaining good estimation of $s_k(\mathbf{r})$). This allows the separation of time scales of the dynamics of $\mathbf{r}(j)$ and the CSMA Markov chain. The extended algorithm is

$$r_k(j+1) = [r_k(j) + \alpha(j)\cdot(\lambda'_k(j) + h(r_k(j)) - s'_k(j))]_D \quad (9)$$

where $D := [0, r_{\max}]$ and the function $h(\cdot) \geq 0$. If $h(\cdot) = 0$, then algorithm (9) reduces to Algorithm 1. If $h(\cdot) > 0$, then algorithm (9) "pretends" to serve some arrival rates higher than the actual ones. In Appendix B, we state some results in [38] (which includes the detailed proofs). In summary: 1) with properly chosen decreasing step sizes and increasing update intervals (e.g., $\alpha(j) = 1/[(j+2)\log(j+2)]$, $T(j) = j+2$) and function $h(\cdot)$, and with $r_{\max} = +\infty$, the vector $\mathbf{r}(j)$ converges and the algorithm is throughput-optimal; 2) with properly chosen constant step sizes $\alpha(j) = \alpha$, $\forall j$ and update intervals $T(j) = T$, $\forall j$, one can arbitrarily approximate the maximal throughput.

In a related work [21], Liu *et al.* carried out a convergence analysis, using a differential-equation method, of a utility maximization algorithm extended from [2] (see also Section IV for the algorithm), although queueing stability was not considered in [21].

### E. Discussion

1) It has been believed that optimal scheduling is NP-complete in general. This complexity is reflected in the mixing time of the CSMA Markov chain (i.e., the time for the Markov chain to approach its stationary distribution). In [38], the upper bound used to quantify the mixing time is exponential in $K$. However, the bound may not be tight in typical wireless networks. For example, in a network where all links conflict, the CSMA Markov chain mixes much faster than the bound.
2) There is some resemblance between the above algorithm (in particular the CSMA Markov chain) and simulated annealing (SA) [20]. SA is an optimization technique that utilizes time-reversible Markov chains to find a maximum of a function. SA can be used, for example, to find the maximal-weighted IS (MWIS) that is needed in Maximal-Weight Scheduling. However, note that our algorithm does not try to find the MWIS via SA. Instead, the stationary distribution of the CSMA Markov chain with a properly chosen $\mathbf{r}^*$ is sufficient to support any $\boldsymbol{\lambda} \in \mathcal{C}$ (Theorem 1).

## III. The Primal-Dual Relationship

In the previous section, we have described the adaptive CSMA algorithm to support any strictly feasible arrival rates. For joint scheduling and congestion control, however, directly using the expression of service rate (3) will lead to a nonconvex problem. This section takes another look at the problem and also helps to avoid the difficulty.

Rewrite (4) as

$$\begin{aligned} \max_{\mathbf{r},\mathbf{h}} \quad & \left\{ \sum_k \lambda_k r_k - \log\left(\sum_j \exp(h_j)\right)\right\} \\ \text{s.t.} \quad & h_j = \sum_{k=1}^{K} x_k^j r_k, \ \forall j \\ & r_k \geq 0, \ \forall k. \end{aligned} \quad (10)$$

For each $j = 1, 2, \ldots, N$, associate a dual variable $u_j$ to the constraint $h_j = \sum_{k=1}^{K} x_k^j r_k$. Write the vector of dual variables as $\mathbf{u} \in \mathcal{R}_+^N$. Then, it is not difficult to find the dual problem of (10) as follows. (The computation was given in [41], but is omitted here due to the limit of space.)

$$\begin{aligned} \max_{\mathbf{u}} \quad & -\sum_i u_i \log(u_i) \\ \text{s.t.} \quad & \sum_i (u_i \cdot x_k^i) \geq \lambda_k, \ \forall k \\ & u_i \geq 0, \sum_i u_i = 1. \end{aligned} \quad (11)$$

where the objective function is the entropy of the distribution $\mathbf{u}$, $H(\mathbf{u}) := -\sum_i u_i \log(u_i)$.[7]

Also, if for each $k$, we associate a dual variable $r_k$ to the constraint $\sum_i (u_i \cdot x_k^i) \geq \lambda_k$ in problem (11), then one can compute that the dual problem of (11) is the original problem $\max_{\mathbf{r}\geq\mathbf{0}} F(\mathbf{r};\boldsymbol{\lambda})$. (This is shown in Appendix A as a by-product of the proof of Proposition 2.) This is not surprising since, in convex optimization, the dual problem of dual problem is often the original problem.

What is interesting is that both $\mathbf{r}$ and $\mathbf{u}$ have concrete physical meanings. We have seen that $r_k$ is the TA of link $k$. Also, $u_i$ can be regarded as the probability of state $x^i$. This observation will be useful in later sections. A convenient way to guess this is by observing the constraint $\sum_i (u_i \cdot x_k^i) \geq \lambda_k$. If $u_i$ is the probability of state $x^i$, then the constraint simply means that the service rate of link $k$, $\sum_i (u_i \cdot x_k^i)$, is larger than the arrival rate.

*Proposition 3:* Given some (finite) TAs of the links [that is, given the dual variable $\mathbf{r}$ of problem (11)], the stationary distribution of the CSMA Markov chain maximizes the partial Lagrangian $\mathcal{L}(\mathbf{u};\mathbf{r}) = -\sum_i u_i \log(u_i) + \sum_k r_k(\sum_i u_i \cdot x_k^i - \lambda_k)$ over all possible distributions $\mathbf{u}$. Also, Algorithm (7) can be viewed as a subgradient algorithm to update the dual variable $\mathbf{r}$ in order to solve problem (11).

*Proof:* Given some finite dual variables $\mathbf{r}$, a partial Lagrangian of problem (11) is

$$\mathcal{L}(\mathbf{u};\mathbf{r}) = -\sum_i u_i \log(u_i) + \sum_k r_k \left(\sum_i u_i \cdot x_k^i - \lambda_k\right). \quad (12)$$

[7] In fact, there is a more general relationship between ML estimation problem such as (4) and Maximal-Entropy problem such as (11) [33], [34]. In [41], on the other hand, problem (11) was motivated by the "statistical entropy" of the CSMA Markov chain.

Denote $\mathbf{u}^*(\mathbf{r}) = \arg\max_{\mathbf{u}} \mathcal{L}(\mathbf{u};\mathbf{r})$, where $\mathbf{u}$ is a distribution. Since $\sum_i u_i = 1$, if we can find some $w$, and $\mathbf{u}^*(\mathbf{r}) > 0$ (i.e., in the interior of the feasible region) such that

$$\frac{\partial \mathcal{L}(\mathbf{u}^*(\mathbf{r});\mathbf{r})}{\partial u_i} = -\log(u_i^*(\mathbf{r})) - 1 + \sum_k r_k x_k^i = w,\ \forall i,$$

then $\mathbf{u}^*(\mathbf{r})$ is the desired distribution. The above conditions are

$$u_i^*(\mathbf{r}) = \exp\left(\sum_k r_k x_k^i - w - 1\right),\ \forall i.\ \text{and}\ \sum_i u_i^*(\mathbf{r}) = 1.$$

By solving the two equations, we find that $w = \log[\sum_j \exp(\sum_k r_k x_k^j)] - 1$ and

$$u_i^*(\mathbf{r}) = \frac{\exp\left(\sum\limits_k r_k x_k^i\right)}{\sum\limits_j \exp\left(\sum\limits_k r_k x_k^j\right)},\ \forall i \tag{13}$$

satisfy the conditions. Note that in (13), $u_i^*(\mathbf{r})$ is exactly the stationary probability of state $x^i$ in the CSMA Markov chain given the TA $\mathbf{r}$ of all links. That is, $u_i^*(\mathbf{r}) = p(x^i;\mathbf{r}),\ \forall i$ [cf. (1)]. Thus, Algorithm (7) is a subgradient algorithm to search for the optimal dual variable. Indeed, given $\mathbf{r}$, $u_i^*(\mathbf{r})$ maximizes $\mathcal{L}(\mathbf{u};\mathbf{r})$; then, $\mathbf{r}$ can be updated by the subgradient algorithm (7), which is the deterministic version of Algorithm 1. The whole system is trying to solve problem (11) or (4). ■

Let $\mathbf{r}^*$ be the optimal vector of dual variables of problem (11). From the above computation, we see that $\mathbf{u}^*(\mathbf{r}^*) = \mathbf{p}(\mathbf{r}^*)$, the optimal solution of (11), is a product-form distribution. Also, $\mathbf{p}(\mathbf{r}^*)$ can support the arrival rates $\boldsymbol{\lambda}$ because it is feasible to (11). This is another way to look at Theorem 1.

## IV. Joint Scheduling and Congestion Control

Now, we combine congestion control with the CSMA scheduling algorithm to achieve fairness among competing flows as well as the maximal throughput. Here, the input rates are distributedly adjusted by the source of each flow.

### A. Formulation and Algorithm

Assume there are $M$ flows, and let $m$ be their index ($m = 1, 2, \ldots, M$). Define $a_{mk} = 1$ if flow $m$ uses link $k$, and $a_{mk} = 0$ otherwise. Let $f_m$ be the rate of flow $m$, and $v_m(f_m)$ be the "utility function" of this flow, which is assumed to be increasing and strictly concave. Assume all links have the same PHY data rates (it is easy to extend the algorithm to different PHY rates).

Assume that each link $k$ maintains a separate queue for each flow that traverses it. Then, the service rate of flow $m$ by link $k$, denoted by $s_{km}$, should be no less than the incoming rate of flow $m$ to link $k$. For flow $m$, if link $k$ is its first link (i.e., the source link), we say $\delta(m) = k$. In this case, the constraint is $s_{km} \geq f_m$. If $k \neq \delta(m)$, denote flow $m$'s upstream link of link $k$ by $up(k,m)$, then the constraint is $s_{km} \geq s_{up(k,m),m}$, where $s_{up(k,m),m}$ is equal to the incoming rate of flow $m$ to link $k$. We also have $\sum_i u_i x_k^i \geq \sum_{m:a_{mk}=1} s_{km},\ \forall k$, i.e., the total service rate of link $k$ is not less than the sum of all flow rates on the link.

Then, consider the following optimization problem:

$$\begin{aligned}
\max_{\mathbf{u},\mathbf{s},\mathbf{f}} \quad & -\sum_i u_i \log(u_i) + \beta \sum_{m=1}^{M} v_m(f_m) \\
\text{s.t.} \quad & s_{km} \geq 0,\ \forall k, m : a_{mk} = 1 \\
& s_{km} \geq s_{up(k,m),m},\ \forall m, k : a_{mk} = 1, k \neq \delta(m) \\
& s_{km} \geq f_m,\ \forall m, k : k = \delta(m) \\
\sum_i u_i x_k^i \geq & \sum_{m:a_{mk}=1} s_{km},\ \forall k \\
& u_i \geq 0,\ \sum_i u_i = 1
\end{aligned} \tag{14}$$

where $\beta > 0$ is a constant weighting factor.

Notice that the objective function is not exactly the total utility, but it has an extra term $-\sum_i u_i \log(u_i)$. As will be further explained in Section IV-B, when $\beta$ is large, the "importance" of the total utility dominates the objective function of (14). (This is similar in spirit to the weighting factor used in [19].) As a result, the solution of (14) approximately achieves the maximal utility. Associate dual variables $q_{km} \geq 0$ to the second and third lines of constraints of (14). Then, a partial Lagrangian (subject to $s_{km} \geq 0$, $\sum_i u_i x_k^i \geq \sum_{m:a_{mk}=1} s_{km}$ and $u_i \geq 0, \sum_i u_i = 1$) is

$$\begin{aligned}
&\mathcal{L}(\mathbf{u},\mathbf{s},\mathbf{f};\mathbf{q}) \\
&\quad = -\sum_i u_i \log(u_i) + \beta \sum_{m=1}^{M} v_m(f_m) \\
&\quad\quad + \sum_{m,k:a_{mk}=1, k\neq\delta(m)} q_{km}(s_{km} - s_{up(k,m),m}) \\
&\quad\quad + \sum_{m,k:,k=\delta(m)} q_{km}(s_{km} - f_m) \\
&\quad = -\sum_i u_i \log(u_i) \\
&\quad\quad + \beta \sum_{m=1}^{M} v_m(f_m) - \sum_{m,k:k=\delta(m)} q_{km} f_m \\
&\quad\quad + \sum_{k,m:a_{mk}=1} [s_{km} \cdot (q_{km} - q_{\text{down}(k,m),m})]
\end{aligned} \tag{15}$$

where $\text{down}(k,m)$ means flow $m$'s downstream link of link $k$ (Note that $\text{down}(\text{up}(k,m),m) = k$). If $k$ is the last link of flow $m$, then define $q_{\text{down}(k,m),m} = 0$.

First, we fix the vectors $\mathbf{u}$ and $\mathbf{q}$ and solve for $s_{km}$ in the subproblem

$$\begin{aligned}
\max_{\mathbf{s}} \quad & \sum_{k,m:a_{mk}=1} [s_{km} \cdot (q_{km} - q_{\text{down}(k,m),m})] \\
\text{s.t.} \quad & s_{km} \geq 0,\ \forall k, m : a_{mk} = 1 \\
\sum_{m:a_{mk}=1} & s_{km} \leq \sum_i (u_i \cdot x_k^i),\ \forall k.
\end{aligned} \tag{16}$$

The solution is easy to find (similar to [1] and related references therein): At link $k$, denote $z_k := \max_{m:a_{mk}=1}(q_{km} - q_{\text{down}(k,m),m})$. 1) If $z_k > 0$, then for a $m' \in \arg\max_{m:a_{mk}=1}(q_{km} - q_{\text{down}(k,m),m})$, let $s_{km'} = \sum_i (u_i x_k^i)$, and let $s_{km} = 0,\ \forall m \neq m'$. In other words, link $k$ serves a flow with the maximal back-pressure $q_{km} - q_{\text{down}(k,m),m}$. 2) If $z_k \le 0$, then let $s_{km} = 0$, $\forall m$, i.e., link $k$ does not serve any flow. Since the transmitter of link $k$ can obtain the value of $q_{\text{down}(k,m),m}$ from a one-hop neighbor (i.e., the receiver of link $k$), this algorithm is distributed. (In practice, the value of $q_{\text{down}(k,m),m}$ can be piggybacked in the ACK packet in IEEE 802.11.)

Plugging the solution of (16) back into (15), we get

$$\mathcal{L}(\mathbf{u}, \mathbf{f}; \mathbf{q}) = \left[ -\sum_i u_i \log(u_i) + \sum_k (z_k)_+ \left( \sum_i u_i x_k^i \right) \right] + \left[ \beta \sum_{m=1}^{M} v_m(f_m) - \sum_{m,k:k=\delta(m)} q_{km} f_m \right]$$

where $z_k$ is the maximal back-pressure at link $k$. Thus, a distributed algorithm to solve (14) is as follows. Denote by $Q_{km}$ the actual queue length of flow $m$ at link $k$. For simplicity, assume that $v'_m(0) \le V$, $\forall m$ for some constant $V < \infty$.

*Algorithm 2: Joint Scheduling and Congestion Control:* Initially, assume that all queues are empty (i.e., $Q_{km}(0) = 0$, $\forall k, m$), and let $q_{km}(0) = 0$, $\forall k, m$. As before, the update interval $T(j) = t_j - t_{j-1}$ and $t_0 = 0$. Here we use constant step sizes and update intervals $\alpha(j) = \alpha, T(j) = T$, $\forall j$. The variables $\mathbf{q}, \mathbf{f}, \mathbf{r}$ are iteratively updated at time $t_j$, $j = 1, 2, \ldots$. Let $\mathbf{q}(j), \mathbf{f}(j), \mathbf{r}(j)$ be their values set at time $t_j$. Denote by $s'_{km}(j)$ the empirical average service rate of flow $m$ at link $k$ in period $j+1$ (i.e., the time between $t_j$ and $t_{j+1}$).

- CSMA scheduling: In period $j+1$, link $k$ lets its TA be $r_k(j) = [z_k(j)]_+$ in the CSMA operation, where $z_k(j) = \max_{m:a_{mk}=1}(q_{km}(j) - q_{\text{down}(k,m),m}(j))$. (The rationale is that, given $\mathbf{z}(j)$, the $\mathbf{u}^*$ that maximizes $\mathcal{L}(\mathbf{u}, \mathbf{f}; \mathbf{q}(j))$ over $\mathbf{u}$ is the stationary distribution of the CSMA Markov chain with $r_k(j) = [z_k(j)]_+$, similar to the proof of Proposition 3.) Choose a flow $m' \in \arg\max_{m:a_{mk}=1}(q_{km}(j) - q_{\text{down}(k,m),m}(j))$. When link $k$ gets the opportunity to transmit: 1) if $z_k(j) > 0$, it serves flow $m'$. (Similar to Algorithm 1, the dummy packets transmitted by link $k$, if any, are counted in $s'_{km'}(j)$.); 2) if $z_k(j) \le 0$, then it transmits dummy packets. These dummy packets are not counted, i.e., let $s'_{km}(j) = 0$, $\forall m$. Also, they are not put into any actual queue at the receiver of link $k$. (A simpler alternative is that link $k$ keeps silent if $z_k(j) \le 0$. That case can be similarly analyzed following [40].)
- Congestion control: For each flow $m$, if link $k$ is its source link, the transmitter of link $k$ lets the flow rate in period $j+1$ be $f_m(j) = \arg\max_{\hat{f}_m \in [0,1]} \{\beta \cdot v_m(\hat{f}_m) - q_{km}(j) \cdot \hat{f}_m\}$. (This maximizes $\mathcal{L}(\mathbf{u}, \mathbf{f}; \mathbf{q}(j))$ over $\mathbf{f}$.)
- The dual variables $q_{km}$ (maintained by the transmitter of each link) are updated (similar to a subgradient algorithm). At time $t_{j+1}$, let $q_{km}(j+1) = [q_{km}(j) - \alpha \cdot s'_{km}(j)]_+ + \alpha \cdot s'_{up(k,m),m}(j)$ if $k \ne \delta(m)$; and $q_{km}(j+1) = [q_{km}(j) - \alpha \cdot s'_{km}(j)]_+ + \alpha \cdot f_m(j)$ if $k = \delta(m)$. (By doing this, approximately $q_{km} \propto Q_{km}$.)

*Remark 1:* As $T \to \infty$ and $\alpha \to 0$, Algorithm 2 approximates the "ideal" algorithm that solves (14), due to the convergence of the CSMA Markov chain in each period. A bound of the achievable utility of Algorithm 2, compared to the optimal total utility $\bar{W}$ defined in (17) is given in [40]. The bound, however, is not very tight: Our simulation below shows good performance without very large $\beta$, $T$, or a very small $\alpha$. Also, since $v'_m(0) \le V < \infty$, $\forall m$, it is clear through the proof of Proposition 5 that $q_{km}(j)$ is uniformly bounded for all $k, m, j$ given any $\beta$. Then, it is easy to show that the queue lengths $Q_{km}$'s are uniformly bounded at all time.

*Remark 2:* In [40], we show that by using similar techniques, the adaptive CSMA algorithm can be combined with optimal routing, anycast or multicast with network coding. So it is a modular MAC-layer protocol which can work with other protocols in the transport layer and the network layer.

### B. Approaching the Maximal Utility

We now show that the solution of (14) approximately achieves the maximal utility when $\beta$ is large. Denote the maximal total utility achievable by $\bar{W}$, i.e.,

$$\bar{W} := \max_{\mathbf{u},\mathbf{s},\mathbf{f}} \sum_m v_m(f_m) \tag{17}$$

subject to the same constraints as in (14). Assume that $\mathbf{u} = \bar{\mathbf{u}}$ when (17) is solved. Also, assume that in the optimal solution of (14), $\mathbf{f} = \hat{\mathbf{f}}$ and $\mathbf{u} = \hat{\mathbf{u}}$. We have the following bound.

*Proposition 4:* The difference between the total utility $(\sum_{m=1}^{M} v_m(\hat{f}_m))$ resulting from solving (14) and the maximal total utility $\bar{W}$ is bounded. The bound of difference decreases with the increase of $\beta$. In particular

$$\bar{W} - (K \cdot \log 2)/\beta \le \sum_m v_m(\hat{f}_m) \le \bar{W}. \tag{18}$$

*Proof:* Notice that $H(\mathbf{u}) = -\sum_i u_i \log(u_i)$, the entropy of the distribution $\mathbf{u}$, is bounded. Indeed, since there are $N \le 2^K$ possible states, one has $0 \le H(\mathbf{u}) \le \log N \le \log 2^K = K \log 2$.

Since in the optimal solution of problem (14), $\mathbf{f} = \hat{\mathbf{f}}$ and $\mathbf{u} = \hat{\mathbf{u}}$, we have $H(\hat{\mathbf{u}}) + \beta \sum_m v_m(\hat{f}_m) \ge H(\bar{\mathbf{u}}) + \beta \bar{W}$. Therefore, $\beta[\sum_m v_m(\hat{f}_m) - \bar{W}] \ge H(\bar{\mathbf{u}}) - H(\hat{\mathbf{u}}) \ge -H(\hat{\mathbf{u}}) \ge -K \cdot \log 2$. Also, clearly $\bar{W} \ge \sum_{m=1}^{M} v_m(\hat{f}_m)$, so (18) follows. ■

## V. Reducing the Queueing Delay

Consider a $\boldsymbol{\lambda} \in \mathcal{C}$ in the scheduling problem in Section II. With Algorithm 1, the long-term average service rates are in general not strictly higher than the arrival rates, so traffic suffers from queueing delay when traversing the links. To reduce the delay, consider a modified version of problem (11)

$$\begin{aligned} \max_{\mathbf{u},\mathbf{w}} \quad & -\sum_i u_i \log(u_i) + c \sum_k \log(w_k) \\ \text{s.t.} \quad & \sum_i (u_i x_k^i) \ge \lambda_k + w_k, \ \forall k \\ & u_i \ge 0, \sum_i u_i = 1 \\ & 0 \le w_k \le \bar{w}, \ \forall k \end{aligned} \tag{19}$$

where $0 < c < 1$ is a small constant. Note that we have added the new variables $w_k \in [0, \bar{w}]$ (where $\bar{w}$ is a constant upper bound), and require $\sum_i u_i x_k^i \geq \lambda_k + w_k$. In the objective function, the term $c \cdot \log(w_k)$ is a penalty function to avoid $w_k$ being too close to 0.

Since $\boldsymbol{\lambda}$ is in the interior of the capacity region, there is a vector $\boldsymbol{\lambda}'$ also in the interior and satisfying $\boldsymbol{\lambda}' > \boldsymbol{\lambda}$ componentwise. Thus, there exist $\mathbf{w}' > 0$ and $\mathbf{u}'$ (such that $\sum_i u_i' x_k^i = \lambda_k' := \lambda_k + w_k'$, $\forall k$) satisfying the constraints. Therefore, in the optimal solution, we have $w_k^* > 0$, $\forall k$ (otherwise, the objective function is $-\infty$, smaller than the objective value that can be achieved by $\mathbf{u}'$ and $\mathbf{w}'$). Thus, $\sum_i u_i^* x_k^i \geq \lambda_k + w_k^* > \lambda_k$. This means that the service rate is strictly larger than the arrival rate, bringing the extra benefit that the queue lengths tend to decrease to 0.

Similar to Section III, we form a partial Lagrangian (with $\mathbf{y} \geq 0$ as dual variables)

$$\begin{aligned}\mathcal{L}(\mathbf{u}, \mathbf{w}; \mathbf{y}) &= -\sum_i u_i \log(u_i) + c \sum_k \log(w_k) \\ &\quad + \sum_k \left[ y_k \left( \sum_i u_i x_k^i - \lambda_k - w_k \right) \right] \\ &= \left[ -\sum_i u_i \log(u_i) + \sum_k \left( y_k \sum_i u_i x_k^i \right) \right] \\ &\quad + \sum_k [c \cdot \log(w_k) - y_k w_k] - \sum_k (y_k \lambda_k). \qquad (20)\end{aligned}$$

Note that the only difference from (12) is the extra term $\sum_k [c \cdot \log(w_k) - y_k w_k]$. Given $\mathbf{y}$, the optimal $\mathbf{w}$ is $w_k = \min\{c/y_k, \bar{w}\}$, $\forall k$, and the optimal $\mathbf{u}$ is the stationary distribution of the CSMA Markov chain with $\mathbf{r} = \mathbf{y}$. Therefore, the subgradient algorithm to update $\mathbf{y}$ is $y_k \leftarrow y_k + \alpha(\lambda_k + w_k - s_k(\mathbf{y}))$.

Since $\mathbf{r} = \mathbf{y}$, we have the following localized algorithm at link $k$ to update $r_k$. Notice its similarity to Algorithm 1.

*Algorithm 3: Enhanced Algorithm 1 to Reduce Queueing Delays:* At time $t_{j+1}$ where $j = 0, 1, 2, \ldots$, let

$$r_k(j+1) = [r_k(j) + \alpha(j) \cdot (\lambda_k'(j) + \min\{c/r_k(j), \bar{w}\} - s_k'(j))]_D \qquad (21)$$

for all $k$, where $\alpha(j)$ is the step size, and $D = [0, r_{\max}]$ where $r_{\max}$ can be $+\infty$. As in Algorithm 1, even when link $k'$ has no backlog, we let it send dummy packets with its current aggressiveness $r_{k'}$. This ensures that the (ideal) average service rate of link $k$ is $s_k(\mathbf{r}(j))$ for all $k$.

Since Algorithm 3 "pretends" to serve some arrival rates higher than the actual arrival rates (due to the positive term $\min\{c/r_k(j), \bar{w}\}$), the queue length $Q_k$ is not only stable, but also tends to be small. The convergence and stability properties of Algorithm 3 when $r_{\max} = \infty$ are discussed in (i) of Appendix B. If $r_{\max} < \infty$, the properties are similar to those in (ii) of Appendix B.

For joint CSMA scheduling and congestion control, a simple way to reduce the delay, similar to [42], is as follows. In item 2 ("congestion control") of Algorithm 2, let the actual flow rate be $\rho \cdot f_m(j)$ where $\rho$ is slightly smaller than 1, and keep other parts of the algorithm unchanged. Then, each link provides a service rate higher than the actual arrival rate. Therefore, the delay is reduced with a small cost in the flow rates.

## VI. Simulations

### A. CSMA Scheduling: i.i.d. Input Traffic With Fixed Average Rates

In our C++ simulations, the transmission time of all links is exponentially distributed with mean 1 ms, and the backoff time of link $k$ is exponentially distributed with mean $1/\exp(r_k)$ ms. The capacity of each link is 1(data unit)/ms. There are six links in "Network 1," whose CG is shown in Fig. 2(a). Define $0 \leq \rho < 1$ as the "load factor," and let $\rho = 0.98$ in this simulation. The arrival rate vector is set to

$$\begin{aligned}\boldsymbol{\lambda} &= \rho * [0.2 * (1,0,1,0,0,0) + 0.3 * (1,0,0,1,0,1) \\ &\quad + 0.2 * (0,1,0,0,1,0) + 0.3 * (0,0,1,0,1,0)] \\ &= \rho * (0.5, 0.2, 0.5, 0.3, 0.5, 0.3)\end{aligned}$$

(data units/ms). We have multiplied by $\rho < 1$ a convex combination of some maximal ISs to ensure that $\boldsymbol{\lambda} \in \mathcal{C}$.

Initially, all queues are empty, and the initial value of $r_k$ is 0 for all $k$. $r_k$ is then adjusted using Algorithm 1 once every $T = 5$ ms (i.e., $T(j) = T$, $\forall j$), with a constant step size $\alpha(j) = \alpha = 0.23$, $\forall j$. Fig. 2(b) shows the evolution of the queue lengths with $r_{\max} = 8$. They are stable despite some oscillations. The vector $\mathbf{r}$ is not shown since in this simulation, it is roughly $\alpha/T$ times the queue lengths. Fig. 2(c) shows the evolution of queue lengths using Algorithm 3 with $c = 0.01$, $\bar{w} = 0.02$ and all other parameters unchanged. The algorithm drives the queue lengths to around zero, thus significantly reducing the queueing delays. Fig. 3 shows the results of Algorithm 3 with $\alpha(j) = 0.46/[(2 + j/1000)\log(2 + j/1000)]$ and $T(j) = (2 + j/1000)$ ms, which satisfy the conditions for convergence in [38]. The constants $c = 0.01$, $\bar{w} = 0.02$, and $r_{\max} = \infty$. To show the negative drift of queues, assume that initially, all queue lengths are 300 data units in Fig. 3. We see that the TA vector $\mathbf{r}$ converges [Fig. 3(a)], and the queues tend to decrease and are stable [Fig. 3(b)]. However, there are more oscillations in the queue lengths than the case with constant step size. This is because when $\alpha(j)$ becomes smaller when $j$ is large, $\mathbf{r}(j)$ becomes less responsive to the variations of queue lengths.

### B. Joint CSMA Scheduling and Congestion Control

In Fig. 4, we simulate a more complex network ("Network 2"). We also go one step further than Network 1 by giving the actual locations of the nodes, not only the CG. Fig. 4(a) shows the network topology, where each circle represents a node. The nodes are arranged in a grid for convenience, and the distance between two adjacent nodes (horizontally or vertically) is 1. Assume that the transmission range is 1, so that a link can only be formed by two adjacent nodes. Assume that two links cannot transmit simultaneously if there are two nodes, one in each link, being within a distance of 1.1 (In IEEE 802.11, for example, DATA and ACK packets are transmitted in opposite directions. This model considers the interference among the two links in

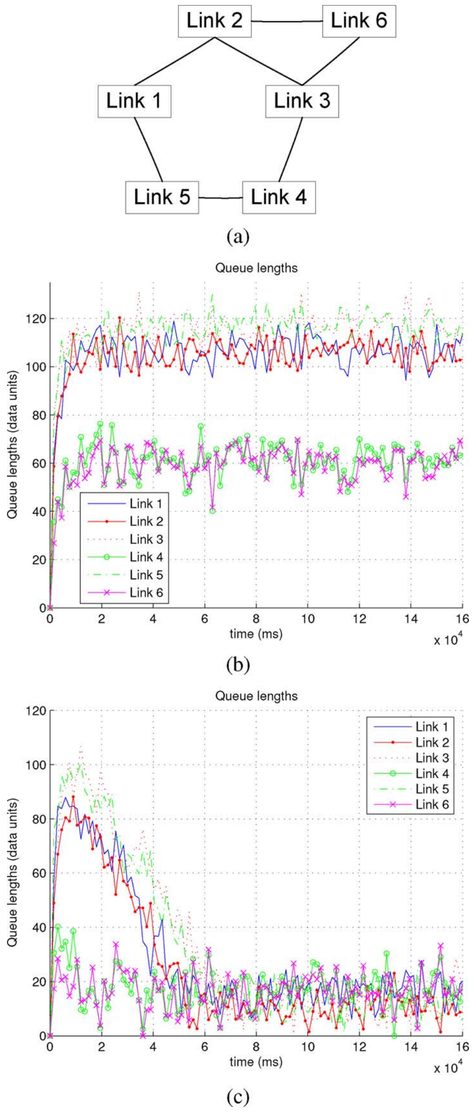


Fig. 2. Adaptive CSMA Scheduling (Network 1). (a) Conflict graph. (b) Queue lengths, with constant step size. The vector $\mathbf{r}$ is not shown since it is proportional to the queue lengths. (c) Queue lengths (with Algorithm 3).

both directions). The paths of three multihop flows are plotted. The utility function of each flow is $v_m(f_m) = \log(f_m + 0.01)$. The weighting factor is $\beta = 3$. (Note that the input rates are adjusted by the congestion control algorithm instead of being specified as in the last subsection.)

Fig. 4(b) shows the evolution of the flow rates, using Algorithm 2 with $T = 5$ ms and $\alpha = 0.23$. We see that they become relatively constant after an initial convergence. By directly solving (17) centrally, we find that the theoretical optimal rates for the three flows are 0.11, 0.134, and 0.134 (data unit/ms), very close to the simulation results. The queues are also stable but not shown here due to the limit on space.

## VII. Implementation Considerations in IEEE 802.11 Networks

### A. *Packet Collisions*

In the idealized CSMA model we used, the sensing time is zero and there is no collision. This allows us to focus on the scheduling problem without worrying about the contention resolution problem. The resulting performance can serve as a benchmark. However, in practice, since the sensing time is not zero, the backoff time is usually chosen to be a multiple of mini-slots, where each mini-slot cannot be arbitrarily small. Therefore, collisions occur given the discrete distribution of backoff times. In this section, we consider this practical issue and discuss alternative algorithms (for 802.11 networks) that are related to the above algorithms with idealized CSMA.

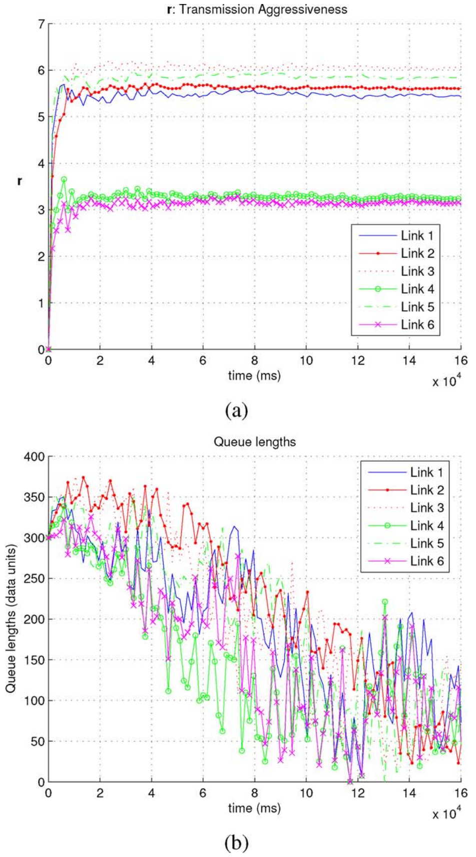


Fig. 3. Decreasing step sizes. (a) The vector $\mathbf{r}$. (b) Queue lengths.

As mentioned earlier, we have recently proposed a model in [30] and [31] that explicitly considered collisions in wireless network without hidden nodes. Moreover, similar algorithms (with probe packets RTS/CTS) have been proposed there to approach the maximal throughput and utility by adjusting the mean transmission times with *fixed* mean backoff times.

In [2] and [21], etc., it was noted that by using small transmission probability in each mini-slot (which increases the backoff times), and correspondingly increasing the transmission times, the collision probability becomes small, in which case the actual CSMA with collisions can be approximated by the idealized CSMA.

In [35], another protocol was proposed to deal with collisions. The protocol has control phases and data phases. Collisions only occur in the control phase, but not in the data phase. The same product-form distribution (1) can be obtained for the data phase, which is then used to achieve the maximal throughput.

In the following, we discuss how to use algorithms in this paper with collisions in mind.

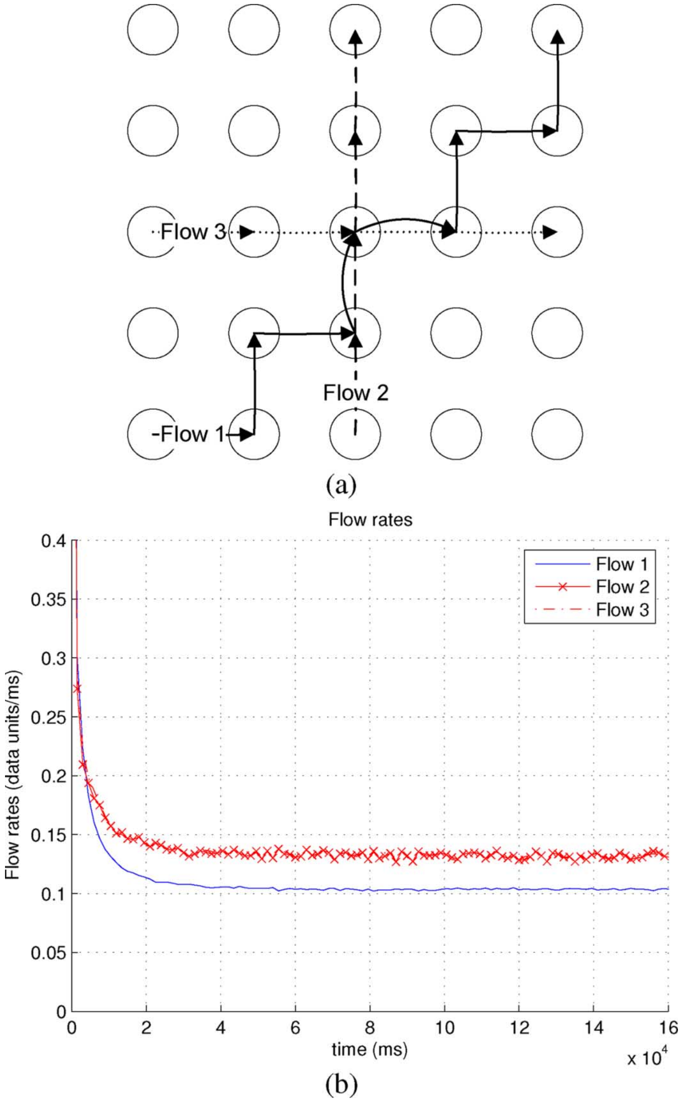


Fig. 4. Flow rates in Network 2 (Grid Topology) with joint scheduling and congestion control. (a) Network 2 and flow directions. (b) Flow rates.

*1) Relationship of TA and the Contention Window in IEEE 802.11:* Assume that for link $k$, the average transmission time is $T_{\rm tr}$. Then, the average backoff time is $T_{\rm tr}/R_k$. Denote by $W_k$ the contention window (CW) in IEEE 802.11 that gives the same average backoff time. (Recall that the distribution of the backoff time is not important as long as it has the correct mean.) Since a random number is uniformly picked from 0 to $W_k - 1$, the average backoff time is $t_m \cdot (W_k - 1)/2$, where $t_m$ is the length of a mini-slot. (For simplicity, we do not consider the Binary Exponential Backoff, or "BEB," in this calculation.) Equating the two quantities gives

$$W_k = \frac{T_{\rm tr}}{R_k}\frac{2}{t_m} + 1. \tag{22}$$

We know that larger CW's lead to lower collision probabilities. By (22), for given $R_k$'s, small mini-slot $t_m$ or large transmission time $T_{\rm tr}$ can lead to large CW. (If $t_m \to 0$ or $T_{\rm tr} \to +\infty$, then collisions can be ignored and we return to the idealized CSMA model.) However, $t_m$ is limited by the sensing time. The mean transmission time $T_{\rm tr}$ can be made large, but should not be too large in practice since that will increase access delays. Therefore, here we impose an upper bound, $r_{\max}$, to all $r_k$'s, where $r_k = \log(R_k)$. This gives $W_k$'s a lower bound: $W_k \ge 2T_{\rm tr}/(\exp(r_{\max}) \cdot t_m) + 1$. For example, assume $T_{\rm tr} = 1$ ms. Recall that a mini-slot in IEEE 802.11a is $t_m := 9$ $\mu$s. If we require that $r_k \le r_{\max} = 2$, then $W_k \ge 31$. These values result in reasonably low collision probabilities if the number of nodes in a collision domain is not too high [36].

Although the upper bound $r_{\max} = 2$ seems small, it can actually achieve a large portion of the capacity region. Consider the simple network in Fig. 1, where the throughput of the two links are $s_1(\mathbf{r}) = R_1/(1+R_1+R_2)$ and $s_2(\mathbf{r}) = R_2/(1+R_1+R_2)$ (for simplicity, here we temporarily assume that collisions are negligible due to the large CWs). The capacity region is $\mathcal{C} = \{(\lambda_1, \lambda_2) | \lambda_1 + \lambda_2 < 1\}$. If $r_1 = r_2 = r_{\max}$, then the total throughput is $2 \cdot \exp(2)/[1 + \exp(2) + \exp(2)] \approx 0.937$, not far from the maximal total throughput 1.

*2) A Condition That Ensures Bounded TA:* In the following, we show that by properly choosing the weighting factor $\beta$ of the total utility in Algorithm 2, it can be guaranteed that every $r_k$ is smaller than $r_{\max}$ at all time if certain conditions are satisfied. (In [37], a similar approach is used to control the amount of backlog in the network.)

*Proposition 5:* Assume that the utility function $v_m(f_m)$ (strictly concave) satisfies $v'_m(0) \le V < \infty$, $\forall m$. Denote by $L$ the largest number of hops of a flow in the network. Then, by setting $\beta = [r_{\max} - (2L-1) \cdot \alpha]/V$ in Algorithm 2, we have $r_k \le r_{\max}$, $\forall k$ at all time.

*Proof:* According to Algorithm 2, the source of flow $m$ solves $f_m(j) = \arg\max_{\hat{f}_m \in [0,1]} \{\beta \cdot v_m(\hat{f}_m) - q_{\delta(m),m}(j) \cdot \hat{f}_m\}$. It is easy to see that if $q_{\delta(m),m}(j) \ge \beta \cdot V$, then $f_m(j) = 0$, i.e., the source stops sending data. Thus $q_{\delta(m),m}(j+1) \le q_{\delta(m),m}(j)$. If $q_{\delta(m),m}(j) < \beta \cdot V$, then $q_{\delta(m),m}(j+1) \le q_{\delta(m),m}(j) + \alpha < \beta \cdot V + \alpha$. Since initially $q_{km}(0) = 0$, $\forall k, m$, by induction, we have

$$q_{\delta(m),m}(j) \le \beta \cdot V + \alpha, \ \forall j, m. \tag{23}$$

Denote $b_{km}(j) := q_{km}(j) - q_{{\rm down}(k,m),m}(j)$. In Algorithm 2, no matter whether flow $m$ has the maximal back-pressure at link $k$, the actual average service rate $s'_{km}(j) = 0$ if $b_{km}(j) \le 0$. That is, $s'_{km}(j) > 0$ only if $b_{km}(j) > 0$. Since $s'_{km}(j) \le 1$, by item 3 of Algorithm 2, $q_{{\rm down}(k,m),m}(j+1) \le q_{{\rm down}(k,m),m}(j) + \alpha$ and $q_{km}(j+1) \ge q_{km}(j) - \alpha$. Then, if $b_{km}(j) > 0$, we have $b_{km}(j+1) \ge b_{km}(j) - 2\alpha > -2\alpha$. If $b_{km}(j) \le 0$, then $b_{km}(j+1) \ge b_{km}(j)$. Since $b_{km}(0) = 0$, by induction, we have

$$b_{km}(j) \ge -2\alpha, \ \forall j, k, m. \tag{24}$$

Since $\sum_{k: a_{mk}=1} b_{km}(j) = q_{\delta(m),m}(j)$, combined with (23) and (24), we have $b_{km}(j) \le \beta \cdot V + \alpha + 2\alpha \cdot (L-1)$. Since $\beta = [r_{\max} - (2L-1) \cdot \alpha]/V$, one has $b_{km}(j) \le r_{\max}$, $\forall j, k, m$. ■

### B. Discrete TA and a Real-World Implementation

Although $r_k$ is continuous in our model, one may find it convenient to quantize $r_k$ into a set of discrete values in a real

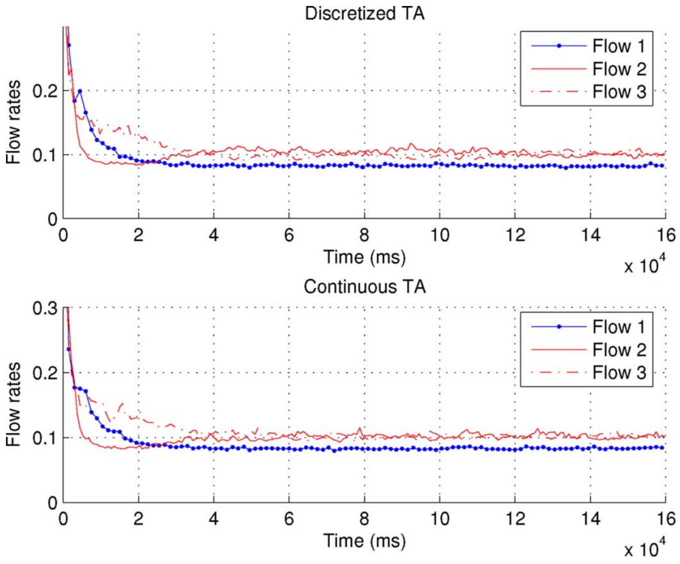


Fig. 5. Flow rates in Network 2 (Grid Topology) with discretized or continuous TA.

implementation. Each discrete value corresponds to a different contention window (a smaller $r_k$ corresponds to a larger CW), and this can be easily mapped to the "service classes" in IEEE 802.11e. Note that here the prioritization is based on the back-pressure instead of service type originally defined in 802.11e. Indeed, in [39], a similar protocol is implemented with 802.11e hardware, and it shows superior performance compared to normal 802.11. (Different from our work, however, [39] only focuses on implementation study. Also, the CW's there are set in a more heuristic way.)

In the following simulation, we set the discrete TA [denoted by $r_{D,k}(j)$] as follows by quantizing the continuous TA, $r_k(j)$, computed by Algorithm 2:

- If $r_k(j) \geq r_{\max}$, then let $r_{D,k}(j) = r_{\max}$. This corresponds to the first class. Then, for $i = 2, 3, \ldots, n_c$, if $r_{\max} - (i-1)G \leq r_k(j) < r_{\max} - (i-2)G$, then let $r_{D,k}(j) = r_{\max} - (i-1)G$, where $G$ is the granularity between adjacent classes, and $n_c$ is the number of classes. In the simulation, we let $r_{\max} = 2$, $n_c = 6$, and $G = \log(2)$ (thus, the CW of class $i+1$ is roughly twice the CW of class $i$).
- Define the minimal TA $r_{\min} := r_{\max} - (n_c - 1)G$. If $r_k(j) < r_{\min}$, then do not transmit packets at all. This is a good approximation since the CW would be quite large with $r_{\min}$ (about 1000). Since transmissions are suspended, the back-pressure tends to increase. The link will resume transmission after $r_k(j)$ goes above $r_{\min}$.

The upper figure in Fig. 5 shows that the resulting flow rates and their fluctuations are similar to those with continuous $\mathbf{r}$ (lower figure). (Collisions and BEB *are* simulated here.) This indicates that the algorithm is robust to the discretization of $\mathbf{r}$. Therefore, using a few prioritized "classes" with different CWs is enough to provide good performance.

## VIII. Conclusion

In this paper, we have proposed a distributed CSMA scheduling algorithm and showed that, under the idealized CSMA, it is throughput-optimal in wireless networks with a general interference model. We have utilized the product-form stationary distribution of CSMA networks in order to obtain the distributed algorithm and the maximal throughput. Furthermore, we have combined that algorithm with congestion control to approach the maximal utility and showed the connection with back-pressure scheduling. The algorithm is easy to implement, and the simulation results are encouraging.

The adaptive CSMA algorithm is a modular MAC-layer component that can work with other algorithms in the transport layer and network layer. In [40], for example, it is combined with optimal routing, anycast, and multicast with network coding.

We also considered some practical issues when implementing the algorithm in an 802.11 setting. Since collisions occur in actual 802.11 networks, we discussed a few recent algorithms that explicitly consider collisions and can still approach throughput-optimality.

Our current performance analysis of Algorithms 1–3 is based on a separation of time scales, i.e., the vector $\mathbf{r}$ is adapted slowly to allow the CSMA Markov chain to closely track the stationary distribution $\mathbf{p}(\mathbf{r})$. The simulations, however, indicate that such slow adaptations are not always necessary. In the future, we are interested to understand more about the case without time-scale separation.

## Appendix A
## Proof the Proposition 2

Consider the convex optimization problem (11), where $\boldsymbol{\lambda}$ is strictly feasible. We now check whether the Slater condition [32, pp. 226–227] is satisfied. Since all the constraints in (11) are linear, we only need to check whether there exists a *feasible* $\mathbf{u}$ that is in the relative interior [32] of the domain $\mathcal{D}$ of the objective function $-\sum_i u_i \log(u_i)$, which is $\mathcal{D} = \{\mathbf{u} | u_i \geq 0\}$. Since $\boldsymbol{\lambda} \in \mathcal{C}$, it can be written as $\boldsymbol{\lambda} = \sum_i \bar{p}_i x^i$ where $\bar{p}_i > 0$, $\forall i$ and $\sum_i \bar{p}_i = 1$. Thus, letting $\mathbf{u} = \bar{\mathbf{p}}$ satisfies the requirement, where $\bar{\mathbf{p}}$ is the vector of $\bar{p}_i$'s. Therefore, the Slater condition is satisfied. As a result, there exist finite dual variables $y_k^* \geq 0, w_i^* \geq 0, z^*$ such that the Lagrangian

$$\begin{aligned}\mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*) \\ = -\sum_i u_i \log(u_i) + \sum_k y_k^* \left( \sum_i u_i \cdot x_k^i - \lambda_k \right) \\ + z^* \left( \sum_i u_i - 1 \right) + \sum_i w_i^* u_i \qquad (25)\end{aligned}$$

is maximized by the optimal solution $\mathbf{u}^*$, and the maximum is attained.

We first claim that the optimal solution satisfies $u_i^* > 0$, $\forall i$. Suppose $u_i^* = 0$ for all $i$'s in a nonempty set $\mathcal{I}$. Since both $\mathbf{u}^*$ and $\bar{\mathbf{p}}$ are feasible for problem (11), any point on the line segment between them is also feasible. Then, if we slightly move $\mathbf{u}$ from $\mathbf{u}^*$ along the direction of $\bar{\mathbf{p}} - \mathbf{u}^*$, the change of the objective function $H(\mathbf{u}) := -\sum_i u_i \log(u_i)$ (at $\mathbf{u}^*$) is proportional to

$$
\begin{aligned}
&(\bar{\mathbf{p}} - \mathbf{u}^*)^T \nabla H(\mathbf{u}^*) \\
&= \sum_i (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1] \\
&= \sum_{i \notin \mathcal{I}} (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1] + \sum_{i \in \mathcal{I}} \bar{p}_i[-\log(u_i^*) - 1].
\end{aligned}
$$

For $i \notin \mathcal{I}$, $u_i^* > 0$, so $\sum_{i \notin \mathcal{I}} (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1]$ is bounded. However, for $i \in \mathcal{I}$, $u_i^* = 0$, so that $-\log(u_i^*) - 1 = +\infty$. Also, since $\bar{p}_i > 0$, we have $(\bar{\mathbf{p}} - \mathbf{u}^*)^T \nabla H(\mathbf{u}^*) = +\infty$. This means that $H(\mathbf{u})$ increases when we slightly move $\mathbf{u}$ away from $\mathbf{u}^*$ toward $\bar{\mathbf{p}}$. Thus, $\mathbf{u}^*$ is not the optimal solution.

Therefore, $u_i^* > 0$, $\forall i$. By complementary slackness, $w_i^* = 0$. Thus, the term $\sum_i w_i^* u_i$ in (25) is 0. Since $\mathbf{u}^*$ maximizes $\mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*)$, it follows that

$$
\frac{\partial \mathcal{L}(\mathbf{u}^*; \mathbf{y}^*, \mathbf{w}^*, z^*)}{\partial u_i} = -\log(u_i^*) - 1 + \sum_k y_k^* x_k^i + z = 0, \ \forall i.
$$

Combining these identities and $\sum_i u_i^* = 1$, we have

$$
u_i^* = \frac{\exp\left(\sum_k y_k^* x_k^i\right)}{\sum_j \exp\left(\sum_k y_k^* x_k^j\right)}, \ \forall i. \tag{26}
$$

Plugging (26) back into (25), we have $\max_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*) = -F(\mathbf{y}^*; \boldsymbol{\lambda})$. Since $\mathbf{u}^*$ and the dual variables $\mathbf{y}^*$ solves (11), $\mathbf{y}^*$ is the solution of $\min_{\mathbf{y} \geq \mathbf{0}} \{-F(\mathbf{y}; \boldsymbol{\lambda})\}$ (and the optimum is attained). Thus, $\sup_{\mathbf{r} \geq 0} F(\mathbf{r}; \boldsymbol{\lambda})$ is attained by $\mathbf{r} = \mathbf{y}^*$. The above proof also shows that (4) is the dual problem of (11).

## Appendix B
## Convergence and Stability Properties of Algorithm (9)

The following are some main results in [38], which includes the detailed proofs.

(i) Let $r_{\max} = +\infty$, i.e., we impose no upper bound on $r_k(j)$. If $\{\alpha(j)\}$ and $\{T(j)\}$ meet certain conditions (which are satisfied by $\alpha(j) = 1/[(j+2)\log(j+2)]$ and $T(j) = j + 2$), and $h(r_k(j)) = \min\{c/r_k(j), \bar{w}\}$ where $c, \bar{w} > 0$ (see Section V for an explanation of the function), then for any strictly feasible $\boldsymbol{\lambda} \in \mathcal{C}$, $\mathbf{r}(j)$ converges with probability 1 to some $\mathbf{r}^{**}$ that satisfies $s_k(\mathbf{r}^{**}) > \lambda_k$, $\forall k$. Also, the queues are "rate-stable" [38]. (With time-varying $\alpha(j), T(j)$, the system is not time-homogeneous, in which case the "positive Harris recurrence" of the queues is not well defined. Therefore, we use the notion of "rate-stable" here. A concern for "rate stability" is that the queue lengths may become large. However, since $s_k(\mathbf{r}^{**}) > \lambda_k$, $\forall k$, it can be shown that the queue lengths return to 0 infinitely often w. p. 1.)

(ii) Let $r_{\max} < +\infty$ and $h(r_k(j)) = \epsilon > 0$. Define the capacity region

$$
\mathcal{C}'(r_{\max}, \epsilon) := \{\boldsymbol{\lambda} | \boldsymbol{\lambda} + \epsilon \cdot \mathbf{1} \in \mathcal{C}, \ \text{and} \ \arg\max_{\mathbf{r} \geq \mathbf{0}} F(\mathbf{r}; \boldsymbol{\lambda} + \epsilon \cdot \mathbf{1}) \in [0, r_{\max}]^K\}.
$$

If $\boldsymbol{\lambda} \in \mathcal{C}'(r_{\max}, \epsilon)$, then there exist constant step size $\alpha(j) = \alpha$ and constant update interval $T(j) = T$ such that all queues are stable. Note that $\mathcal{C}'(r_{\max}, \epsilon) \to \mathcal{C}$ as $r_{\max} \to +\infty$ and $\epsilon \to 0$. Therefore, the maximal throughput can be arbitrarily approximated.

(iii) The case with $h(r_k(j)) = 0$ (i.e., Algorithm 1): Similar to (i), for any $\boldsymbol{\lambda} \in \mathcal{C}$, with $r_{\max} = +\infty$ and $\alpha(j), T(j)$ in (i), $\mathbf{r}(j)$ converges with probability 1 to $\mathbf{r}^*$, the solution of (4), which satisfies $s_k(\mathbf{r}^*) \geq \lambda_k$. $\forall k$, and the queues are rate-stable. Similar to (ii), with $r_{\max} < +\infty$ and assume that $\boldsymbol{\lambda} \in \mathcal{C}'(r_{\max}, 0)$, then one can choose constant $\alpha(j) = \alpha$ and $T(j) = T$ such that the long-term average service rates are arbitrarily close to the arrival rates. As $r_{\max} \to \infty$, $\mathcal{C}'(r_{\max}, 0) \to \mathcal{C}$.

## Acknowledgment

The authors thank the anonymous reviewers for their constructive comments and D. Shah for suggesting using increasing update intervals in one of the convergence proofs.

## References

[1] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[2] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 23–26, 2008, pp. 1511–1519.

[3] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 499–507.

[4] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 27–38, Jun. 2006.

[5] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proc. ACM SIGMETRICS*, Jun. 2007, pp. 313–324.

[6] J. W. Lee, M. Chiang, and R. A. Calderbank, "Utility-optimal random-access control," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2741–2751, Jul. 2007.

[7] P. Gupta and A. L. Stolyar, "Optimal throughput allocation in general random access networks," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2006, pp. 1254–1259.

[8] X. Wu and R. Srikant, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[9] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees in maximal scheduling in wireless networks," in *Proc. 43rd Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2005, pp. 1557–1567.

[10] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, 2006.

[11] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1103–1111.

[12] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1139–1147.

[13] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *Proc. ACM MobiHoc*, May 2009, pp. 165–174.

[14] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proc. IEEE INFOCOM*, Miami, Florida, Mar. 2005, vol. 3, pp. 1804–1814.

[15] P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints," in *Proc. IEEE Conf. Decision Control*, 2007, pp. 1156–1161.

[16] A. Proutiere, Y. Yi, and M. Chiang, "Throughput of random access without message passing," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2008, pp. 509–514.

[17] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2008, pp. 498–502.

[18] Y. Xi and E. M. Yeh, "Throughput optimal distributed control of stochastic wireless networks," in *Proc. WiOpt*, 2006, pp. 1–10.

[19] M. J. Neely, E. Modiano, and C. P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[20] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, 1988.

[21] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Convergence and tradeoff of utility-optimal CSMA," [Online]. Available: http://arxiv.org/abs/0902.1996

[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[23] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

[24] F. P. Kelly*, Reversibility and Stochastic Networks*. New York: Wiley, 1979.

[25] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet radio networks," *IEEE Trans. Commun.*, vol. COMM-35, no. 3, pp. 267–274, Mar. 1987.

[26] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, vol. 1, pp. 23–34.

[27] S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-envelope computation of throughput distributions in CSMA wireless networks," in *Proc. IEEE ICC*, 2009, pp. 1–6.

[28] M. Durvy, O. Dousse, and P. Thiran, "Border effects, fairness, and phase transition in large wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 601–609.

[29] L. Jiang and S. C. Liew, "Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 34–49, Jan. 2008.

[30] L. Jiang and J. Walrand, "Approaching throughput-optimality in a distributed CSMA algorithm: Collisions and stability," in *Proc. ACM Mobihoc S3 Workshop*, May 2009, pp. 5–8.

[31] L. Jiang and J. Walrand, "Approaching throughput-optimality in a distributed CSMA algorithm with contention resolution," UC Berkeley, Tech. Rep., 2009 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-37.html

[32] S. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[33] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, 2008.

[34] P. Whittle*, Systems in Stochastic Equilibrium*. New York: Wiley, 1986.

[35] J. Ni and R. Srikant, "Distributed CSMA/CA algorithms for achieving maximum throughput in wireless networks," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2009, p. 250.

[36] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[37] M. J. Neely and R. Urgaonkar, "Cross layer adaptive control for wireless mesh networks," *Ad Hoc Netw.*, vol. 5, no. 6, pp. 719–743, Aug. 2007.

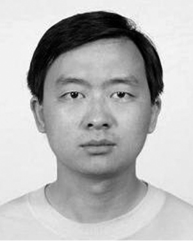
[38] L. Jiang and J. Walrand, "Convergence and stability of a distributed CSMA algorithm for maximal network throughput," UC Berkeley, Tech. Rep., 2009 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-43.html

[39] A. Warrier, S. Ha, P. Wason, and I. Rhee, "DiffQ: Differential backlog congestion control for wireless multi-hop networks," Dept. Computer Science, North Carolina State Univ., Tech. Rep., 2008.

[40] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," UC Berkeley, Tech. Rep., 2009 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-124.html

[41] L. Jiang and J. Walrand, "A distributed algorithm for maximal throughput and optimal fairness in wireless networks with a general interference model," UC Berkeley, EECS Tech. Rep., 2008 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-38.html

[42] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 2936–2940.



**Libin Jiang** received the B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2003, and the M.Phil. degree in information engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2005.

He is currently working toward the Ph.D. degree in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. His research interest includes wireless networks, communications and game theory.

**Jean Walrand** (S'71–M'80–SM'90–F'93) received the Ph.D. degree in electrical engineering and computer science from the University of California (UC), Berkeley

He has been a Professor at UC Berkeley since 1982. He is the author of *An Introduction to Queueing Networks* (Englewood Cliffs, NJ: Prentice Hall, 1988) and *Communication Networks: A First Course* (2nd ed., New York: McGraw-Hill, 1998) and coauthor of *High Performance Communication Networks* (2nd ed., San Mateo, CA: Morgan Kaufman, 2000).

Prof. Walrand is a Fellow of the Belgian American Education Foundation and a recipient of the Lanchester Prize and the Stephen O. Rice Prize.