Convergence and Tradeoff of Utility-Optimal CSMA

Jiaping Liu†, Yung Yi*, Alexandre Proutiere‡, Mung Chiang†, and H. Vincent Poor†

***Abstract*—It has been recently suggested by Jiang and Walrand [25] that adaptive carrier sense multiple access (CSMA) can achieve optimal utility without any message passing in wireless networks. In this paper, a generalization of this algorithm is considered. In the continuous-time model, a proof is presented of the convergence of these adaptive CSMA algorithms to arbitrarily close to utility optimality, without assuming that the network dynamics freezes while the CSMA parameters are updated. In the more realistic, slotted-time model, the impact of collisions on the utility achieved is characterized, and the tradeoff between optimality at equilibrium and short-term fairness is quantified.**

I. Introduction

Design of distributed scheduling algorithms in wireless networks has been extensively studied under various metrics of efficiency and fairness. In their seminal work [1], Tassiulas and Ephremides developed a centralized scheduling algorithm, Max-Weight scheduling, achieving throughput optimality, i.e., stabilizing any arrival for which there exists a stabilizing scheduler. Since then, there has been a large array of lower-complexity, more distributed scheduling algorithms, using the ideas of randomization (pick-and-compare scheduling), weight approximation (maximal/greedy scheduling), or random access with queue-length exchanges, e.g., [2]–[11], to achieve large stability region under unsaturated arrivals of traffic at each node in the network. For saturated arrivals, optimizing a utility function, which captures efficiency and fairness at the equilibrium, has been studied for slotted-Aloha random access, e.g., [12]–[17]. Together with the principle of Layering as Optimization Decomposition, advances in scheduling have also been translated into improvements in joint congestion control, routing, and scheduling over multihop wireless networks, e.g., [18]–[21]. There are many more studied in this topic, as discussed in more detail in surveys such as [22].

A main bottleneck that remains is the need for message passing. Tradeoffs of the time complexity of message passing with throughput and delay have been studied recently in [6], [7], [23], [24]. Message passing reduces "effective" performance, is vulnerable to security attacks, and makes the algorithms not fully distributed. This naturally leads to the following question on simplicity-driven design: *Can random access without message passing approach some type of performance optimality?* The answer was suggested to be positive last year, first in [25] for wireless network, with a similar development in a different context in [26]. Convergence proof and tradeoff were presented in [27], based on which this paper is developed.

In [28]–[30], it has been shown that even *non-adaptive* carrier sense multiple access (CSMA) algorithms, where each link accesses the channel with a fixed probability, are able to provide average throughput close to optimality. Turning to random access with *adaptive* channel access rate, a simulated-annealing based approach was proposed in [31]. A similar idea has been developed recently in [26], [32] for queue stability with unsaturated arrivals: users can adapt their access channel rate depending on their queue size, so that the system dynamics under the random CSMA algorithm solves the Max-Weight problem. As discussed in [29], one issue is that when the buffer of a given user becomes large, its channel access rate should also become large. Consequently, to ensure queue stability and to control the system behavior for arbitrarily large buffers, one needs to design a CSMA protocol with arbitrarily large access rates. This is made possible in [26], [32] by implementing idealized continuous-time CSMA algorithms, where Poisson clocks are used to control the channel accesses, and to ensure zero collisions. By simply limiting the virtual buffer sizes, the problem of large buffers and stability in the implementation of the simulated annealing technique may be avoided[1].

In [25], utility optimality for saturated arrivals (or, rate stability for unsaturated arrivals) is studied, also leveraging the simulated annealing technique. An adaptive CSMA algorithm, without message passing, is developed to maximize utility in the idealized continuous-time model. More recently, [33] proposed an algorithm that is asymptotically optimal in the slotted-time model, by using RTS/CTS-like message passing. More on slotted-time models will be discussed in Section IV.

The contributions of this paper are as follows:

1) We first extend the algorithms in [25], and develop a rigorous proof of the convergence of these algorithms, without assuming that the network dynamics freeze while the CSMA parameters are being updated, for the continuous-time Poisson clock model. New proof techniques are developed to overcome the difficulty of the coupling between the control of CSMA parameters and the queueing network dynamics.
2) We then turn to the more realistic discrete-time contention and backoff model, and quantify the effect of collisions. We reveal and characterize the tradeoff between long-

†: Department of Electrical Engineering, Princeton University, USA. Email: {jiapingl,chiangm,poor}@princeton.edu. ∗: Department of Electrical Engineering, KAIST, South Korea. Email: yiyung@ee.kaist.ac.kr. ‡: Microsoft Research, Cambridge, UK. Email: alexandre.proutiere@microsoft.com.

[1]The algorithm in [32] requires message passing to reach consensus on maximum queue length in the network in order to achieve maximum queue stability.

term efficiency and short-term fairness: short-term fairness decreases *exponentially* as efficiency loss is reduced. Similarly to other distributed scheduling algorithms, there is a 3-dimensional tradeoff [23]: the price of optimality and zero message passing here is delay experienced by some nodes.

The rest of this paper is organized as follows: In Section II, we describe the system model and the Utility-Optimal CSMA (UO-CSMA))algorithms, followed by the formal convergence proof in Section III. In Section IV, we study the impact of collisions in the discrete-time model, and quantify the tradeoff between long-term efficiency and short-term fairness. We conclude with a list of future directions on this topic in Section V.

II. Models and Algorithms

A. Network and interference model

We consider a wireless network composed by a set $\mathcal{L}$ of L links. Interference is modeled by a symmetric, boolean matrix $A \in \{0,1\}^{L\times L}$, where $A_{kl} = 1$ if link k interferes link l, and $A_{kl} = 0$ otherwise. Define by $\mathcal{N} \subset \{0,1\}^L$ the set of the N feasible link activation profiles, or schedules. A schedule $m \in \mathcal{N}$ is a subset of non-interfering active links (i.e., for any $m \in \mathcal{N}$, $k, l \in m$, $A_{kl} = 0$). We assume that the transmitters can transmit at a fixed unit rate when active.

B. Scheduling and utility maximization

The network is assumed to handle single-hop data connections. However, the results presented here can be easily extended to multi-hop connections (e.g., using classical *back-pressure* ideas [1]). The transmitter of each link is saturated, i.e., it always has packets to send. A scheduling algorithm decides at each time which links are activated. Denote by $\boldsymbol{\gamma}^s = (\gamma_l^s, l \in \mathcal{L})$ the long-term throughputs achieved by scheduling algorithm s. The throughput vector of any scheduling algorithm has to belong to the *rate region* Γ defined by

$$\Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}_+^L : \exists \boldsymbol{\pi} \in \mathbb{R}_+^N, \\ \forall l \in \mathcal{L}, \gamma_l \leq \sum_{m\in\mathcal{N}:m_l=1} \pi_m, \quad \sum_{m\in\mathcal{N}} \pi_m = 1\}.$$

In the above, for any schedule $m \in \mathcal{N}$, π_m can be interpreted as the proportion of time schedule m is activated. As is a standard in problems with saturated arrivals, the objective is to design a scheduling algorithm maximizing the total network-wide utility. Specifically, let $U : \mathbb{R}^+ \to \mathbb{R}$ be an increasing, strictly concave, differentiable objective function. We wish to design an algorithm to solve the following optimization problem:

$$\begin{aligned} \max \quad & \Sigma_{l\in\mathcal{L}} U(\gamma_l), \\ \text{s.t.} \quad & \boldsymbol{\gamma} \in \Gamma. \end{aligned} \tag{1}$$

We denote by $\boldsymbol{\gamma}^\star = (\gamma_l^\star, l \in \mathcal{L})$ the optimizer of (1). Most distributed schemes proposed in the literature to date to solve (1) make use of a dual decomposition of the problem into a rate control and a scheduling problem: A virtual queue is associated with each link; a rate control algorithm defines the rate at which packets are sent to the virtual queues, and a scheduling algorithm decides, depending on the level of the virtual queues, which schedule to use with the aim of stabilizing all virtual queues. The main challenge reduces to developing a distributed and efficient scheduling algorithm. Many solutions proposed so far are semi-distributed implementations of the max-weight scheduler introduced in [1], and require information about the queues to be passed around among the nodes or links (e.g., see a large set of references in [22]). This signaling overhead increases communication complexity and reduces effective throughput.

C. Efficiency of CSMA

CSMA-based random access is the most popularly used distributed scheduling algorithms in wireless networks. They are based on random back-off algorithms such as the Decentralized Coordinated Function (DCF) in IEEE802.11. The two basic principles behind CSMA schemes are (i) to detect whether the channel is busy before transmitting, and to refrain from starting a transmission when the channel is sensed busy, and (ii) to wait a random period of time before any transmission to limit the probability of collisions.

The network dynamics under CSMA have been extensively studied in the literature. The following popular model is due to Kelly [34], and has been recently revisited by e.g. [28], [30]. In this model, the transmitter of link l waits an exponentially distributed random period of time with mean $1/\lambda_l$ before transmitting, and when it initiates a transmission, it keeps the channel for an exponentially distributed period of time with mean μ_l. This CSMA algorithm is denoted by CSMA(λ_l, μ_l) in the rest of the paper. Define $\boldsymbol{\lambda} = (\lambda_l, l \in \mathcal{L})$ and $\boldsymbol{\mu} = (\mu_l, l \in \mathcal{L})$. When each link l runs CSMA(λ_l, μ_l), the network dynamics can be captured through a reversible process [35]: If $m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t)$ denotes the active schedule at time t, then $(m^{\boldsymbol{\lambda},\boldsymbol{\mu}}(t), t \geq 0)$ is a continuous-time reversible Markov chain whose stationary distribution $\boldsymbol{\pi}^{\boldsymbol{\lambda},\boldsymbol{\mu}}$ is given by

$$\forall m \in \mathcal{N}, \quad \pi_m^{\boldsymbol{\lambda},\boldsymbol{\mu}} = \frac{\prod_{l:m_l=1} \lambda_l\mu_l}{\sum_{n\in\mathcal{N}} \prod_{l:n_l=1} \lambda_l\mu_l},$$

where by convention $\prod_{l\in\emptyset}(\cdot) = 1$. It is worth noting that due to the reversibility of the process, the above stationary distribution does not depend on the distributions of the back-off durations or of the channel holding times, provided that they are of mean $1/\lambda_l$ and μ_l, respectively, for link l. This insensitivity property allows us to cover a more realistic scenario with uniformly distributed back-off delays and deterministic channel holding times.

Under the above continuous-time model, collisions are mathematically impossible, leading to tractability as a first step of the study. In practice, however, time is slotted and the back-off periods are multiple of slots, which inevitably causes collisions. The impact of collisions is discussed in details in Section IV.

Under the CSMA(λ_l, μ_l)'s algorithms, the link throughputs are given by:

$$\forall l \in \mathcal{L}, \quad \gamma_l^{\boldsymbol{\lambda},\boldsymbol{\mu}} = \sum_{m\in\mathcal{N}:m_l=1} \pi_m^{\boldsymbol{\lambda},\boldsymbol{\mu}}.$$

An important result, proved in [25] (Propositions 1 and 2), states that any throughput vector $\boldsymbol{\gamma} \in \Gamma$ can be *approached* using CSMA($\boldsymbol{\lambda}, \boldsymbol{\mu}$) algorithms. More precisely, we have:

Lemma 1 ([25]): For any $\boldsymbol{\gamma}$ in the interior of Γ, there exist $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}_+^L$ such that

$$\forall l \in \mathcal{L}, \quad \gamma_l \leq \gamma_l^{\boldsymbol{\lambda},\boldsymbol{\mu}}.$$

The above lemma expresses the *optimality* of CSMA scheduling schemes, and it suggests that for approaching the solution of (1), one may use CSMA algorithms.

D. Utility-optimal adaptive CSMA algorithms

We now describe a generic adaptive CSMA-based algorithm to approximately solve (1). The algorithm is an extension of those proposed in [25], and does not require any message passing. Time is divided into *frames* of fixed durations, and the transmitters of each link update their CSMA parameters (i.e., λ_l, μ_l for link l) at the beginning of each frame. To do so, they maintain a virtual queue, denoted by $q_l[t]$ in frame t, for link l. The algorithm operates as follows:

UO-CSMA

1) During frame t, the transmitter of link l runs CSMA($\lambda_l[t], \mu_l[t]$), and records the amount $S_l[t]$ of service received during this frame;
2) At the end of frame t, it updates its virtual queue and its CSMA parameters according to:

$$q_l[t+1] = \left[q_l[t] + \frac{b[t]}{W'(q_l[t])} \big(U'^{-1}(\frac{W(q_l[t])}{V}) - S_l[t] \big) \right]_{q^{\min}}^{q^{\max}},$$

and sets $\lambda_l[t+1]$ and $\mu_l[t+1]$ such that their product is equal to $\exp\{W(q_l[t+1])\}$.

In the above algorithm, $b : \mathbb{N} \to \mathbb{R}$ is a step size function; $W : \mathbb{R}^+ \to \mathbb{R}^+$ is a strictly increasing and continuously differentiable function, termed the *weight function*; V, $q^{\min}$, $q^{\max}(> q^{\min})$ are positive parameters, and $[\cdot]_c^d \equiv \max(d, \min(c, \cdot))$. We will later see that proper choice of b ensures convergence. V controls the accuracy of the algorithm, and the function W controls the transient behavior.

Since the performance of CSMA algorithms depends on the products $\lambda_l \mu_l$ only, we have the choices in UO-CSMA to either update the λ_l's (the transmission intensities) and fix the μ_l's (the transmission durations), or to update the μ_l's and fix the λ_l's, or to update both the λ_l's and μ_l's.

III. Continuous-Time Model: Convergence Proof

UO-CSMA may be interpreted as a stochastic approximation algorithm [36]. The main difficulty in analyzing the convergence of UO-CSMA lies in the fact that the updates in the virtual queues, and hence in the CSMA parameters, depend on the random service processes $(S_l[t], t \geq 0)$. The service processes $(S_l[t], l \in \mathcal{L})$ received by the various links in turn depend on the state of the network at the end of frame $t-1$, and on the updated CSMA parameters $(\boldsymbol{\lambda}[t], \boldsymbol{\mu}[t])$. The convergence proof would have been much simpler if we could assume that the network dynamics freeze in between CSMA parameter updates.

As we will demonstrate, it is possible to represent UO-CSMA as a stochastic approximation algorithm with *controlled Markov noise* (see Borkar [37]). For any vector $\boldsymbol{q} \in \mathbb{N}^L$, we denote by $\pi^{\boldsymbol{q}}$ the distribution on $\mathcal{N}$ resulting from the dynamics of the CSMA(λ_l, μ_l) algorithms, where for all $l \in \mathcal{L}$, $\lambda_l \mu_l = \exp(W(q_l))$. In other words,

$$\forall m \in \mathcal{N}, \quad \pi_m^{\boldsymbol{q}} = \frac{\exp(\sum_{l \in m} W(q_l))}{\sum_{m' \in \mathcal{N}} \exp(\sum_{l \in m'} W(q_l))}. \tag{2}$$

To prove the convergence of UO-CSMA, we will need the following assumption.

Assumption 1: If $\boldsymbol{q}^0 \in \mathbb{R}_+^L$ solves $W(q_l^0) = VU'(\sum_{m: m_l = 1} \pi_m^{\boldsymbol{q}^0})$, then for all $l \in \mathcal{L}$, $q^{\min} \leq q_l^0 \leq q^{\max}$, for all $l \in \mathcal{L}$.

Note that for example, if the utility function U is such that $U'(0) < +\infty$, then Assumption 1 is satisfied when $q^{\min} \leq W^{-1}(VU'(1))$ and $q^{\max} \geq W^{-1}(VU'(0))$. The next theorem states the convergence of UO-CSMA under diminishing step-sizes, towards a point that is arbitrarily close to the utility-optimizer.

Theorem 1: Assume $\sum_{t=0}^{\infty} b[t] = \infty$, and $\sum_{t=0}^{\infty} b[t]^2 < \infty$. Under Assumption 1, for any initial condition $\boldsymbol{q}[0]$, UO-CSMA converges in the following sense:

$$\lim_{t \to \infty} \boldsymbol{q}[t] = \boldsymbol{q}_\star \text{ and } \lim_{t \to \infty} \boldsymbol{\gamma}[t] = \boldsymbol{\gamma}_\star, \text{ almost surely,}$$

where $\gamma_\star$ and $\boldsymbol{q}_\star$ are such that $(\gamma_\star, \boldsymbol{\pi}^{q_\star})$ is the solution of the following convex optimization problem (over $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$):

$$\begin{array}{ll} \max & V \Sigma_{l \in \mathcal{L}} U(\gamma_l) - \Sigma_{m \in \mathcal{N}} \pi_m \log \pi_m \\ \text{s.t.} & \gamma_l \leq \Sigma_{m \in \mathcal{N}: m_l = 1} \pi_m, \quad \Sigma_{m \in \mathcal{N}} \pi_m = 1. \end{array} \tag{3}$$

Furthermore UO-CSMA approximately solves (1) as:

$$\left| \Sigma_{l \in \mathcal{L}} \big(U(\gamma_{\star,l}) - U(\gamma_l^\star) \big) \right| \leq \log |\mathcal{N}| / V. \tag{4}$$

Proof. Recall that $\gamma_l^\star$ is the solution of (1), and $\gamma_{\star,l}$ is the converging point of UO-CSMA. In a first step, we show that in UO-CSMA, the random services $S_l[t]$'s achieved under the CSMA algorithms can be averaged - as if the frame t was long enough so that the $S_l[t]$'s reach their ergodic averages. We also show that the evolutions of the CSMA parameters $\lambda_l[t]$, and $\mu_l[t]$ asymptotically approach to deterministic trajectories (see Lemma 2). In the second step, we prove that the resulting averaged algorithm converges to the solution of (3). The latter step follows the approach used in [25]. The main contribution in our proof is in Step 1 and Lemma 2.

Step 1. From the discrete-time sequence $(\boldsymbol{q}[t], t \geq 0)$, we define a continuous function $\bar{\boldsymbol{q}}(\cdot)$ as follows. Define for all n, $t_n = \sum_{i=1}^n b[i]$, and for all for all $t_n < t \leq t_{n+1}$,

$$\bar{q}_l(t) = q_l[n] + (q_l[n+1] - q_l[n]) \times \Big(\frac{t - t_n}{t_{n+1} - t_n} \Big). \tag{5}$$

Lemma 2 (Convergence and averaging): Fix $\tau > 0$. Denote by $\tilde{q}$ the solution of the following system of ordinary differential equation (o.d.e.'s): for all $l \in \mathcal{L}$,

$$d\tilde{q}_l/dt = \left[U'^{-1}\Big(W_l(\tilde{q}_l)/V\Big) - \sum_{m\in\mathcal{N}:m_l=1} \pi^{\tilde{q}}(m)\right] \times \frac{\mathbf{1}_{\{q^{\min}\le\tilde{q}_l\le q^{\max}\}}}{W'(\tilde{q}_l)}, \quad (6)$$

with $\tilde{\boldsymbol{q}}(\tau) = \bar{\boldsymbol{q}}(\tau)$. Then we have that for all $T > 0$,

$$\lim_{\tau\to\infty} \sup_{t\in[\tau,\tau+T]} \|\bar{\boldsymbol{q}}(t) - \tilde{\boldsymbol{q}}(t)\| = 0 \quad \text{a.s.} \quad (7)$$

Lemma 2 shows that the trajectory of the continuous interpolation $\bar{\boldsymbol{q}}$ of the sequence of the virtual queues $\boldsymbol{q}$ asymptotically approaches that of $\tilde{\boldsymbol{q}}$. Note that in the limiting o.d.e.'s, the service $S_l[t]$ received on each link is averaged with respect to (w.r.t) the stationary distribution $\boldsymbol{\pi}^{\tilde{\boldsymbol{q}}(t)}$ (as if the virtual queues were frozen). Proving this averaging property constitutes the key challenge in analyzing the convergence of UO-CSMA.

Proof of Lemma 2. We attach to each link l a variable $a_l[t]$, where $a_l[t] = 1$ if the link is active at time t (at the end of slot t), and 0 otherwise. Now it can be easily seen that $\boldsymbol{Y}[t] = (\boldsymbol{S}[t], \boldsymbol{a}[t])$ is a non-homogeneous Markov chain whose transition kernel between times t and $t+1$ depends on $\boldsymbol{q}[t]$ only. Now the updates of the virtual queues in UO-CSMA can be written as:

$$q_l[t+1] = q_l[t] + b[t] \times h(q_l[t], Y_l[t]),$$

where

$$h(q, Y) = \frac{1}{W'(q)}(U'^{-1}(W(q)/V) - S)) \cdot \mathbf{1}_{\{q^{\min}\le q_l\le q^{\max}\}},$$

and where $Y = (S, a)$. As a consequence, UO-CSMA can be seen as a stochastic approximation algorithm with controlled Markov noise as defined in [37], [38]. To complete the proof of Lemma 2, we check the sufficient conditions for convergence provided in [38]:

1) The transition kernel of $\boldsymbol{Y}[t]$, parametrized by $\boldsymbol{q}[t]$, is continuous in $\boldsymbol{q}[t]$ (because the transition rates from one state to another are determined by the $\lambda_l[t]$'s and μ_l's, which are continuous in the $q_l[t]$'s). Note also that fixing $\boldsymbol{q}[t] = \boldsymbol{q}_0$ for all time t, the obtained Markov chain $\boldsymbol{Y}[t]$ is ergodic (its state-space is finite and it is irreducible) with stationary distribution $\boldsymbol{\pi}^{\boldsymbol{q}_0}$.

2) h is continuous and Lipschitz in the first argument, uniformly in the second argument. This can be easily checked, given the properties of the utility and weight functions U and W and observing that we restrict our attention to the compact set $[q^{\min}, q^{\max}]$.

3) *Stability* condition: $q_l[t] \le q^{\max}$ for all $l \in \mathcal{L}$ and $t \ge 0$.

4) *Tightness* condition (corresponding to (†) in [38][p. 71]): This is satisfied since $\boldsymbol{Y}[t]$ has a finite state-space (cf. conditions (6.4.1) and (6.4.2) in [38][pp.76]). □

In view of Lemma 2, if there exists an equilibrium $\boldsymbol{q}_\star$ such that $\lim_{t\to\infty} \tilde{\boldsymbol{q}}(t) = \boldsymbol{q}_\star$, then we would also have $\lim_{t\to\infty} \boldsymbol{q}[t] = \boldsymbol{q}_\star$ a.s. (this can be shown as in [39]).

Step 2. To complete the convergence proof, we show as in [25] that (6) may be interpreted as a sub-gradient algorithm (projected on a bounded interval) solving the Lagrange dual problem of (3). Let $D(\boldsymbol{\nu}, \eta)$ denote the dual function of (3). Then we show that (6) is the sub-gradient algorithm of:

$$\min\ D(\boldsymbol{\nu}, \eta),\ \text{s.t. } \nu^{\min} \le \nu_l \le \nu^{\max},\ \forall l \in \mathcal{L}. \quad (8)$$

Here we include the upper-bound $\nu^{\max}$ (resp. lower-bound $\nu^{\min}$) that corresponds to the limitation of the q_l's: $\nu^{\max} = W(q^{\max})$ (resp. $\nu^{\min} = W(q^{\min})$). The Lagrangian of (3) is given by

$$\begin{aligned} L(\boldsymbol{\gamma}, \boldsymbol{\pi}; \boldsymbol{\nu}, \eta) =& \big(\Sigma_{l\in\mathcal{L}} V \log\gamma_l - \nu_l\gamma_l\big) \\ &+ \big(\Sigma_{l\in\mathcal{L}}\nu_l\Sigma_{m\in\mathcal{N}:m_l=1}\pi_m \\ &- \Sigma_{m\in\mathcal{N}}\pi_m\log\pi_m\big) + \eta\big(\Sigma_{m\in\mathcal{N}}\pi_m - 1\big). \end{aligned}$$

Then the Karush-Kuhn-Tucker (KKT) conditions of (3) are given by

$$VU'(\gamma_l) = \nu_l,\ \forall l \in \mathcal{L}, \quad (9)$$

$$-1 - \log\pi_m + \Sigma_{l:m_l=1}\nu_l + \eta = 0,\ \forall m \in \mathcal{N}, \quad (10)$$

$$\nu_l \times \big(\gamma_l - \Sigma_{m\in\mathcal{N}:m_l=1}\pi_m\big) = 0, \quad (11)$$

$$\eta \times \Big(\sum_{m\in\mathcal{N}} \pi_m - 1\Big), \quad (12)$$

$$\forall l \in \mathcal{L}, \nu_l \ge 0. \quad (13)$$

The sub-gradient of (9) (when accounting for (11)) is:

$$d\nu_l/dt = \Big(U'^{-1}\Big(\nu_l/V\Big) - \Sigma_{\substack{m\in\mathcal{M}\\:m_l=1}} \pi^{\tilde{\boldsymbol{q}}}_m\Big) \cdot \mathbf{1}_{\{\nu^{\min}\le\nu_l\le\nu^{\max}\}}. \quad (14)$$

Eq. (14) is equivalent to (6) when $\nu_l = W(\tilde{q}_l)$ for all $l \in \mathcal{L}$, provided that the solution $\boldsymbol{\nu}_\star = (\nu_{\star,l}, l \in \mathcal{L})$ of (8) without the constraints $\nu^{\min} \le \nu \le \nu^{\max}$ actually belongs to the interval $[\nu^{\min}, \nu^{\max}]$. The latter condition is satisfied in view of Assumption 1. Since (8) is a strictly convex optimization problem, (14) converges to its unique equilibrium $\boldsymbol{\nu}_\star$, and hence (6) converges to $q_\star$ such that for all $l \in \mathcal{L}$, $W(q_{\star,l}) = \nu_{\star,l}$. Finally, (10) and (12) are solved for $\boldsymbol{\pi} = \boldsymbol{\pi}^{\boldsymbol{q}_\star}$ and

$$\eta = 1 - \log\left(\sum_{m\in\mathcal{N}} \exp\Big(\sum_{l\in\mathcal{L}:m_l=1} W(q_{\star,l})\Big)\right).$$

To prove the inequality (4), we just remark that (1) is equivalent to the following optimization problem:

$$\begin{aligned} \max\quad & V\Sigma_{l\in\mathcal{L}}U(\gamma_l) \\ \text{s.t.}\quad & \gamma_l \le \Sigma_{m\in\mathcal{N}:m_l=1}\pi_m, \quad \Sigma_{m\in\mathcal{N}}\pi_m = 1. \end{aligned} \quad (15)$$

Eq. (4) is obtained by comparing (3) and (15), and using the fact that the entropy $\sum_m \pi_m \log\pi_m$ is always bounded by $\log|\mathcal{N}|$. The proof of Theorem 1 is complete. □

Under the assumption of Theorem 1, the CSMA parameters of the various transmitters $((\lambda_l[t], \mu_l[t]), l \in \mathcal{L})$ are such that their products $(\lambda_l[t]\mu_l[t], l \in \mathcal{L})$ converge to $(\rho_{\star,l} = \exp(W(q_{\star,l})), l \in \mathcal{L})$ almost surely when $t \to \infty$, and the limiting products are characterized by the following set of equations: For all $l \in \mathcal{L}$,

$$U'^{-1}\left(\frac{\log(\rho_{\star,l})}{V}\right) = \frac{\sum_{m:m_l=1}\prod_{j\in m}(\rho_{\star,j})}{\sum_{m\in\mathcal{N}}\prod_{j\in m}(\rho_{\star,j})} \quad (= \gamma_{\star,l}).$$

From these equations, we deduce that increasing V tends to increase the $\rho_{\star,l}$'s which in turn improves the efficiency of the algorithm. The downside of large V is slower convergence.

IV. Slotted-Time Model: Collision and Tradeoff

In the previous section, we have analyzed the convergence of UO-CSMA in the ideal continuous-time setting where collisions are made mathematically impossible. In practical implementaions, however, time is slotted and collisions may occur. We consider the following model for the slotted CSMA: The transmitter of link l starts a transmission at the end of a slot with probability p_l if the slot has been sensed idle. When a link is active, it can experience either a successful transmission or a collision. When a link is currently successfully transmitting, it releases the channel with probability $1/\mu_l$ at the end of a slot. In the case of a collision, interfering links involved in the collision all stop to transmit simultaneously.

We consider two types of collisions:

(a) *Short collisions.* The links involved in a collision all release the channel with probability $1/\mu$ at the end of a slot. Short collisions may represent RTS/CTS-like procedures: before transmitting links probe the channel with a small signaling message.

(b) *Long collisions.* The collision duration is equal to the maximum transmission durations of links involved in the collisions. To model long collisions, we assume that the links involved in a collision all release the channel with probability $1/\mu_c$ at the end of a slot, where c denotes the set of links experiencing the collision, and $\mu_c = \max_{l\in c} \mu_l$. Long collisions occur when RTS/CTS-like procedures are not implemented.

In the following, we denote by s-CSMA(p_l, μ_l, μ) and s-CSMA(p_l, μ_l) the above slotted CSMA algorithm with short and long collisions, respectively.

A. Impact of collisions on efficiency

We now investigate the impact of collisions on the performance of CSMA algorithms. We consider long collisions only. The case of short collisions can be analyzed similarly. Assume that the transmitter of link l implements the s-CSMA(p_l, μ_l) algorithm. Define by $m[t]$ the resulting schedule used in slot t. Note that $m[t]$ may take any value in $\mathcal{M} = \{0,1\}^L$ due to the possibility of collisions (if $m_l[t] = 1 = m_k[t]$ and $A_{kl} = 1$, then links k and l experience a collision during slot t).

We introduce more notation: for any schedules $m, m' \in \mathcal{M}$, let $s(m)$ denote the set of links successfully transmitting in schedule m; let $s(m, m')$ be the set of links successfully transmitting in both m and m'; $s(m \setminus m')$ is the set of links successfully transmitting in m but not in m'; let $c(m)$ be the set of collisions in m (note that each $c \in c(m)$ is a set of links, and by convention, we write $l \in c(m)$ if $\exists c \in c(m) : l \in c$); let $c(m, m')$ be the set of collisions in both m and m'; let $c(m \setminus m')$ be the set of collisions in m but not in m'; finally, let $n(m)$ be the links that has a neighbor transmitting in m, i.e., $l \in n(m)$ if $\exists k \in s(m) \cup c(m) : A_{kl} = 1$.

Now $(m[t], t \in \mathbb{N})$ is a discrete Markov chain whose transition kernel $(\beta_{m,m'}, m, m' \in \mathcal{M})$ is given by

$$\begin{aligned}\beta_{m,m'} = &\prod_{l\in s(m,m')} (1 - \frac{1}{\mu_l}) \prod_{l\in s(m\setminus m')} \frac{1}{\mu_l} \prod_{c\in c(m,m')} (1 - \frac{1}{\mu_c}) \\ &\times \prod_{c\in c(m\setminus m')} \frac{1}{\mu_c} \prod_{l\in s(m'\setminus m)\cup c(m'\setminus m)} p_l \\ &\times \prod_{l\notin w(m)} (1 - p_l),\end{aligned}$$

where $w(m) = m \cup n(m) \cup s(m' \setminus m) \cup c(m' \setminus m)$. Then one can easily verify that the Markov chain $(m[t], t \in \mathbb{N})$ is reversible as stated in the following lemma.

Lemma 3: Let $\mathbf{0}$ be the state such that no link is active. Denote by $\pi^{\boldsymbol{p},\boldsymbol{\mu},s}$ the distribution such that there exists G (a normalizing constant) such that, for all $m \in \{0,1\}^L \setminus \{\mathbf{0}\}$,

$$\pi^{\boldsymbol{p},\boldsymbol{\mu},s}_m = G^{-1} \prod_{l\in s(m)} \frac{p_l\mu_l}{1-p_l} \prod_{c\in c(m)} \left[\mu_c \prod_{l\in c} \frac{p_l}{1-p_l}\right],$$

and

$$\pi^{\boldsymbol{p},\boldsymbol{\mu},s}_{\mathbf{0}} = \frac{1}{\prod_{l\in\mathcal{L}}(1-p_l)}.$$

Then the following local balance equations are satisfied:

$$\forall m, m', \quad \beta_{m,m'}\pi^{\boldsymbol{p},\boldsymbol{\mu},s}_m = \beta_{m',m}\pi^{\boldsymbol{p},\boldsymbol{\mu},s}_{m'}.$$

Hence, $\pi^{\boldsymbol{p},\boldsymbol{\mu},s}$ is the stationary distribution of the Markov chain $(m[t], t \in \mathbb{N})$.

Note that the superscript s in $\pi^{\boldsymbol{p},\boldsymbol{\mu},s}$ indicates tht time is slotted. Under the s-CSMA(λ_l, μ_l)'s algorithms, the link throughputs are given by:

$$\forall l \in \mathcal{L}, \quad \gamma^{\boldsymbol{p},\boldsymbol{\mu},s}_l = \sum_{m\in\mathcal{N}:m_l=1} \pi^{\boldsymbol{p},\boldsymbol{\mu},s}_m. \tag{16}$$

From the above analysis, if we wish to get throughputs under slotted CSMA algorithms very close to those obtained under the continuous-time CSMA algorithms, we need either (i) to keep the collision duration μ negligible compared to the channel holding times μ_l's (for the case with short collisions), or (ii) or to keep the transmission probabilities p_l's close to 0, and to have very large channel holding times. The condition (i) could be ensured using RTS/CTS-like procedures and having very large channel holding times. The condition (ii) would be met if for all $l \in \mathcal{L}$, $p_l = \delta\alpha_l$, $\mu_l = \xi_l/\delta$ with $\delta << 1$. In such case, in view of Lemma 3 and (16), we have

$$\forall l \in \mathcal{L}, \gamma^{\boldsymbol{p},\boldsymbol{\mu},s}_l = \gamma^{\boldsymbol{\alpha},\boldsymbol{\xi}}_l - C_l\delta + o(\delta),$$

where for all $l \in \mathcal{L}$, $C_l > 0$ is a constant depending on the α_l's and ξ_l's, and on the network topology.

To adapt UO-CSMA to the practical scenario where time is slotted, condition (i) is not sufficient. Indeed, the efficiency of UO-CSMA in the continuous-time setting relies on the fact that at any time t, the probability that a link, say l, becomes active should be proportional to $\lambda_l[t]$ if its neighbors are idle. If we impose (i) only, the probability at which link l becomes active is not proportional to p_l, but depends in a complicated manner on the transmission probabilities of its neighbors. In such cases, there is no clear mapping between the p_l's (in

the slotted-time system) and the λ_l's (in the continuous-time system).

When imposing condition (ii), this mapping is clear. We can adapt UO-CSMA to the slotted-time setting by choosing a very small parameter δ, by letting the transmission probabilities and channel holding times be equal to $\delta\alpha_l[t]$ and $\xi_l[t]/\delta$ at time t for link l, and by updating the parameters $(\alpha_l[t], \xi_l[t])$'s as in UO-CSMA (where $\lambda_l[t] = \alpha_l[t]$ and $\mu_l[t] = \xi_l[t]$). Since we want to keep the collision rates at a very low level, we need to keep the transmission probabilities very small, which in turn means that in the updates of the $\alpha_l[t]$'s and the $\xi_l[t]$'s in UO-CSMA should be such that the $\alpha_l[t]$'s remain bounded - this is possible since in UO-CSMA what matters are the products $\alpha_l[t]\xi_l[t]$'s only, not their individual values. With this modification of UO-CSMA, we ensure that for all $l \in \mathcal{L}$, the long-term link throughput of link l is $\gamma_{\star,l} - C_l\delta + o(\delta)$.

B. Short-term fairness vs. Long-term efficiency

As we discussed at the end of Section III, if we want the resulting link throughputs of UO-CSMA to be close to the solution of (1), the products of the transmission probabilities and of the channel holding times need to be very large. In the adaptation of UO-CSMA to the slotted-time scenario, this implies that the channel holding times are very large, since the transmission probabilities must remain very small (to ensure very low collision rates). This further implies that the delay between two successive successful transmissions on a link is very large as well. In other words, to ensure efficiency, we need to sacrifice *short-term fairness*.

Another source of short-term unfairness with UO-CSMA is the fact that if a link is interfered with by a lot of links (compared to other links), before transmitting it needs to wait until all its neighbors become inactive. This waiting time can be very long, especially if these neighbors do not sense each other. When the link finally gets access to the channel, it then needs to hold the channel for a duration that is much larger than the transmission durations of its neighbors, in order to achieve throughput fairness. This may considerably exacerbate short-term unfairness.

We now quantify the two above observations. We first define the short-term fairness index of link l as $1/T_l$ where T_l is the average delay between two successive successful transmissions on this link. This is in contrast to the standard notion of *long-term fairness*, which is often captured by the α-fair utility function and refers to fairness at equlibrium. For illustrative purposes, we consider a simple star network: it is composed of $L+1$ links, where link 1 is interfered with by all other links ($A_{1k} = 1$ for all $k > 1$) and where link k, $k > 1$, is interfered with only by link 1 ($A_{kl} = 0$ for all $k, l > 1$). At time t, the transmission probability for link l is $\delta \times \alpha_l[t]$ and the channel holding time is $\xi_l[t]/\delta$. We consider long collisions whose durations are equal to the maximum duration of the channel holding times of the links involved in the collision. For this network, the solution of (1) is $\gamma_1^\star = 1/(L+1)$ and $\gamma_l^\star = L/(L+1)$ for all $l > 1$. Now we run UO-CSMA to update the parameters $(\alpha_l[t], \xi_l[t])$. As mentioned above, the parameters $\alpha_l[t]$ need to be bounded. Without loss of generality, we assume here that they are constant and equal to 1, and hence UO-CSMA consists of updating the parameters $\xi_l[t]$'s. Assume that we wish to guarantee that after convergence, the throughput of link l is at least $\gamma_l^\star \times (1-\epsilon)$. From the analysis in the previous subsection, we know that by scaling δ as ϵ, the throughput of link l is equal to $\gamma_{\star,l} \times (1-\epsilon/2+o(\epsilon))$. Let $\xi_{\star,l}$ be the value of $\xi_l[t]$ after convergence of UO-CSMA. Note that, for all $l > 1$, by symmetry, $\xi_{\star,l} = \xi_\star$. Now we have

$$\gamma_{\star,1} = \frac{\xi_{\star,1}}{(1+\xi_\star)^L + \xi_{\star,1}},$$

and for all $l > 1$,

$$L\gamma_{\star,l} = \frac{(1+\xi_\star)^L - 1}{(1+\xi_\star)^L + \xi_{\star,1}}.$$

Achieving for all links l $\gamma_{\star,l} \geq \gamma_l^\star(1-\epsilon/2)$ requires first that $\xi_{\star,1} \approx \xi_\star^L/L$ and then that $\xi_\star$ scales as $1/\epsilon$. Finally, the channel holding time for channel $l > 1$ has to scale as $1/\epsilon^2$ whereas that for link 1 has to scale as $1/\epsilon^{2L}$. Using classical results in return times of Markov chains [40], we now have that for all links l the short-term fairness index $1/T_l$ scales as ϵ^{2L}. This quantifies the trade-off between efficiency and short-term fairness when implementing UO-CSMA in slotted-time systems. Achieving high efficiency requires to deteriorate short-term fairness considerably: in the above star network, to ensure that the throughputs are at a distance at most ϵ from the utility-optimal throughputs, the short-term fairness index has to scale as ϵ^{2L}.

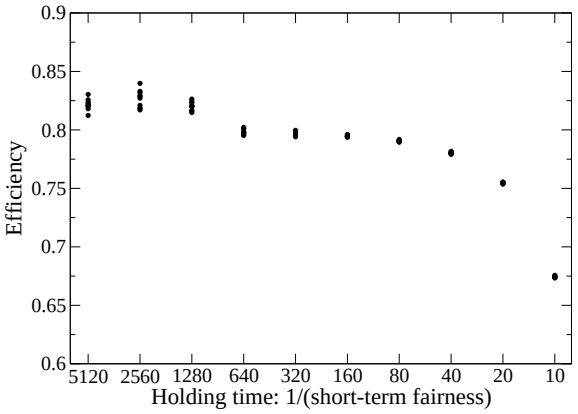

Fig. 1. Efficiency vs. short-term fairness tradeoff in a 3-link linear network. Algorithm parameters: $b[t] = 0.001$, $W(x) = x$, $V = 1$, $\epsilon\alpha^{\max} = 0.1$.

We illustrate this tradeoff numerically using a simple 3-link linear network, where links 1 and 2 (resp. 3 and 2) are interefering with each other, but links 1 and 3 are interference-free. Figure 1 shows the efficiency (i.e., $1-\epsilon$) as a function of 1/(short-term fairness index). 10 experiments were carried out with different random seeds for each value on the x-axis. In UO-CSMA, to limit collisions, we maintain all transmission probabilities at a level less than 0.1, i.e., $\epsilon \times \alpha^{\max} = 0.1$. Note that we can achieve a quite good efficiency for random access without message passing, e.g. 85%. Pushing this efficiency further up requires longer holding times.

V. Conclusion and Future Work

Achieving optimality in terms of throughput and fairness has been known to require scheduling algorithms with message passing. Recent works suggest adaptive CSMA without message passing can achieve utility-optimality arbitrarily closely. In this paper, we have confirmed, through a proof that does not rely on freezing of network dynamics in between parameter updates, that indeed this is true for the idealized, continuous-time model, but there is an exponentially large price to pay in terms of short-term fairness in the realistic, slotted-time model. The algorithm development ideas and convergence proof techniques have been based on a combination of the powerful techniques of loss network modeling and simulated annealing for distributed scheduling from the 1980s.

In addition to extending to multihop cross-layer models, there are also more challenging next steps, especially the characterization and design of transient behavior, including short-term fairness and delay, through the algorithm parameters like V and the function W. Achieving queue stability without message passing also remains open. Perhaps most importantly, given that "simplicity" is the main attractiveness of this class of adaptive CSMA algorithms, implementing and deploying the proposed algorithms in an operational network will help bridge the gap between theory and practice in wireless scheduling.

ACKNOWLEDGMENTS

We thank helpful discussions with L. Jiang, R. Srikant, D. Shah, and J. Walrand. This work has been supported in part by IT R&D program of MKE/IITA [2009-F-045-01], Korea Research Council of Fundamental Science and Technology, and US PECASE and ONR YIP awards.

References

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

[2] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radionetworks and input queued switches," in *Proceedings of Infocom*, San Francisco, CA, 1998.

[3] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *43rd Annual Conference on Communication, Control and Computing*, Monticello, IL, 2005.

[4] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of ACM Sigmetrics*, Saint Malo, France, 2006.

[5] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *Proceedings of Infocom*, Anchorage, AK, 2007.

[6] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proceedings of ACM Sigmetrics*, San Diego, CA, 2007.

[7] S. Ray and S. Sarkar, "Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning," in *Proceedings of Information Theory and Applications Second Workshop*, La Jolla, CA, 2007.

[8] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Proceedings of Infocom*, Anchorage, AK, 2007.

[9] Y. Yi and M. Chiang, "Wireless scheduling with O(1) complexity for m-hop interference model," in *Proceedings of IEEE International Conference on Communications*, Beijing, China, 2008.

[10] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," in *Proceedings of IEEE Infocom*, Anchorage, AK, 2007.

[11] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, 2006.

[12] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in aloha networks," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1858–1862, 2004.

[13] J. W. Lee, M. Chiang, and R. A. Calderbank, "Utility-optimal medium access control: reverse and forward engineering," in *Proceedings of IEEE Infocom*, Barcelona, Spain, 2006.

[14] X. Wang and K. Kar, "Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access," in *Proceedings of ACM Mobihoc*, Urbana-Champaign, IL, 2005.

[15] A. H. Mohsenian-Rad, J. Huang, M. Chiang, and V. W. S. Wong, "Utility-optimal random access: Optimal performance without frequent explicit message passing," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 898–911, 2009.

[16] ——, "Utility-optimal random access: Reduced complexity, fast convergence, and robust performance," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 898–911, 2009.

[17] J. Liu, A. Stolyar, M. Chiang, and H. V. Poor, "Queue backpressure random access in multihop wireless networks: Optimality and stability," *IEEE Transactions on Information Theory*, 2009, to appear.

[18] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proceedings of IEEE Infocom*, Miami, FL, 2005.

[19] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of IEEE Infocom*, Miami, FL, 2005.

[20] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Joint optimal congestion control, routing, and scheduling in wireless ad hoc networks," in *Proceeding of IEEE Infocom*, Barcelona, Spain, 2006.

[21] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE Journal on Selected Areas of Communication (JSAC), Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1514–1524, 2006.

[22] Y. Yi and M. Chiang, "Stochastic network utility maximization and wireless scheduling," 2009, to be published as a book chapter of *Next-Generation Internet Architectures and Protocols*, Cambridge University Press. Also, available at http://lanada.kaist.ac.kr/pubs/scheduling.pdf.

[23] Y. Yi, A. Proutiere, and M. Chiang, "Complexity in wireless scheduling: Impact and tradeoffs," in *Proceedings of ACM Mobihoc*, Hong Kong, China, 2008.

[24] Y. Yi, J. Zhang, and M. Chiang, "Delay and effective throughput of wireless scheduling in heavy traffic regimes: Vacation model for complexity," in *Proceedings of ACM Mobihoc*, New Orleans, LA, 2009.

[25] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proceedings of 46th Annual Conference on Communication, Control and Computing*, Monticello, IL, Sep. 2008.

[26] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in *Proceedings of Conference on Information Sciences and Systems*, Princeton, NJ, 2008.

[27] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Random access without message passing: Throughput, fairness and tradeoffs," *MSR Technical Report*, Sep. 2008.

[28] M. Durvy and P. Thiran, "Packing approach to compare slotted and non-slotted medium access control," in *Proceedings of IEEE Infocom*, Barcelona, Spain, 2006.

[29] A. Proutiere, Y. Yi, and M. Chiang, "Throughput of random access without message passing," in *Proceedings of Conference on Information Sciences and Systems*, Princeton, NJ, 2008.

[30] C. Bordenave, D. McDonald, and A. Proutiere, "Performance of random medium access control: An asymptotic approach," in *Proceedings of ACM Sigmetrics*, Annapolis, MD, 2008.

[31] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.

[32] J. Shin, D. Shah, and S. Rajagopalan, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proceedings of ACM Sigmetrics*, Seattle, WA, 2009.

[33] J. Ni and R. Srikant, "Distributed csma/ca algorithms for achieving maximum throughput in wireless networks," in *Proceedings of nformation Theory and Applications Workshop*, La Jolla, CA2009.
[34] F. Kelly, "Stochastic models of computer communication systems," *Journal of the Royal Statistical Society*, vol. 47, no. 3, pp. 379–395, 1985.
[35] ——, *Reversibility and Stochastic Networks*. Wiley, Chichester, 1979.
[36] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
[37] V. Borkar, "Stochastic approximation with controlled markov noise," *Systems and Control Letters*, vol. 55, pp. 139–145, 2006.
[38] ——, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency (Cambridge University Press), 2008.
[39] M. Benaim, "A dynamical system approach to stochastic approximation," *SIAM J. on Control and Optimization*, vol. 34, pp. 437–472, 1996.
[40] P. Bremaud, *Markov Chaing: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York, 1999.