

On the Design of Efficient CSMA Algorithms for Wireless Networks

J. Ghaderi and R. Srikant
Department of ECE and Coordinated Science Lab.
University of Illinois at Urbana-Champaign
{jghaderi, rsrikant}@illinois.edu

Abstract—Recently, it has been shown that CSMA algorithms which use queue length-based link weights can achieve throughput optimality in wireless networks. In particular, a key result by Rajagopalan, Shah, and Shin (2009) shows that, if the link weights are chosen to be of the form $\log \log(q)$ (where q is the queue-length), then throughput optimality is achieved. In this paper, we tighten their result by showing that throughput optimality is preserved even with weight functions of the form $\log(q)/g(q)$, where $g(q)$ can be a function that increases arbitrarily slowly. The significance of the result is due to the fact that weight functions of the form $\log(q)/g(q)$ seem to achieve the best delay performance in practice.

I. INTRODUCTION

Efficient operation of wireless networks has always been a difficult task due to the inherent broadcast nature of the wireless medium. Transmission by a user can cause an interference for its neighbors. If two neighboring users transmit at the same time, the *Signal-to-Noise-plus-Interference Ratio* (SINR) of the users' links could go below the required SINR for the successful decoding of data packets at their corresponding receivers. In this case, we say that their messages *collide* with each other. Therefore, multiple users can transmit at the same time provided that they do not cause significant interference for each other. The users need a distributed *Medium Access Control* (MAC) protocol to determine which users should transmit which makes the optimal operation even harder.

CSMA (Carrier sense Multiple access) type protocols are an important class of MAC protocols due to their simplicity of implementation, and have been widely used in practice. e.g., in WLANs (IEEE 802.11 Wi-Fi) or emerging wireless mesh networks. In these protocols, each user listens to the channel and can transmit, with some probability, only when the channel is not busy. Despite the extreme simplicity of the CSMA-type algorithms, their efficiency have been always questionable. In this paper, we consider efficient design of such CSMA-type algorithms that can achieve maximum throughput and good delay performance.

The wireless network can be modeled by its conflict graph (or interference model), where two communication links form two neighboring nodes in the conflict graph, if they cannot transmit simultaneously. The well-known result of Tassiulas and Ephremides [1] states that the Maximum Weight Scheduling (MWS) algorithm is throughput optimal, where weights are queue-lengths. However, for a general network, MWS

involves finding the maximum weight independent set of the conflict graph in each time slot which is a formidable task, and hence, is not implementable. This has led to a rich amount of literature on design of approximate algorithms to alleviate the computational complexity of the MWS algorithm.

Recently, it has been shown that it is possible to design CSMA algorithms that are throughput-optimal. Reference [9] develops an algorithm that adaptively chooses the CSMA parameters under a time-scale separation assumption, i.e., the CSMA Markov chain converges to its stationary distribution instantaneously compared to the time-scale of adaptation of the CSMA parameters. This time-scale separation assumption was later verified by a stochastic approximation type argument [10], [11]. In particular, an important recent work by Rajagopalan, Shah, and Shin [4] builds an algorithm upon a Metropolis-Hastings sampling mechanism (Glauber dynamics over the set of feasible schedules) along with selection of link weights to be of the form $\log \log(q)$ (q is the queue-length). To establish the efficiency of the algorithm, they present a novel adiabatic-like theorem for the underlying queueing network: by choosing the weights to be of the form $\log \log q$, the underlying Markov chain behaves in an adiabatic manner such that it remains close to its equilibrium distribution. Similar algorithms with fixed link weights were developed earlier in [8] and [12].

Although a weight function of the form $\log \log q$ stabilizes the network, the resulting scheduling algorithm reacts very slowly to changes in queue lengths which, in turn, results in a poor delay performance. In this paper, we show that, by choosing weights to be of the form $\log q/g(q)$, a network adiabatic property still holds. As a result, we prove that the CSMA algorithms with such weight functions are throughput optimal. The function g can grow arbitrarily slowly such that $\log q/g(q)$ behaves very similarly to $\log(q)$. The significance of the result is due to the fact that such weight functions seem to also achieve the best delay performance in practice.

The remainder of the paper is organized as follows. In section II, we briefly describe our model of wireless networks. The main results of the paper are presented in section III. Section IV is devoted to the proofs. Finally, we will end the paper with some concluding remarks.

II. WIRELESS NETWORK MODEL

Consider a set of nodes where each node could be a source and/or a destination for another source. For now, we assume a single hop communication scheme but the results are naturally extendable to the multihop case. Therefore, there are N communication links, each of which corresponds to a source-destination pair.

Time is slotted and arrival process to each link is assumed to be discrete-time, where $a_l(t)$ is the number of packets arriving at link l in time slot t . For simplicity, assume that $\{a_l(t)\}_{t=0}^{\infty}$, for $l = 1, \dots, N$, are independent Bernoulli processes with parameter $\lambda = [\lambda_l; l = 1, \dots, N]$. In each time slot, one packet could be successfully transmitted over a link if the link Signal-to-Interference-plus-Noise-Ratio (SINR) is high enough. We use the notion of the *conflict graph* to capture the interference constraints or technological ones¹. Let $G(V, E)$ denote the conflict graph of the wireless network, where each node in the conflict graph is a communication link in the wireless network. There is an edge $(l, k) \in E$ between nodes l and k if simultaneous transmissions over communication links l and k are not successful. Therefore, at each time slot, the active links should form an independent set of G , i.e., no two scheduled nodes can share an edge in G . Formally, a schedule can be represented by a vector $X = [x_s : s = 1, \dots, N]$ such that $x_s \in \{0, 1\}$ and $x_i + x_j \leq 1$ for all $(i, j) \in E$. Let \mathcal{M} denote the set of all feasible schedules.

Each link l is associated with a queue q_l , where the queue dynamics are given by

$$q_l(t) = (q_l(t-1) - x_l(t))^+ + a_l(t)$$

for $t \geq 0$ and $l = 1, \dots, N$. The vector of queue lengths is denoted by $q(t) = [q_l(t) : l = 1, \dots, N]$.

A scheduling algorithm is a policy to determine which schedule to be used in each time slot. The capacity region of the network is defined to be the set of all arrival rates λ that can be supported by the network, i.e., for which there exists a scheduling algorithm that can stabilize the queues. It is known, e.g. [1], that the capacity region is given by

$$\Lambda = \{\lambda \geq 0 : \exists \mu \in \text{Co}(\mathcal{M}), \lambda < \mu\}$$

where $\text{Co}(\cdot)$ is the convex hull operator. When dealing with vectors, inequalities are interpreted component-wise.

A scheduling algorithm is throughput-optimal if it can stabilize the network for any arrival rate in Λ . An important class of the throughput-optimal algorithms is the *maximum-weight scheduling* (MWS) algorithm where at each time slot t , the scheduling decision $\rho(t)$ satisfies

$$\rho(t) = \arg \max_{X \in \mathcal{M}} \sum_{l=1}^N x_l w_l(t).$$

where $w_l(t)$ is the weight of link l at time slot t . In [1], it was proved that the MWS algorithm is throughput-optimal for $w_l(t) = q_l(t)$. A natural generalization of the MWS algorithm

in [2] uses a weight $f(q_l(\cdot))$ instead of $q_l(\cdot)$ with the following properties.

- 1) $f : [0, \infty] \rightarrow [0, \infty]$ is a nondecreasing continuous function with $\lim_{q_l \rightarrow \infty} f(q_l) = \infty$.
- 2) Given any $M_1, M_2 > 0$, and $0 < \epsilon < 1$, there must exist a $Q < \infty$ such that for $q_l > Q$:

$$(1 - \epsilon)f(q_l) \leq f(q_l - M_1) \leq f(q_l + M_2) \leq (1 + \epsilon)f(q_l)$$

Lemma 1. *Suppose f is a strictly concave and monotonically increasing function, with $f(0) = 0$, then it satisfies the conditions (1) and (2) above.*

See the appendix of [14] for the proof. In this paper, we use a function f with properties of Lemma 1.

III. MAIN RESULT

A. Basic Algorithm

For our algorithm, we choose the weight of link l to be

$$\tilde{w}_l(t) = \max(w_l(t), w_{min}(t)) \quad (1)$$

where

$$w_l(t) = f(q_l(t)), \quad (2)$$

$$w_{min}(t) = \frac{\epsilon}{2N} f(q_{max}(t)), \quad (3)$$

$$f(x) = \frac{\log(1+x)}{g(x)}, \quad (4)$$

and $q_{max}(t)$ is the length of the largest queue in the network at time t and assumed to be known. The function $g(x)$ is a strictly increasing function chosen such that f is a strictly concave increasing function, for example $g(x) = \log(e + \log(1+x))$ or $g(x) = (\log(1+x))^\theta$ for some $0 < \theta < 1$. In this paper, all log's are in base e .

Consider the conflict graph $G(V, E)$ of the network as defined earlier. Denote the neighbors of i by a set $\mathcal{N}(i) = \{k \in V : (i, k) \in E\}$. At each time slot t , a node i is chosen uniformly at random, with probability $\frac{1}{N}$, then

- (i) If $x_j(t-1) = 0$ for all nodes $j \in \mathcal{N}(i)$, then $x_i(t) = 1$ with probability $\frac{\exp(\tilde{w}_i(t))}{1 + \exp(\tilde{w}_i(t))}$, and $x_i(t) = 0$ with probability $\frac{1}{1 + \exp(\tilde{w}_i(t))}$. Otherwise, $x_i(t) = 0$.
- (iii) $x_j(t) = x_j(t-1)$ for all $j \neq i$.

The following theorem states the main result regarding the throughput optimality of the algorithm.

Theorem 1. *Consider any $\epsilon > 0$. The algorithm can stabilize the network for any $\lambda \in (1 - \epsilon)\Lambda$, if the weight function is chosen to be in the form of $f(x) = \frac{\log(1+x)}{g(x)}$. The function $g(x)$ is a strictly increasing function chosen such that f is a strictly concave increasing function. In particular, the algorithm with the following weight functions is throughput-optimal: $f(x) = \frac{\log(1+x)}{\log(e + \log(1+x))}$ or $f(x) = (\log(1+x))^{1-\theta}$ for some $0 < \theta < 1$.*

¹For example, a node cannot transmit and receive at the same time.

B. Distributed Implementation

The basic algorithm is based on Glauber-Dynamics with one site update at each time. For distributed implementation, we need a randomized mechanism to select a link uniformly at each time slot. We use the Q-CSMA idea [3] to perform the link selection as follows. Each time slot is divided into a control slot and a data slot. The control slot, which is much smaller than the data slot, is used to generate a transmission schedule for the data slot. First, the network selects a set of links $m(t)$ that do not conflict with each other. Then, it performs the Glauber-Dynamics updates, in parallel, over links $m(t)$ to produce a transmission schedule $X(t)$ for data transmission. $m(t)$ is called the decision schedule at time t . For example, a simple randomized mechanism to generate $m(t)$ is as follows. In control slot t , each link l sends an INTENT message with probability $1/2$. If l does not hear any INTENT messages from its neighboring links $\mathcal{N}(l)$, it will be included in $m(t)$, otherwise it will not be included in $m(t)$. Therefore, by the end of the control slot, any feasible decision schedule $m(t) \subseteq \mathcal{M}$ could be selected with a positive probability $\alpha(m(t))$. Once a link knows whether it is included in the decision schedule, it can determine its state in the data slot based on its carrier sensing information (i.e., whether its conflicting links were active in the previous data slot) and the activation probability for the current slot (based on its queue length).

To determine the weight at each link l , $q_{max}(t)$ is needed. Instead, each link l can maintain an estimate of $q_{max}(t)$. We can use the procedure suggested in [4] to estimate $q_{max}(t)$, and use Lemma 2 of [4] to complete the stability proof. So we do not pursue this issue here. In practical networks $\frac{\epsilon}{2N} \log(1 + q_{max})$ is small and we can use the weight function f directly, and thus, there may not be any need to know $q_{max}(t)$.

Corollary 1. *Under the weight function f specified in Theorem 1, the distributed algorithm can stabilize the network for any $\lambda \in (1 - \epsilon)\Lambda$.*

IV. PROOF OF THE MAIN RESULT

Before we start the proof, some preliminaries, regarding stationary distribution and mixing time of Glauber dynamics, are needed.

A. Preliminaries

Consider a time-homogenous discrete-time Markov chain over the finite state-space \mathcal{M} . For simplicity, we index the elements of \mathcal{M} by $1, 2, \dots, r$, where $r = |\mathcal{M}|$. Assume the Markov chain is irreducible and aperiodic, so that a unique stationary distribution $\pi = [\pi(1), \dots, \pi(r)]$ always exists.

1) *Distance between probability distributions:* First, we introduce two convenient norms on \mathbb{R}^r that are linked to the stationary distribution. Let $\ell^2(\pi)$ be the real vector space \mathbb{R}^r endowed with the scalar product

$$\langle z, y \rangle_\pi = \sum_{i=1}^r z(i)y(i)\pi(i).$$

Then, the norm of z with respect to π is defined as

$$\|z\|_\pi = \left(\sum_{i=1}^r z(i)^2 \pi(i) \right)^{1/2}.$$

We shall also use $\ell^2(\frac{1}{\pi})$, the real vector space \mathbb{R}^r endowed with the scalar product

$$\langle z, y \rangle_{\frac{1}{\pi}} = \sum_{i=1}^r z(i)y(i) \frac{1}{\pi(i)}$$

and its corresponding norm. For any two strictly positive probability vectors μ and π , the following relationship holds

$$\|\mu - \pi\|_{\frac{1}{\pi}} = \left\| \frac{\mu}{\pi} - 1 \right\|_\pi \geq 2\|\mu - \pi\|_{TV},$$

where $\|\pi - \mu\|_{TV}$ is the total variation distance

$$\|\pi - \mu\|_{TV} = \frac{1}{2} \sum_{i=1}^r |\pi(i) - \mu(i)|.$$

2) *Glauber dynamics:* Consider a graph $G(V, E)$. Glauber dynamics is a Markov chain to generate the independent sets of G . So, the state space \mathcal{M} consists of all independent sets of G . Let $|V| = N$. Given a weight vector $\tilde{W} = [\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_N]$, at each time t , a node i is chosen uniformly at random, with probability $\frac{1}{N}$, then

- (i) If $x_j(t-1) = 0$ for all nodes $j \in \mathcal{N}(i)$, then $x_i(t) = 1$ with probability $\frac{\exp(\tilde{w}_i)}{1 + \exp(\tilde{w}_i)}$, or $x_i(t) = 0$ with probability $\frac{1}{1 + \exp(\tilde{w}_i)}$. Otherwise, $x_i(t) = 0$.
- (iii) $x_j(t) = x_j(t-1)$ for all $j \neq i$.

The corresponding Markov chain is irreducible, aperiodic, and reversible over \mathcal{M} , and its stationary distribution is given by

$$\pi(\rho) = \frac{1}{Z} \exp\left(\sum_{i \in \rho} \tilde{w}_i\right); \quad \rho \in \mathcal{M}, \quad (5)$$

where Z is the normalizing constant.

The basic algorithm uses a time-varying version of the above Glauber dynamics, where the weights change with time. This yields a time-inhomogeneous Markov chain but we will see that, for the proper choice of weights, it behaves similarly to the Glauber dynamics.

3) *Mixing time of Glauber dynamics:* The convergence to steady state distribution is geometric with a rate equal to the *second largest eigenvalue modulus* (SLEM) of the transition matrix as it is described next [6].

Lemma 2. *Let P be an irreducible, aperiodic, and reversible transition matrix on the finite state space \mathcal{M} with the stationary distribution π . Then, the eigenvalues of P are ordered in such a way that*

$$\lambda_1 = 1 > \lambda_2 \geq \dots \geq \lambda_r > -1,$$

and for any initial probability distribution μ_0 on \mathcal{M} , and for all $n \geq 1$

$$\|\mu_0 \mathbf{P}^n - \pi\|_{\frac{1}{\pi}} \leq \sigma^n \|\mu_0 - \pi\|_{\frac{1}{\pi}}, \quad (6)$$

where $\sigma = \max\{\lambda_2, |\lambda_r|\}$ is the SLEM of P .

The following Lemma gives an upper bound on the SLEM $\sigma(P)$ of Glauber dynamics.

Lemma 3. *For the Glauber Dynamics with the weight vector \tilde{W} on a graph $G(V, E)$ with $|V| = N$,*

$$\sigma \leq 1 - \frac{1}{16^N \exp(4N\tilde{w}_{max})},$$

where $\tilde{w}_{max} = \max_{i \in V} \tilde{w}_i$.

See [14] for the proof. We define the mixing time as $T = \frac{1}{1-\sigma}$, so

$$T \leq 16^N \exp(4N\tilde{w}_{max}) \quad (7)$$

Simple calculation, based on Lemma 2, reveals that the amount of time needed to get close to the stationary distribution is proportional to T .

B. A key lemma

At any time slot t , given the weight vector $\tilde{W}(t) = [\tilde{w}_1(t), \dots, \tilde{w}_N(t)]$, the MWS algorithm should solve

$$\max_{\rho \in \mathcal{M}} \sum_{i \in \rho} \tilde{w}_i(t),$$

instead, our algorithm tries to simulate a distribution

$$\pi_t(\rho) = \frac{1}{Z} \exp\left(\sum_{i \in \rho} \tilde{w}_i(t)\right); \quad \rho \in \mathcal{M}, \quad (8)$$

i.e., the stationary distribution of Glauber dynamics with the weight vector $\tilde{W}(t)$ at time t .

Let P_t denote the transition probability matrix of Glauber dynamics with the weight vector $\tilde{W}(t)$. Also let μ_t be the true probability distribution of the inhomogeneous-time chain, over the set of schedules \mathcal{M} , at time t . Therefore, we have $\mu_t = \mu_{t-1}P_t$. Let π_t denote the stationary distribution of the time-homogenous Markov chain with $P = P_t$ as in (8). By choosing proper w_{min} and $f(\cdot)$, we aim to ensure that μ_t and π_t are close enough, i.e.,

$$\|\pi_t - \mu_t\|_{TV} \leq \delta/4$$

for some δ arbitrary small.

Let $w_{max}(t) = f(q_{max}(t))$. The following lemma gives a sufficient condition under which the probability distribution of the inhomogeneous Markov chain is close to the stationary distribution of the homogenous chain.

Lemma 4. *Given any $\delta > 0$, $\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4}$ holds for all $t \geq t^*$, if*

$$\alpha_t T_{t+1} \leq \delta/16 \text{ for all } t > 0, \quad (9)$$

where

- (i) $\alpha_t = 2Nf'(f^{-1}(w_{min}(t+1)) - 1)$,
- (ii) t^* is the smallest t such that

$$\sum_{k=1}^t \frac{1}{T_k^2} \geq \ln(4/\delta) + N(w_{max}(0) + \log 2)/2, \quad (10)$$

and T_{t+1} is the mixing time of the Glauber dynamics with the weight vector $\tilde{W}(t+1)$.

See [14] for the proof. Lemma 4 states a condition under which $\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4}$ for all $t \geq t^*$. Instead, assume that (9) holds only when $\|q(t)\| \geq q_{th}^2$ for a constant $q_{th} > 0$. Let t_1 be the first time that $\|q(t)\|$ hits q_{th} . Then, after that, it takes t^* time slots for the chain to get close to π_t if $\|q(t)\|$ remains above q_{th} for $t_1 \leq t \leq t_1 + t^*$. Alternatively, we can say that $\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4}$ if $\|q(t)\| \geq q_{th} + t^*$ since at each time slot at most one departure can happen and this guarantees that $\|q(t)\| \geq q_{th}$ for, at least, the past t^* time slots. This immediately implies the following Lemma that we will use in the proof of the main result.

Lemma 5. *Given any $\delta > 0$, $\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4}$ holds when $\|q(t)\| \geq q_{th} + t^*$, if there exists a q_{th} such that*

$$\alpha_t T_{t+1} \leq \delta/16 \text{ whenever } \|q(t)\| > q_{th}, \quad (11)$$

where

- (i) $\alpha_t = 2Nf'(f^{-1}(w_{min}(t+1)) - 1)$
- (ii) $T_t \leq 16^N \exp(4Nw_{max}(t))$
- (ii) t^* is the smallest t such that

$$\sum_{k=t_1: \|q(t_1)\| = q_{th}}^{t_1+t^*} \frac{1}{T_k^2} \geq \ln(4/\delta) + N(f(q_{th}) + \log 2)/2. \quad (12)$$

In the above Lemma, condition (ii) is based on the upper bound of (7) and the fact that $\tilde{w}_{max}(t) = w_{max}(t)$.

In other words, Lemma 5 states that when queue lengths are large, the observed distribution of the schedules is close to the desired stationary distribution.

Remark 1. *We will later see that, to satisfy condition (11) and to find a finite t^* satisfying (12) in Lemma 5, the function $f(\cdot)$ cannot be faster than $\log(\cdot)$. In fact, the function f must be slightly slower than $\log(\cdot)$ to make the weight dynamics slow enough such that the distribution of the schedules remains close to the stationary distribution.*

Remark 2. *The above Lemma is a generalization of Lemma 12 (Network Adiabatic Theorem) of [4]. Here we consider general functions $f(\cdot)$, whereas [4] considers a particular function $\log \log(\cdot)$. The generalization allows us to use functions which are close to $\log(\cdot)$ and perform much better than $\log \log(\cdot)$ in simulations. The proof of Lemma 4 is presented in the appendix of [14].*

C. Throughput optimality

Throughput optimality follows from the following Lemma [2]. (The proof has been eliminated due to page constraints, see [14] for the complete proof.)

Lemma 6. *For a scheduling algorithm, if given any $0 < \epsilon < 1$ and $0 < \delta < 1$, there exists a $B(\delta, \epsilon) > 0$ such that: in any*

²In this paper, $\|q(t)\| = \|q(t)\|_\infty = \max_i q_i(t) = q_{max}(t)$.

time slot t , with probability larger than $1 - \delta$, the scheduling algorithm chooses a schedule $X(t) \in \mathcal{M}$ that satisfies

$$\sum_{i \in X(t)} w_i(t) \geq (1 - \epsilon) \max_{\rho \in \mathcal{M}} \sum_{i \in \rho} w_i(t)$$

whenever $\|q(t)\| > B(\delta, \epsilon)$, then the scheduling algorithm is throughput-optimal.

Our algorithm satisfies the above condition for

$$B = \max \left\{ q_{th} + t^*, f^{-1} \left(\frac{N \log 2 + \log \frac{2}{\delta}}{\epsilon/2} \right) \right\}. \quad (13)$$

Remark 3. Throughput optimality in Lemma 6 means that, for all the rates inside the capacity region, the Markov chain describing the system will be positive recurrent.

D. A class of weight functions with the maximum throughput property

In this section, we describe a family of weight functions f that yield a maximum throughput algorithm.

The function f needs to satisfy Lemma 5. Roughly speaking, since the mixing time T is exponential in w_{max} , $f'(f^{-1}(w_{min}))$ must be in the form of $e^{-w_{min}}$; otherwise it will be impossible to satisfy $\alpha_t T_{t+1} < \delta/16$ for any arbitrarily small δ as $\|q(t)\| \rightarrow \infty$. The only function with such a property is the $\log(\cdot)$ function. In fact, it turns out that f must grow slightly slower than $\log(\cdot)$ as we show next to satisfy (11), and to ensure the existence of a finite t^* in Lemma 5.

Consider weight functions of the form $f(x) = \frac{\log(1+x)}{g(x)}$ where $g(x)$ is a strictly increasing function, chosen such that f satisfies the conditions of Lemma 1. For example, by choosing functions that grow much slower than $\log(1+x)$, like $g(x) = \log(e + \log(1+x))$, we can make $f(x)$ behave approximately like $\log(1+x)$ for large ranges of x .

Assume $g(0) \geq 1$, then

$$f'(x) \leq \frac{1}{1+x}. \quad (14)$$

The inverse of f cannot be expressed explicitly, however, it can be written as

$$f^{-1}(x) = \exp(xg(f^{-1}(x))) - 1. \quad (15)$$

Therefore,

$$f'(f^{-1}(w_{min}) - 1) \leq \frac{1}{f^{-1}(w_{min})} \quad (16)$$

$$= \frac{1}{\exp(w_{min}g(f^{-1}(w_{min}))) - 1} \quad (17)$$

Using (16), the conditions of Lemma 5 are satisfied if there exists a q_{th} large enough such that

$$2N16^N \exp(4Nw_{max}) \frac{1}{\exp(w_{min}g(f^{-1}(w_{min}))) - 1} \leq \delta/16 \quad (18)$$

for $\|q(t)\| \geq q_{th}$.

Using (15) and noting that $w_{min} = \frac{\epsilon}{2N}w_{max}$, (18) can be written as

$$2N16^N \exp \left(w_{min} \left[\frac{8N^2}{\epsilon} - g(f^{-1}(w_{min})) \right] \right) \quad (19)$$

$$\times \left(1 + \frac{1}{f^{-1}(w_{min})} \right) \leq \delta/16 \quad (20)$$

Consider fixed, but arbitrary, N and ϵ . As $q_{max} \rightarrow \infty$, $w_{max} \rightarrow \infty$, and consequently $w_{min} \rightarrow \infty$ and $f^{-1}(w_{min}) \rightarrow \infty$. Therefore, the exponent $\frac{8N^2}{\epsilon} - g(f^{-1}(w_{min}))$ is negative for q_{max} large enough, and thus, there is a threshold q_{th} such that for all $q_{max} > q_{th}$, the condition (19) is satisfied. To be more accurate, it suffices to choose

$$q_{th} = f^{-1} \left(\frac{2N}{\epsilon} \times \max \left\{ \log \left(\frac{64N16^N}{\delta} \right), f \left(g^{-1} \left(\frac{16N^2}{\epsilon} \right) \right) \right\} \right). \quad (21)$$

Then, it follows from Lemma 5 that $\|\pi_t - \mu_t\|_{TV} \leq \frac{\delta}{4}$, whenever $\|q(t)\| > q_{th} + t^*$.

Remark 4. The assumption $g(0) \geq 1$ is not required, since, as we saw in the above analysis, only the asymptotic behavior of g is important. If we choose q_{th} large enough such that

$$g(f^{-1}(w_{min}(t)) - 1) \geq 1 \quad (22)$$

when $\|q(t)\| \geq q_{th}$, then (16) holds and the rest of the analysis follows exactly. In particular, in order to get an explicit formula for f^{-1} , we can choose $g(x) = \log(1+x)^\theta$ for some $0 < \theta < 1$. The weight function for such a g is $f(x) = (\log(1+x))^{1-\theta}$, and f^{-1} has the closed form

$$f^{-1}(x) = \exp(x^{\frac{1}{1-\theta}}) - 1.$$

Then (21) yields

$$q_{th} = \exp \left(\max \left\{ \frac{2N}{\epsilon} \log \left(\frac{64N16^N}{\delta} \right), \frac{2N}{\epsilon} \left(\frac{16N^2}{\epsilon} \right)^{\frac{1}{\theta}} \right\}^{\frac{1}{1-\theta}} \right). \quad (23)$$

It is easy to check that for $q(t) \geq \exp \left(\left(\frac{2N}{\epsilon} \right)^{\frac{1}{1-\theta}} \log(1+e) \right)$, $w_{min}(t) \geq f(e)$ which satisfies (22). Therefore, obviously, (22) also holds for q_{th} of (23).

The last step of the proof is to determine the constant B in (13), so we need to find t^* . Let t_1 be the first time that $q_{max}(t)$ hits q_{th} , then

$$\begin{aligned} \sum_{k=t_1}^{t_1+t} \frac{1}{T_k^2} &\geq 16^{-2N} \sum_{k=t_1}^{t_1+t} e^{-8Nf(q_{max}(k))} \\ &= 16^{-2N} \sum_{k=t_1}^{t_1+t} e^{-8N \frac{\log(1+q_{max}(k))}{g(q_{max}(k))}} \\ &= 16^{-2N} \sum_{k=t_1}^{t_1+t} (1 + q_{max}(k))^{-\frac{8N}{g(q_{max}(k))}} \\ &\geq 16^{-2N} \sum_{k=1}^t (1 + q_{th} + k)^{-\frac{8N}{g(q_{th})}} \\ &\geq 16^{-2N} t(1 + q_{th} + t)^{-\frac{8N}{g(q_{th})}} \end{aligned}$$

Therefore, by Lemma 5, it suffices to find the smallest t that satisfies

$$16^{-2N}t(1 + q_{th} + t)^{-\frac{8N}{g(q_{th})}} \geq \log(4/\delta) + \frac{N}{2} \log(2(1 + q_{th}))$$

for a threshold q_{th} large enough satisfying (21). Recall that $g(\cdot)$ is an increasing function, therefore, by choosing q_{th} large enough, $\frac{8N}{g(q_{th})}$ can be made arbitrary small. Then a finite t^* always exists since

$$\lim_{t^* \rightarrow \infty} t^*(1 + q_{th} + t^*)^{-\frac{8N}{g(q_{th})}} = \infty.$$

This concludes the proof of the main Theorem.

E. Extension of the proof to the distributed implementation

The distributed algorithm is based on multiple site-update (or parallel operating) Glauber dynamics as defined next. Consider the graph $G(V, E)$ as before and a weight vector $\tilde{W} = [\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_N]$. At each time t , a decision schedule $m(t) \subseteq \mathcal{M}$ is selected at random with positive probability $\alpha(m(t))$. Then, for all $i \in m(t)$,

- (i) If $x_j(t-1) = 0$ for all nodes $j \in \mathcal{N}(i)$, then $x_i(t) = 1$ with probability $\frac{\exp(\tilde{w}_i)}{1 + \exp(\tilde{w}_i)}$, or $x_i(t) = 0$ with probability $\frac{1}{1 + \exp(\tilde{w}_i)}$.
Otherwise, $x_i(t) = 0$.

- (ii) $x_j(t) = x_j(t-1)$ for all $j \notin m(t)$.

The Markov chain $X(t)$ is aperiodic and irreducible if $\cup_{m \in \mathcal{M}_0} m = V$ (See [3] for more detail). Also, it can be shown that $X(t)$ is reversible, and it has the same stationary distribution as regular Glauber dynamics in (8). Here, we will assume that $\alpha_{min} := \min_m \alpha(m) \geq (1/2)^N$. Then, the mixing time of the chain is characterized by the following Lemma.

Lemma 7. *For the multiple site-update Glauber Dynamics with the weight vector \tilde{W} on a graph $G(V, E)$ with $|V| = N$,*

$$T \leq \frac{64^N}{2} \exp(4N\tilde{w}_{max}). \quad (24)$$

where $\tilde{w}_{max} = \max_{i \in V} \tilde{w}_i$.

See [14] for the proof. The distributed algorithm uses a time-varying version of the multiple-site update Glauber dynamics, where the weights change with time. Although the upperbound of Lemma 7 is loose, it is sufficient to prove the optimality of the algorithm. The analysis is the same as the argument for the basic algorithm. Let D and W denote the lengths of the data slot and the control slot. Thus, the distributed algorithm can achieve a fraction $\frac{D}{D+W}$ of the capacity region. In particular, recall the simple randomized mechanism, in section III-B, where each node joins the decision schedule by sending an INTENT message with probability $1/2$. Note that in this case $\alpha_{min} \geq (1/2)^N$, and also it suffices to allocate a short mini-slot at the beginning of the slot for the purpose of control. By choosing the data slot to be much larger than the control slot, the algorithm can approach the full capacity.

Remark 5. *Since we completed the paper, we learned that the authors of [4], [13] also know the fact that general*

weight functions $f(q)$ also achieve throughput optimality if the function $f(\cdot)$ satisfies conditions very similar to our condition in (18). However, since the result did not appear in the published [4] or online version [13] of their paper, we were unaware of this extension.

V. CONCLUSIONS

In this paper, we considered the design of efficient CSMA algorithms that are throughput optimal and have a good delay performance. The algorithm is essentially a Glauber Dynamics with, potentially, multiple-site updates at each time-slot. Access probabilities depend on links weights, where the weight of each link is chosen to be an appropriate function of its queue-length. In particular, we showed that weight functions of the form $f(q) = \log(q)/g(q)$ yield throughput-optimality and low delay performance. The function $g(q)$ can grow arbitrarily slowly such that $f(q) \approx \log(q)$.

ACKNOWLEDGMENTS

The authors would like to thank Bo (Rambo) Tan, at the Coordinated Science Lab., for his help with the simulations. The research was funded by AFOSR Grant FA-9550-08-1-0432, Army MURIs, DTRA Grant HDTRA1-08-1-0016, and NSF Grant CNS 07-21286.

REFERENCES

- [1] L. Tassiulas and A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximal throughput in multihop radio networks, *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1948, December 1992.
- [2] A. Eryilmaz, R. Srikant, and J. R. Perkins, Stable scheduling policies for fading wireless channels. *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411-424, April 2005.
- [3] J. Ni, B. Tan and R. Srikant, Q-CSMA: Queue length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks, *IEEE INFOCOM Mini-Conference*, 2010.
- [4] S. Rajagopalan, D. Shah and J. Shin, Network adiabatic theorem: an efficient randomized protocol for contention resolution, *ACM SIGMETRICS/Performance*, pp. 133-144, 2009.
- [5] M. Jerrum and A. Sinclair, Approximating the permanent, *SIAM Journal of Computing*, vol. 18, pp. 1149-1178, 1989.
- [6] P. Bremaud, Markov chains, Gibbs fields, Monte Carlo simulation, and queues, *Springer-Verlag*, New York 1999, 2nd edition, 2001.
- [7] S. P. Meyn and R. L. Tweedie, Criteria for stability of Markovian processes I: Discrete time chains, *Advances in Applied Probability*, vol. 24, pp. 542-574, 1992.
- [8] R. R. Boorstyn, A. Kershbaum, B. Maglaris, and V. Sahin, Throughput analysis in multihop CSMA packet radio networks, *IEEE Transactions on Communications*, vol. 35, no. 3, pp. 267-274, March 1987.
- [9] L. Jiang and J. Walrand, A distributed CSMA algorithm for throughput and utility maximization in wireless networks, *46th Annual Allerton Conference on Communication, Control and Computing*, September 2008.
- [10] L. Jiang and J. Walrand, Convergence and stability of a distributed CSMA algorithm for maximal network throughput, *IEEE Conference on Decision and Control*, 2009.
- [11] A. Proutiere, Y. Yi, and M. Chiang, Throughput of random access without message passing, *Proc. of CISS*, Princeton, NJ, March 2008.
- [12] S. C. Liew, C. Kai, J. Leung, B. Wong, Back-of-the-envelope computation of throughput distributions in CSMA wireless networks, *IEEE ICC*, June 2009.
- [13] J. Shin and D. Shah, Randomized Scheduling Algorithm for Queueing Networks, available online at <http://arxiv.org/abs/0908.3670v1>.
- [14] J. Ghaderi and R. Srikant, On the Design of Efficient CSMA Algorithms for Wireless Networks, *Technical Report*, available online at <http://www.illinois.edu/~jghaderi/csma-arxiv.pdf>