WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* **00**: 1–12 (2009)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.0000

# Towards Utility-optimal Random Access Without Message Passing

J. Liu[1], Y. Yi[2], A. Proutiere[3], M. Chiang[1]*, H. V. Poor[1]

[1] *Electrical Engineering Dept., Princeton University, NJ, USA*
[2] *Electrical Engineering Dept., Korea Advanced Institute of Science and Technology, Daejon, Republic of Korea*
[3] *Microsoft Research, Cambridge, UK*

## Summary

It has been recently suggested by Jiang and Walrand that adaptive carrier sense multiple access (CSMA) can achieve optimal utility without any message passing in wireless networks. In this paper, after a survey of recent work on random access, 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 be arbitrarily close to utility optimality, without assuming that the network dynamics converge to an equilibrium in between consecutive CSMA parameter updates. In the more realistic, slotted-time model, the impact of collisions on the utility achieved is characterized, and the tradeoff between optimality and short-term fairness is quantified. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: CSMA, Random Access, Scheduling, Wireless Network, Distributed control, Optimization, Stochastic Network, Network Utility Maximization

## 1. Introduction

The 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, a large array of lower-complexity, more distributed scheduling algorithms has been developed, using the ideas of randomization (pick-and-compare scheduling), weight approximation (maximal/greedy scheduling), or random access with queue-length exchanges, e.g., in [2, 3, 4, 5, 6, 7, 8, 9, 10, 11], to achieve large stability region under unsaturated arrivals of traffic at each node in the network. For saturated arrivals, maximizing a utility function, which captures efficiency and fairness at the equilibrium, has been studied for slotted-Aloha random access, e.g., [12, 13, 14, 15, 16, 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, 19, 20, 21, 22]. There are many more studies on this topic, as discussed in surveys such as [23].

A main bottleneck remaining in all the afore-mentioned studies is the need for message passing. Tradeoffs of the time complexity of message passing with throughput and delay have been studied recently in [7, 6, 24, 25]. Message passing reduces the "effective" performance, is vulnerable to security

*Correspondence to: yiyung@ee.kaist.ac.kr


*Prepared using* wcmauth.cls *[Version: 2007/01/09 v1.00]*

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 [26] for wireless network, with a similar development in a different context in [27]. A convergence proof for algorithms and an analysis of the achieved tradeoff between efficiency and short-term fairness have been presented in [28], based on which this paper is developed.

In this paper, we first provide a review of the literature related to random-access based scheduling schemes for wireless networks. We then extend the algorithms in [26], and develop a rigorous proof of the convergence of these algorithms. The proof does not assume that the network dynamics converge in between updates of the carrier-sense-multiple-access (CSMA) parameters, and holds for the continuous-time Poisson clock model. Finally we turn to more realistic discrete-time contention and backoff models, and quantify the effect of collisions. We reveal and characterize the tradeoff between long-term efficiency and short-term fairness: short-term fairness decreases significantly as efficiency loss is reduced. Similarly to other distributed scheduling algorithms, there is a 3-dimensional tradeoff [24]: the price of optimality without message passing is delay experienced by some nodes.

The rest of this paper is organized as follows: In Section 2 we review the literature related to random-access based scheduling algorithms in wireless networks. In Section 3, we describe the system model and the Utility-Optimal CSMA (UO-CSMA) algorithms, and prove the convergence of this class of algorithms. In Section 3.5, we further 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 discussion of future directions on this topic in Section 4.

## 2. Random-access Based Scheduling

### 2.1. Slotted Aloha

The simplest random access medium access control (MAC) protocol for wireless systems is slotted Aloha, in which time is slotted and at the beginning of each slot, each link accesses the channel with a given probability, should the corresponding transmitter have some packets to send.

Assume that we have $L$ links, and, initially, that the transmission probabilities $\boldsymbol{p} = (p_1, \dots, p_L)$ on the links are fixed. If links are always backlogged, then the long-term throughput achieved on link $l$ $\mu_l(\boldsymbol{p})$ is given by

$$\mu_l(\boldsymbol{p}) = p_l \prod_{l \in I(l)} (1 - p_l), \quad (1)$$

where $I(l)$ denotes the set of links interfering with link $l$. We now introduce the *rate region* $\Lambda_{sA}$ of slotted Aloha algorithms as the set of rate vectors $\boldsymbol{\mu}(\boldsymbol{p}) = (\mu_1(\boldsymbol{p}), \dots, \mu_L(\boldsymbol{p}))$ that can be achieved on the various links for all possible values of the transmission probabilities $\boldsymbol{p}$:

$$\Lambda_{\rm sA} = \{\boldsymbol{v} \in [0,1]^L \mid \exists \boldsymbol{p} \in [0,1]^L, \text{s.t.} \boldsymbol{v} \le \boldsymbol{\mu}(\boldsymbol{p})\}, \quad (2)$$

where in the above expression, $\le$ is taken component-wise. Note that $\Lambda_{sA}$ is not convex and is strictly included in $\Gamma$, the set of achievable rates using centralized scheduling algorithms (see Section 3 for a formal definition). In Figure 1, we stress the difference between $\Lambda_{sA}$ and $\Gamma$ for the simplistic network of two interfering links. This difference illustrates the well-konwn inefficiency of slotted-Aloha protocols. All scheduling schemes based on these protocols inherit this inefficiency.

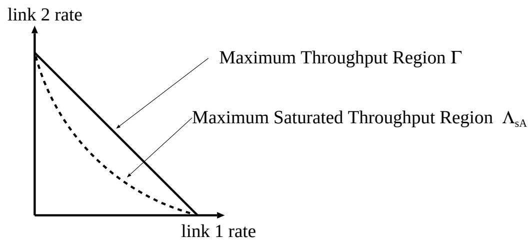


Fig. 1. $\Gamma$ and $\Lambda_{sA}$

In [12], the authors consider networks with saturated arrivals and single-hop connections, and aim at identifying the Proportional Fair point in $\Lambda_{sA}$. First notice that because of the non-convexity of $\Lambda_{sA}$, the problem has to be modified (a change in variables) so as to make the underlying optimization problem convex. It has been shown in [13] that the problem of identifying the $\alpha$-fair point in $\Lambda_{sA}$ can be transformed to a convex program provided that $\alpha \geq 1$. The algorithm proposed in [12] to achieve Proportional Fairness consists of adapting the transmission probabilities on each link, but requires the use of message passing.

In [29], the authors reverse-engineer the commonly used Exponential Back-off (EB) algorithm in slotted



*Wirel. Commun. Mob. Comput.* **00**: 1–12 (2009)
DOI: 10.1002/wcm

Aloha, i.e., they discovered the selfish utility maximization problems implicitly solved by the EB algorithm, and showed that the resulting Nash equilibrium is generally not optimal in network utility.

The aforementioned results have been generalized to the case of multi-hop connections, see e.g. [14] and [30]. The proposed algorithms again require message passing. In [15, 16], it has been shown that one can reduce message passing significantly in the case of networks with specific topologies.

The results have also been generalized to the case of unsaturated arrivals (packets are generated according to exogenous stochastic processes), in the case of single-hop connection in [31], and of multi-hop connections in [17].

### 2.2. Random access schemes with constant time control phase

The inefficiency of slotted-Aloha stems from the inevitable collisions and a lack of proper contention control. Interfering links with non-negligible transmission probabilities experience frequent collisions. To alleviate this problem, some authors propose to divide each time slot into two parts: a control part and a transmission part. The control part has $M$ mini-slots, where $M$ is a parameter. The $M$ mini-slots are used to sense the neighborhood activities, and decide the schedule of data transmissions. Here we survey this method using the algorithms proposed in [8, 11] and apply to the case of unsaturated sources. To simplify the exposition, we restrict our attention to the case of the single-hop interference model.

Let $A(v)$ be the incident links to node $v$. In each slot $t$, each link $l$ performs the following:

Step 1. Compute the *normalized queue length* $0 \le x_l(t) \le 1$ using queue lengths of the interfering neighbors via message passing:

$$x_l(t) = \frac{Q_l(t)}{\max\left[\sum_{k\in A(tx(l))} Q_k(t), \sum_{k\in A(rx(l))} Q_k(t)\right]}, \quad (3)$$

where $tx(l)$ and $rx(l)$ refer to the transmitter and the receiver of a link $l$.

Step 2. Contends for each mini-slot $m$ with probability $p_l = f(x_l(t), M)$ for a given function $f$, if the contention signals from its interfering links are not sensed prior to the mini-slot $m$.

The function $f(\cdot)$, referred to as the *access function*, controls the aggressiveness of the medium access algorithm and must be appropriately chosen to strike the balance between collisions and channel utilization. We may classify the algorithms in the literature according to the choice of the access function. Two types of access functions have been considered so far:

$$\text{Type I} : f(x_l(t), M) = g(M)\frac{x_l(t)}{M},$$

and

$$\text{Type II} : f(x_l(t), M) = 1 - \exp\Big(-g(M)\frac{x_l(t)}{M}\Big),$$

where $g(\cdot)$ is an increasing function.

In Type I algorithms, the choice of $g(M) = 1$ considered in [8] leads to a $(1/3 - 1/M)$-throughput optimal scheduling scheme[†]. The authors in [8] further show that $g(M) = (\sqrt{M} - 1)/2$ results in a $(1/2 - 1/\sqrt{M})$-throughput optimal scheduler. Using Type II algorithms, the authors of [10] show that with $g(M) = \log(2M)/2$, the achieved throughput ratio is at least $\frac{1}{2} - \frac{\log(2M)}{2M}$, which outperforms Type I algorithms.

### 2.3. CSMA

The random access algorithms presented in Section 2.1 and Section 2.2 have suboptimal, poor performance, and require message passing. One may actually wonder whether it is possible to design distributed schedulers achieving near optimal, performance without using message passing.

In [32, 33, 34], it has been shown that even *non-adaptive* CSMA algorithms, where each link accesses the channel with a fixed probability, are able to provide average throughputs close to optimality. Using *adaptive* CSMA protocols without message passing, it has been recently suggested in [26, 27] that one could fill this optimality gap. The idea behind the algorithms presented in these papers is similar to that developed by Hajek in [35], where a simulated-annealing approach is used to realize Max-Weight schedules.

More precisely, when the problem is to stabilize the network in the case of unsaturated users, the authors of [27, 36] propose to adapt the users' access rates as a function of their buffer sizes. When the number of packets waiting in a user's buffer becomes large, this user becomes more aggressive and increases its channel access rate. As discussed in [33], 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

[†]A scheme is $\gamma$-throughput optimal if for any vector $\boldsymbol{\lambda}$ in $\Gamma$, the scheme stabilizes the system for an arrival rate vector $\gamma \times \boldsymbol{\lambda}$.




DOI: 10.1002/wcm

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 [27] and [36] by implementing idealized continuous-time CSMA algorithms, in which Poisson clocks are used to control the channel accesses, and to ensure zero collisions. In practice, however, time is slotted and collisions cannot be avoided. More recently, [37] proposed a throughput-optimal algorithm in the slotted-time model, but requires the use of RTS/CTS-like message passing. Related work also includes [38] that proposed asymptotically throughput-optimal when there exists a large number of nodes, and [39] that improves throughput and fairness by reducing exposed and hidden nodes in 802.11 networks.

In [26], the authors developed distributed utility-optimal algorithms in the case of saturated users, also leveraging the simulated annealing technique. The proposed algorithms are adaptive CSMA without message passing in the idealized continuous-time model, In [28], based on which this paper is written, we presented a proof of convergence under no timescale separation and an analytical quantification of the impact collisions in a slotted-time system. Recall that the timescale separation assumption states that in between two updates of the CSMA parameters, the network dynamics converge immediately to an equilibrium distribution. In practice however, the network dynamics can be represented by a Markov process whose convergence to its stationary regime is not instantaneous. In the present paper, we provide a formal proof of convergence of adaptive CSMA-based algorithms that are extensions of those proposed in [26], without the timescale separation assumption. The proof is based on recent advances in the understanding of stochastic approximation algorithms with controlled random noise, see e.g. [40]. The proposed algorithms are then shown to maximize network utility in the continuous-time model. In the slotted-time model, a tradeoff between long-term efficiency and short-term fairness is revealed.

## 3. UO-CSMA (Utility-Optimal CSMA)

### 3.1. Model and objective

***Model***

We consider a wireless network composed of 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 with 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 $m \in \mathcal{N}$, $m_k = 1 = m_l$ implies that $A_{kl} = 0$). We assume that the transmitters can transmit at a fixed unit rate when active. The network is assumed to handle single-hop data connections. However, the results here can be extended to multi-hop connections. 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.

***Objective***

Denote by $\boldsymbol{\gamma}^s = (\gamma_l^s, l \in \mathcal{L})$ the long-term throughputs achieved by scheduling algorithm $s$ on the various links. The throughput vector of any scheduling algorithm has to belong to the *rate region* $\Gamma$ defined by

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

In the above, for any schedule $m \in \mathcal{N}$, $\pi_m$ can be interpreted as the proportion of time schedule $m$ is activated. As often in the case of saturated users, 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 solving the following optimization problem:

$$\begin{aligned} \max \quad & \Sigma_{l\in\mathcal{L}} U(\gamma_l), \\ \text{s.t.} \quad & \boldsymbol{\gamma} \in \Gamma. \end{aligned} \tag{4}$$

We denote by $\boldsymbol{\gamma}^\star = (\gamma_l^\star, l \in \mathcal{L})$ the optimizer of (4). Most distributed scheduling schemes proposed in the literature to solve (4) 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.

### 3.2. Efficiency of CSMA

CSMA-based random access algorithms are the most widely 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 model is due to Kelly [41], and has been recently revisited by e.g. [32] and [34]. 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 $\mu_l$. This CSMA algorithm is denoted by CSMA$(\lambda_l, \mu_l)$ 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$(\lambda_l, \mu_l)$, the network dynamics can be captured through a reversible process [42]: 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 $\mu_l$, 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 multiples of slots, which inevitably causes collisions. The impact of collisions is discussed in detail in Section 3.5.

Under the CSMA$(\lambda_l, \mu_l)$'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 [26] (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 ( [26])** *For any $\boldsymbol{\gamma}$ in the interior of $\Gamma$, 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 (4), one may use CSMA algorithms.

### 3.3. Utility-optimal adaptive CSMA algorithms

We now describe a generalized, adaptive CSMA-based algorithm to approximately solve (4). The algorithm is an extension of those proposed in [26], and does not require any message passing. Time is divided into *frames* of fixed durations, and the transmitter of each link updates its CSMA parameters (i.e., $\lambda_l$ and $\mu_l$ for link $l$) at the beginning of each frame. To do so, it maintains 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

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

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*; $V$, $q^{\min}$, and $q^{\max} (> q^{\min})$ are positive parameters, and $[\cdot]_c^d \equiv \max(d, \min(c, \cdot))$. We will later see that proper choice of $b$ ensures convergence. The parameter $V$ controls the accuracy of the algorithm.

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 $\lambda_l$'s (the transmission intensities) and fix the $\mu_l$'s (the transmission durations), or to update the $\mu_l$'s and fix the $\lambda_l$'s, or to update both the $\lambda_l$'s and $\mu_l$'s.

### 3.4. Convergence analysis

UO-CSMA may be interpreted as a stochastic approximation algorithm [43]. More precisely, we show that UO-CSMA is a stochastic approximation algorithm with *controlled Markov noise* as defined in [44]. 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 converge instantaneously in between CSMA parameter updates, i.e., if we could assume that $S_l[t]$ is the service received by link $l$ averaged over the stationary distribution $\pi^{\boldsymbol{\lambda}[t],\boldsymbol{\mu}[t]}$. But, this assumption is not realistic in this situation.

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$(\boldsymbol{\lambda}, \boldsymbol{\mu})$ algorithms, where for all $l \in \mathcal{L}$, $\lambda_l \mu_l = \exp(W(q_l))$. In other words,

$$\forall m \in \mathcal{N}, \quad \pi^{\boldsymbol{q}}_m = \frac{\exp(\sum_{l \in m} W(q_l))}{\sum_{m' \in \mathcal{N}} \exp(\sum_{l \in m'} W(q_l))}. \tag{5}$$

We also denote by $\boldsymbol{\gamma}[t] = (\gamma_l[t], l \in \mathcal{L})$ the vector representing the cumulative average throughputs of the various links up to frame $t$, i.e.,

$$\forall l \in \mathcal{L}, \quad \gamma_l[t] = \frac{1}{t} \sum_{n=0}^{t-1} S_l[n].$$

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

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

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

*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 $\boldsymbol{\pi}$):*

$$\begin{aligned} \textit{max} \quad & V \sum_{l \in \mathcal{L}} U(\gamma_l) - \Sigma_{m \in \mathcal{N}} \pi_m \log \pi_m \\ \textit{s.t.} \quad & \gamma_l \leq \sum_{m \in \mathcal{N}: m_l = 1} \pi_m, \ \sum_{m \in \mathcal{N}} \pi_m = 1. \end{aligned} \tag{6}$$

*Furthermore UO-CSMA approximately solves (4) as*

$$\Big| \sum_{l \in \mathcal{L}} \big(U(\gamma_{\star,l}) - U(\gamma_l^\star)\big) \Big| \leq \log |\mathcal{N}| / V. \tag{7}$$

*Proof.* As an important 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 $t$ were 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 deterministic trajectories (see Lemma 2). In the second step, we prove that the resulting averaged algorithm converges to the solution of (6). To do this, we use a similar approach as that used in [26]. The main contribution in our proof is in Step 1.

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 $t_n < t \leq t_{n+1}$,

$$\bar{q}_l(t) = q_l[n] + (q_l[n+1] - q_l[n]) \times \left(\frac{t - t_n}{t_{n+1} - t_n}\right). \tag{8}$$

Also define $S_l(t) = S_l[n] 1_{t_n \leq t < t_{n+1}}$.

**Lemma 2 (Convergence and averaging)** *Fix $\tau > 0$. Denote by $\tilde{q}$ the solution of the following system of*



DOI: 10.1002/wcm

*stochastic differential equations: for all $l \in \mathcal{L}$,*

$$\dot{q}_l = \left[U'^{-1}\Big(W_l(q_l)/V\Big) - S_l(t)\right] \times \frac{\mathbf{1}_{\{q^{\min} \le q_l \le q^{\max}\}}}{W'(q_l)}, \quad (9)$$

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

$$\lim_{\tau \to \infty} \sup_{t \in [\tau, \tau+T]} \|\bar{\boldsymbol{q}}(t) - \tilde{\boldsymbol{q}}(t)\| = 0 \quad a.s. \quad (10)$$

*Furthermore, almost surely, every limit point of the trajectories of (9) is a fixed point of the following system of ordinary differential equations (o.d.e.'s): for all $l \in \mathcal{L}$,*

$$\dot{q}_l = \left[U'^{-1}\Big(W_l(q_l)/V\Big) - \sum_{m \in \mathcal{N}: m_l = 1} \pi_m^q\right] \times \frac{\mathbf{1}_{\{q^{\min} \le q_l \le q^{\max}\}}}{W'(q_l)}. \quad (11)$$

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 $\tilde{\boldsymbol{q}}$. Furthermore, the fixed points of (9) are basically those of (11). Note that in the latter 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. 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 [40] and [44]. To complete the proof of Lemma 2, we check the sufficient conditions for convergence provided in [40]:
1) The transition kernel of $\boldsymbol{Y}[t]$, parameterized 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 $\mu_l$'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 [40][p. 71]): This is satisfied since $\boldsymbol{Y}[t]$ has a finite state-space (cf. conditions (6.4.1) and (6.4.2) in [40][pp.76]).

Having checked these conditions, we can now apply Lemma 4 and Theorem 7 (or Corollary 8) of [40], and the lemma follows. □

In view of Lemma 2, if the system of o.d.e.'s (11) has a unique fixed point $\boldsymbol{q}_\star$, then we would have $\lim_{t\to\infty} \tilde{\boldsymbol{q}}(t) = \boldsymbol{q}_\star$. We would also have $\lim_{t\to\infty} \boldsymbol{q}[t] = \boldsymbol{q}_\star$ a.s.. (This can be shown as in [45].)

Step 2. To complete the convergence proof, we show as in [26] that (11) may be interpreted as a sub-gradient algorithm (projected onto a bounded interval) of the dual of problem (6). Note that the latter problem is strictly convex, and hence this sub-gradient algorithm will converge to the solution of (6). This will imply that (11) converges to its unique fixed point, and will prove the convergence of UO-CSMA to the solution of (6) in view of Step 1.

The Lagrangian of (6) is given by

$$L(\boldsymbol{\gamma}, \boldsymbol{\pi}; \boldsymbol{\nu}, \eta) = \Big(\sum_{l \in \mathcal{L}} VU(\gamma_l) - \nu_l \gamma_l\Big) + \Big(\sum_{l \in \mathcal{L}} \nu_l \sum_{m \in \mathcal{N}: m_l = 1} \pi_m - \sum_{m \in \mathcal{N}} \pi_m \log \pi_m\Big) - \eta\Big(\sum_{m \in \mathcal{N}} \pi_m - 1\Big).$$

‡Note that equations (9) still have a stochastic component through the random variables $S_l(t)$, and they are referred to as *non-autonomous o.d.e.'s*.

Then the Karush-Kuhn-Tucker (KKT) conditions of (6) are given by

$$VU'(\gamma_l) = \nu_l,\ \forall l \in \mathcal{L}, \tag{12}$$

$$-1 - \log \pi_m + \sum_{l:m_l=1} \nu_l - \eta = 0,\ \forall m \in \mathcal{N}, \tag{13}$$

$$\nu_l \times (\gamma_l - \sum_{m\in\mathcal{N}:m_l=1} \pi_m) = 0, \tag{14}$$

$$\eta \times (\sum_{m\in\mathcal{N}} \pi_m - 1) = 0, \tag{15}$$

$$\forall l \in \mathcal{L}, \nu_l \geq 0. \tag{16}$$

We introduce the variables $\boldsymbol{q}$ with $q_l = W^{-1}(\nu_l)$ for $l \in \mathcal{L}$, and the bounds $\nu^{\max} = W(q^{\max})$ and $\nu^{\min} = W(q^{\min})$. By choosing

$$\eta = \log\left(\sum_m \exp(\sum_{l:m_l=1} W(q_l))\right) - 1,$$

and $\pi = \pi^{\boldsymbol{q}}$, we solve (13) and (15). Accounting for (14), the sub-gradient of (12) is

$$\dot{\nu}_l = \left( U'^{-1}\Big(\nu_l/V\Big) - \sum_{\substack{m\in\mathcal{N}\\ :m_l=1}} \pi_m^{\tilde{\boldsymbol{q}}} \right), \tag{17}$$

which is equivalent to (11), provided that $\nu_l(t)$ remains between $[\nu^{\min}, \nu^{\max}]$. Under Assumption 1, the solution $\boldsymbol{\nu}_\star = (\nu_{\star,l}, l \in \mathcal{L})$ of the dual of problem (6) without the constraints $\nu^{\min} \leq \nu \leq \nu^{\max}$ actually belongs to the interval $[\nu^{\min}, \nu^{\max}]$, and is the fixed point of (17). So by convexity of the problem, (17) converges to $\nu_\star$, and hence (11) converges to $\boldsymbol{q}_\star$, which concludes the proof of the convergence of UO-CSMA.

To prove the inequality (7), we just remark that (4) is equivalent to the following optimization problem:

$$\begin{aligned} \max \quad & V\Sigma_{l\in\mathcal{L}} U(\gamma_l) \\ \text{s.t.} \quad & \gamma_l \leq \Sigma_{m\in\mathcal{N}:m_l=1} \pi_m, \\ & \Sigma_{m\in\mathcal{N}} \pi_m = 1. \end{aligned} \tag{18}$$

Eq. (7) is obtained by comparing (6) and (18), 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 a large $V$ is slower convergence.

## 3.5. Slotted-time models: Collisions and tradeoff

### *Long and short collisions*

In the previous subsection, we have analyzed the convergence of UO-CSMA in the ideal continuous-time setting in which collisions are made mathematically impossible. In practical implementation, however, time is slotted and collisions may occur. We consider the following model for 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 to be 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$(p_l, \mu_l, \mu)$ and s-CSMA$(p_l, \mu_l)$ the above slotted CSMA algorithm with short and long collisions, respectively.

### *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



DOI: 10.1002/wcm

analyzed similarly. Assume that the transmitter of link $l$ implements the s-CSMA$(p_l, \mu_l)$ 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 further 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 $m'$; let $s(m \setminus m')$ be 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 $\exists k \in s(m) \cup c(m)$ such that $A_{kl} = 1$ (note that by convention, if $l \in n(m)$ then $l \notin m$) .

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

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

where $w(m,m')$ is the set of links that could be activated starting from schedule $m$ but that are not active in $m'$. In other words, $w(m,m') = \mathcal{L} \setminus (m \cup m' \cup n(m))$. Then one can show 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 probability distribution such that, for all* $m \in \{0,1\}^L \setminus \{\mathbf{0}\}$,

$$\pi_m^{\boldsymbol{p},\boldsymbol{\mu},s} = \pi_{\mathbf{0}}^{\boldsymbol{p},\boldsymbol{\mu},s} \prod_{l \in s(m)} (p_l \mu_l) \prod_{c \in c(m)} \left[ \mu_c \prod_{l \in c} p_l \right] \prod_{l \in n(m)} (1 - p_l).$$

*Then the following local balance equations are satisfied:*

$$\forall m, m', \quad \beta_{m,m'} \pi_m^{\boldsymbol{p},\boldsymbol{\mu},s} = \beta_{m',m} \pi_{m'}^{\boldsymbol{p},\boldsymbol{\mu},s}.$$

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

*Proof.* Note first that $s(m,m') = s(m',m)$ and $c(m,m') = c(m',m)$. Then remark that:

$$\frac{\prod_{l \in s(m \setminus m')} \frac{1}{\mu_l}}{\prod_{l \in s(m' \setminus m)} \frac{1}{\mu_l}} = \frac{\prod_{l \in s(m)} \frac{1}{\mu_l}}{\prod_{l \in s(m')} \frac{1}{\mu_l}},$$

and

$$\frac{\prod_{c \in c(m \setminus m')} \frac{1}{\mu_c}}{\prod_{c \in c(m' \setminus m)} \frac{1}{\mu_c}} = \frac{\prod_{c \in c(m)} \frac{1}{\mu_c}}{\prod_{c \in c(m')} \frac{1}{\mu_c}}.$$

Similarly,

$$\frac{\prod_{l \in s(m' \setminus m) \cup c(m' \setminus m)} p_l}{\prod_{l \in s(m \setminus m') \cup c(m \setminus m')} p_l} = \frac{\prod_{l \in s(m') \cup c(m')} p_l}{\prod_{l \in s(m) \cup c(m)} p_l}.$$

Finally, since

$$\prod_{l \in w(m,m')} (1 - p_l) = \frac{\prod_{l \in \mathcal{L}} (1 - p_l)}{\prod_{l \in m \cup m'} (1 - p_l) \prod_{l \in n(m)} (1 - p_l)},$$

we have

$$\frac{\prod_{l \in w(m,m')} (1 - p_l)}{\prod_{l \in w(m',m)} (1 - p_l)} = \frac{\prod_{l \in n(m')} (1 - p_l)}{\prod_{l \in n(m)} (1 - p_l)}.$$

We deduce that

$$\frac{\beta_{m,m'}}{\beta_{m',m}} = \frac{\pi_{m'}^{\boldsymbol{p},\boldsymbol{\mu},s}}{\pi_m^{\boldsymbol{p},\boldsymbol{\mu},s}}.$$

□

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

$$\forall l \in \mathcal{L}, \quad \gamma_l^{\boldsymbol{p},\boldsymbol{\mu},s} = \sum_{m \in \mathcal{N} : m_l = 1} \pi_m^{\boldsymbol{p},\boldsymbol{\mu},s}. \tag{19}$$

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 $\mu$ negligible compared to the channel holding times $\mu_l$'s (for the case with



DOI: 10.1002/wcm

short collisions), or (ii) to keep the transmission probabilities $p_l$'s close to 0, and to have very large channel holding times. Condition (i) could be ensured using RTS/CTS-like procedures and having very large channel holding times. Condition (ii) would be met if for all $l \in \mathcal{L}$, $p_l = \delta\alpha_l$ and $\mu_l = \xi_l/\delta$ with $\delta << 1$. In such case, in view of Lemma 3 and (19), we have

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

where for all $l \in \mathcal{L}$, $C_l > 0$ is a constant depending on the $\alpha_l$'s and $\xi_l$'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 $\lambda_l$'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 $\delta$, 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 $\gamma_{\star,l} - C_l\delta + o(\delta)$.

### *Short-term fairness vs. Long-term efficiency*

As we discussed at the end of Section 3.4, if we want the resulting link throughputs of UO-CSMA to be close to the solution of (4), 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 $\alpha$-fair utility function and refers to fairness at equilibrium.

For illustrative purpose, 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 (4) 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 $\xi_l[t]$. 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 $\delta$ as $\epsilon$, 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 $l > 1$,

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

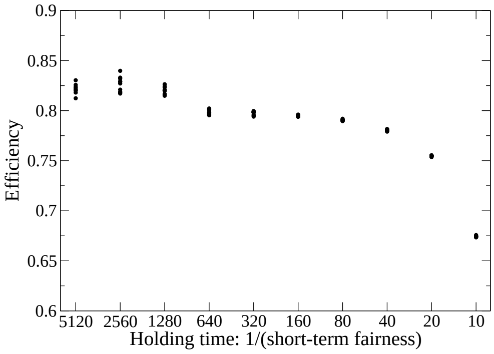


Fig. 2. Efficiency vs. short-term fairness tradeoff in a 3-link linear network. Algorithm parameters: $b[t] = 0.001$, $W(x) = x$, $V = 1$, $\epsilon\alpha^{\max} = 0.1$.

Achieving $\gamma_{\star,l} \geq \gamma_l^\star(1-\epsilon/2)$ for all links $l$ requires that $\xi_{\star,1} \approx \xi_\star^L/L$ and 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 [46], we now have that for all links $l$ the short-term fairness index $1/T_l$ scales as $\epsilon^{2L}$. This quantifies the trade-off between efficiency and short-term fairness when implementing UO-CSMA in slotted-time systems. Achieving high efficiency requires a considerable deterioration in short-term fairness: in the above star network, to ensure that the throughputs are at a distance at most $\epsilon$ from the utility-optimal throughputs, the short-term fairness index has to scale as $\epsilon^{2L}$.

We illustrate this tradeoff numerically using a simple 3-link linear network, where links 1 and 2 (resp. 3 and 2) are interfering with each other, but links 1 and 3 are interference-free. Figure 2 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 quite good efficiency for random access without message passing, e.g., 85%. Pushing this efficiency even higher would sacrifice short-term fairness further.

## 4. 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 the assumption that the network dynamics converge to an equilibrium in between parameter updates, that indeed this is true for the idealized, continuous-time model. However, there is an exponentially large price to pay in terms of short-term fairness in the more realistic, slotted-time model. The algorithm development and convergence proof techniques have been based on a combination of the 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 maximum queue stability, in addition to maximum rate stability or utility optimality, 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.

## Acknowledgement

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, the Air Force Office of Scientific Research under Grants FA9550-08-1-0480 and FA9550-09-1-0134, the Office of Naval Research Young Investigator Award N00014-07-1-0864, and Presidential Early Career Award for Scientists and Engineers N00014-09-1-0449.

## References

1. Tassiulas L, Ephremides A. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control* 1992; **37**(12):1936–1949.
2. Tassiulas L. Linear complexity algorithms for maximum throughput in radionetworks and input queued switches. *Proceedings of IEEE Infocom*, San Francisco, CA, 1998.
3. Chaporkar P, Kar K, Sarkar S. Throughput guarantees through maximal scheduling in wireless networks. *Proceedings of the 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2005.
4. Modiano E, Shah D, Zussman G. Maximizing throughput in wireless networks via gossiping. *Proceedings of ACM Sigmetrics*, Saint Malo, France, 2006.



DOI: 10.1002/wcm

5. Eryilmaz A, Ozdaglar A, Modiano E. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. *Proceedings of Infocom*, Anchorage, AK, 2007.
6. Sanghavi S, Bui L, Srikant R. Distributed link scheduling with constant overhead. *Proceedings of ACM Sigmetrics*, San Diego, CA, 2007.
7. Ray S, Sarkar S. Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning. *Proceedings of the Information Theory and Applications Second Workshop*, La Jolla, CA, 2007.
8. Joo C, Shroff NB. Performance of random access scheduling schemes in multi-hop wireless networks. *Proceedings of Infocom*, Anchorage, AK, 2007.
9. Yi Y, Chiang M. Wireless scheduling with O(1) complexity for m-hop interference model. *Proceedings of IEEE International Conference on Communications*, Beijing, China, 2008.
10. Gupta A, Lin X, Srikant R. Low-complexity distributed scheduling algorithms for wireless networks. *Proceedings of IEEE Infocom*, Anchorage, AK, 2007.
11. Lin X, Rasool S. Constant-time distributed scheduling policies for ad hoc wireless networks. *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, 2006.
12. Kar K, Sarkar S, Tassiulas L. Achieving proportional fairness using local information in aloha networks. *IEEE Transactions on Automatic Control* 2004; **49**(10):1858–1862.
13. Lee JW, Chiang M, Calderbank R. Utility-optimal random-access control. *IEEE Transactions on Wireless Communications* 2007; **25**:1135–1147.
14. Wang X, Kar K. Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access. *Proceedings of ACM Mobihoc*, Urbana-Champaign, IL, 2005.
15. Mohsenian-Rad AH, Huang J, Chiang M, Wong VWS. Utility-optimal random access: Optimal performance without frequent explicit message passing. *IEEE Transactions on Wireless Communications* 2009; **8**(2):898–911.
16. Mohsenian-Rad AH, Huang J, Chiang M, Wong VWS. Utility-optimal random access: Reduced complexity, fast convergence, and robust performance. *IEEE Transactions on Wireless Communications* 2009; **8**(2):898–911.
17. Liu J, Stolyar A, Chiang M, Poor HV. Queue backpressure random access in multihop wireless networks: Optimality and stability. *IEEE Transactions on Information Theory* 2009; **55**(9).
18. Stolyar AL. Maximizing queueing network utility subject to statbility: greedy primal-dual algorithm. *Queueing Systems Theory and Applications* 2005; **50**(4):401–457.
19. Lin X, Shroff NB. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *Proceedings of IEEE Infocom*, Miami, FL, 2005.
20. Neely MJ, Modiano E, Li C. Fairness and optimal stochastic control for heterogeneous networks. *Proceedings of IEEE Infocom*, Miami, FL, 2005.
21. Eryilmaz A, Srikant R. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *Proceedings of IEEE Infocom*, Miami, FL, 2005.
22. Chen L, Low SH, Chiang M, Doyle JC. Joint optimal congestion control, routing, and scheduling in wireless ad hoc networks. *Proceeding of IEEE Infocom*, Barcelona, Spain, 2006.
23. Yi Y, Chiang M. Stochastic network utility maximization and wireless scheduling 2009. To be published as a book chapter in *Next-Generation Internet Architectures and Protocols*, Cambridge University Press. Also, available at http://lanada.kaist.ac.kr/pubs/scheduling.pdf.
24. Yi Y, Proutiere A, Chiang M. Complexity in wireless scheduling: Impact and tradeoffs. *Proceedings of ACM Mobihoc*, Hong Kong, China, 2008.
25. Yi Y, Zhang J, Chiang M. Delay and effective throughput of wireless scheduling in heavy traffic regimes: Vacation model for complexity. *Proceedings of ACM Mobihoc*, New Orleans, LA, 2009.
26. Jiang L, Walrand J. A CSMA distributed algorithm for throughput and utility maximization in wireless networks. *Proceedings of the 46th Annual Conference on Communication, Control and Computing*, Monticello, IL, 2008.
27. Rajagopalan S, Shah D. Distributed algorithm and reversible network. *Proceedings of the Conference on Information Science and Systems*, Princeton, NJ, 2008.
28. Liu J, Yi Y, Proutiere A, Chiang M, Poor HV. Random access without message passing: Throughput, fairness and tradeoffs. *MSR Technical Report* 2008; .
29. Lee JW, Tang A, Chiang M, Calderbank AR. Reverse engineering mac: A game-theoretic model. *IEEE Journal of Selected Areas in Communication* 2007; **25**(6):1135–1147.
30. Lee JW, Chiang M, Calderbank AR. Jointly optimal congestion and contention control based on network utility maximization. *IEEE Communication Letters* 2006; **10**(3):216–218.
31. Stolyar A. Dynamic distributed scheduling in random access network. *Journal of Applied Probability* 2008; **45**(2):297–313.
32. Durvy M, Thiran P. Packing approach to compare slotted and non-slotted medium access control. *Proceedings of IEEE Infocom*, Barcelona, Spain, 2006.
33. Proutiere A, Yi Y, Chiang M. Throughput of random access without message passing. *Proceedings of the Conference on Information Sciences and Systems*, Princeton, NJ, 2008.
34. Bordenave C, McDonald D, Proutiere A. Performance of random medium access control: An asymptotic approach. *Proceedings of ACM Sigmetrics*, Annapolis, MD, 2008.
35. Hajek B. Cooling schedules for optimal annealing. *Mathematics of Operations Research* 1988; **13**(2):311–329.
36. Shin J, Shah D, Rajagopalan S. Network adiabatic theorem: An efficient randomized protocol for contention resolution. *Proceedings of ACM Sigmetrics*, Seattle, WA, 2009.
37. Ni J, Srikant R. Distributed CSMA/CA algorithms for achieving maximum throughput in wireless networks. *Proceedings of the Information Theory and Applications Workshop*, La Jolla, CA, 2009.
38. Marbach P, Eryilmaz A. A backlog-based csma-mechanism to achieve fairness and throughput-optimality in multihop wireless networks. *Proceedings of the 46th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2008.
39. Jiang LB, Liew SC. Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks. *IEEE Transactions on Mobile Computing* 2008; **7**(1):34–49.
40. Borkar V. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency (Cambridge University Press), 2008.
41. Kelly F. Stochastic models of computer communication systems. *Journal of the Royal Statistical Society* 1985; **47**(3):379–395.
42. Kelly F. *Reversibility and Stochastic Networks*. Wiley, Chichester, 1979.
43. Kushner HJ, Yin GG. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
44. Borkar V. Stochastic approximation with controlled markov noise. *Systems and Control Letters* 2006; **55**:139–145.
45. Benaim M. A dynamical system approach to stochastic approximation. *SIAM Journal on Control and Optimization* 1996; **34**:437–472.
46. Bremaud P. *Markov Chain: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York, 1999.




DOI: 10.1002/wcm