A Distributed CSMA Algorithm for Wireless Networks based on Ising Model

Yi Wang and Ye Xia
Department of Computer and Information Science and Engineering
University of Florida, Gainesville, FL 32611, USA
Email: {yiwan, yx1}@cise.ufl.edu

***Abstract*—Recent studies on queue-length-based randomized link scheduling algorithms have shown their throughput-optimality. But simulation results indicate that the packet delay in such algorithms can be quite large, even when the traffic intensity is low compared with the network capacity. The reason can be traced to the fact that these algorithms need the queues to be sufficiently large before the links have a good chance to be activated for transmission. In this paper, we propose a new randomized scheduling algorithm based on the Ising model in physics. The algorithm does not require queue build-up for link activation and can thus give better delay performance. It is easily implementable in a distributed fashion and it retains throughput-optimality.**

I. Introduction

Efficient utilization of the network resources is vitally important in wireless networks, as the capacity of such networks is often severely limited. Despite the capacity limitation, network users often demand high-bandwidth and low-delay network services for applications such as video or other types of streaming. Even with today's improved wireless network technologies, such demands are still not met. Link transmission scheduling is one of the key mechanisms for improvement in both network resource utilization and user perceived performance. An ideal link scheduling algorithm should achieve high throughput, low delay, and it should do so at low complexity.

The well-known max-weight scheduling algorithm [1] is throughput-optimal, in that it can stabilize the network queues for all arrival rate vectors in the interior of the capacity region. However, this algorithm involves solving an NP-hard combinatorial problem on each time slot. Thus, it is not practical for large wireless networks. Another group of algorithms uses schedules of lower complexity and achieves a fraction of the capacity region. Such algorithms include the longest-queue-first (LQF) schedule [7] [8], which has good delay performance at the expense of throughput reduction. LQF involves sorting all the link queues on each time slot, which requires global (i.e., network-wide) information and control and can be time-consuming to do. Another family of simple scheduling algorithms has also been studied intensively, the random access algorithms in which the link activation probabilities are dependent on the queue sizes [2] [3] [4] [6] [9]. They can be implemented similarly to the Carrier Sensing Multiply Access (CSMA) scheme used in practical systems such as WIFI 802.11x. The implementation is built on distributed algorithms, which require only local information and control. Some of these algorithms have been proven to be throughput-optimal [2] [3] [4].

Recent studies [5][13] suggest that it is impossible to design a scheduling algorithm for general wireless networks that is both throughput-optimal and has low delay. Our simulation experiments on a queue-length-based CSMA-like algorithm, called Q-CSMA [6], have revealed that it leads to fairly large delay, as the queues can become very large (see Section IV). Large queues can happen even when the incoming traffic intensity is low, i.e., the arrival rate vector is well within the network capacity region. The causes can be partially understood by observing the activation probabilities of the links. For each link v, the activation probability is $e^{W_v}/(1+e^{W_v})$, where the weight W_v is usually a slowly increasing function of link v's queue length Q_v, e.g. $W_v(Q_v) \propto \log(Q_v + 1)$. As a result, even if link v's neighboring links are all idle at the moment when v is selected for consideration of activation, link v is unlikely to be activated unless its packet queue is large enough. Thus, opportunities for transmissions are not sufficiently utilized by the links in the neighborhood until the queues become large.

There is a related queue-based CSMA-like algorithm [9], for which it has been shown that the queue dynamics is a fast-mixing Markov chain on a part of the capacity region, provided the degree of the interference graph is bounded. Fast mixing implies the queues rapidly approach stationarity, which appears to imply smaller queue sizes, and hence, lower delay. Our simulation experiments have shown that the queues are in fact very large. Moreover, the algorithm appears to be not throughput-optimal. In Q-CSMA, low packet delay may also be achieved by switching to greedy maximal scheduling when the traffic rates are low [6].

This paper reports a different CSMA-like algorithm, which shows improvement in delay over earlier algorithms. The algorithm is based on a physical model called the Ising model. It takes into account the ON/OFF status of the neighboring (interfering) links when calculating the activation probability for a link. The effect is that it "encourages" a link to activate if all the link's neighbors are OFF and "discourages" a link if one or more of its neighbors are ON. The algorithm can naturally eliminate unnecessary queue build-up that is present in other CSMA-like algorithms and it results in much smaller queue sizes and better delay performance. In addition, the scheduling algorithm is provably throughput-optimal. Simulation results have confirmed the improved delay performance.

The rest of the paper is organized as follows. In Section II, we describe the system model and notations. In Section III, we introduce the Ising model, present our distributed scheduling algorithm, and discuss the throughput-optimality and delay performance of the algorithm. In Section IV, we present the simulation results of both our algorithm and the Q-CSMA algorithm and compare their performance. We conclude the paper in Section V.

II. System Model and Notations

We consider a single-channel wireless network characterized by an undirected *interference graph*, $G = (V, E)$, where the vertex set V represents the wireless links and the edge set E indicates the interference relations between the links. Link u and v are connected by an edge $(u, v) \in E$ if and only if their transmissions interfere each other. The interfering links cannot be activated for transmission simultaneously. Thus, a *schedule* is an independent set of the graph G.

We assume the links all have identical capacity and the packets are all of an identical size. We consider a discrete-time system where the time slot size is equal to the transmission time of one packet. Thus, each link's capacity is one packet per time slot. At the beginning of each time slot, scheduling decisions are made whether each of the links will be activated for transmission on that time slot.

We represent a schedule by a vector $x \in \{-1, 1\}^{|V|}$. If a link l is included in the schedule, the l^{th} entry x_l is set to 1. We also say link l is ON or *activated*. Otherwise, $x_l = -1$ and link l is said to be *OFF*. A schedule always means a feasible schedule; that is, no two interfering links are both ON in a schedule. The set of all schedules is denoted by $\mathcal{M}$. An arbitrary vector in $\{-1, 1\}^{|V|}$ is called a *configuration*, which may or may not be feasible. A scheduling algorithm is a way to choose a schedule in each time slot.

Since the focus of the paper is on MAC-layer scheduling, we consider one-hop traffic, i.e., after transmitted by a link, a packet leaves the network. Packets arrive at the transmitters of the links according to a discrete-time finite-state Markov chain. The arrivals for different links are independent of each other. The system state at time slot t can be described by the queue sizes, the ON-OFF status of the links, and the number of arrivals at time t. The schedules considered in this paper depend only on the queue sizes and the ON-OFF status of the links. Hence, the system state is also a discrete-time Markov chain. System stability means that the Markov chain is positive recurrent.

The stability region is defined as the set of all arrival rate vectors for which there exists a scheduling algorithm that stabilizes the network queues. A scheduling algorithm is said to be *throughput-optimal* if it can stabilize the queues under any arrival rate vector in the stability region. The capacity region is the closure of the stability region [1].

III. Algorithm and Analysis

One drawback of the existing CSMA-like algorithms is that, when the queue size of a link is not large enough, the activation probability is very small. The link is unlikely to be activated even if no other links in the neighborhood are transmitting. Our algorithm is more mindful of the ON-OFF status of the neighboring links in the scheduling decision. For a link that has packets to transmit, it should be encouraged to transmit if no interfering links are transmitting. The inspiration of the algorithm is from a model in physics, called the *Ising model*. We will briefly introduce the Ising model first.

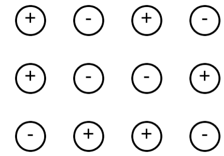

Fig. 1. Ising model and spin values

A. Ising Model

The Ising model [10][12] is a mathematical model of ferromagnetism. It uses spin variables with two possible values, $+1$ or -1, to represent magnetic dipole moments. The spin variables are described as the vertices of a graph, usually, a lattice, and neighboring spin variables can interact with each other. Given such a graph, a configuration σ is a vector that specifies all the spin values and $\sigma(v)$ is the spin value at vertex v (see Fig. 1 for an example).

In this model, the energy of a configuration σ is given by $H(\sigma) = -\sum_{v \sim w} \sigma(v)\sigma(w)$, where $v \sim w$ means v and w are neighbors in the graph. As one can see, the energy increases with the number of neighboring pairs whose spin values disagree. The *Gibbs distribution* corresponding to energy H is a probability distribution, denoted by μ, on the configuration space, Ω. Under parameter $\beta > 0$, it is given by $\mu(\sigma) = \frac{1}{Z(\beta)} e^{-\beta H(\sigma)}$, where $Z(\beta)$ is the normalizing constant. Clearly, $Z(\beta) = \sum_{\sigma \in \Omega} e^{-\beta H(\sigma)}$.

The Glauber dynamics on Ising model is a Markov chain of configurations with Ω as the state space, whose stationary distribution is the Gibbs distribution. Given the current configuration (i.e., state) σ, the Markov chain makes a transition to a new configuration σ' according to the following rule: First, pick a vertex v uniformly at random from the graph; and then, choose the spin value for v to be either $+1$ or -1 randomly according to the probabilities $q(+1; \sigma, v)$ or $q(-1; \sigma, v) = 1 - q(+1; \sigma, v)$, respectively, where

$$q(+1; \sigma, v) = \frac{e^{\beta S(\sigma, v)}}{e^{\beta S(\sigma, v)} + e^{-\beta S(\sigma, v)}}. \quad (1)$$

Here, $S(\sigma, v) = \sum_{w: w \sim v} \sigma(w)$. Note that the new configuration σ' may differ from the current configuration σ only at vertex v.

B. Modified Ising Model and Glauber Dynamics

The Gibbs distribution for the Ising model puts more probability masses on lower-energy configurations, in which the neighboring spins tend to agree in value. For the wireless link scheduling problem, each link can be in either ON or OFF state[1]. We will use the value 1 or -1 to represent an ON or OFF state, respectively. A link in the ON state is also said to be *activated*.

In a transmission schedule, which by definition is feasible, any pair of neighboring links (with respect to the interference graph) cannot be both in the ON state. In a maximal schedule, a link must be in the ON state if its neighbors are all OFF.

We will later propose a randomized scheduling algorithm, which generates a stationary distribution on the space of

[1] A link in the ON state may or may not be transmitting a packet. A link in the OFF state cannot transmit a packet.

schedules. Similar to several other scheduling algorithms of this family, we wish to have the probability mass to be concentrated on the max-weight schedules, where each link weight is some increasing function of the link's queue size. As in earlier algorithms, the concentration happens when the queue sizes are sufficiently large. However, for smaller queue sizes, the earlier algorithms do not even favor maximal schedules, and this behavior causes the queue sizes to grow even when the traffic intensity is light or moderate.

Since we prefer maximal schedules, we wish the probability mass to be also concentrated on the maximal schedules under all queue-size regimes. For that objective, we will consider the following modification to the Ising model. The underlying graph is the interference graph $G = (V, E)$ in which each vertex is a wireless link.

A *configuration* of the system is a $|V|$-dimensional vector that describes the ON-OFF status of all the links. The configuration space is $\Omega \triangleq \{-1, 1\}^{|V|}$. Note that the space of schedules $\mathcal{M}$ is a subset of Ω. Given a vector $\sigma \in \Omega$, $\sigma(v) = 1$ indicates link v is ON and $\sigma(v) = -1$ indicates link v is OFF. Under $\sigma \in \Omega$, we associate a *spin value* with each link v and denote it by $s_\sigma(v)$. For each link v, $s_\sigma(v)$ takes a value from the set $\{A_v, -1\}$, where $A_v > 0$. If link v is ON in σ, $s_\sigma(v) = A_v$; if it is OFF, $s_\sigma(v) = -1$. Note that, in the standard Ising model, $A_v = +1$ for all v. Hence, our modification is a generalization. In the eventual algorithm, each A_v is an increasing function of link v's queue size. For now, let us consider it fixed.

We define the energy of configuration $\sigma \in \Omega$ under the vector $A = (A_v)_{v \in V}$ as

$$H(\sigma, A) = -\sum_{(v,w) \in E} s_\sigma(v) s_\sigma(w). \tag{2}$$

Remark 1: When there is no ambiguity, we use the simplified notation $H(\sigma)$ instead.

Note that a neighboring pair, v and w, contributes the following to the total energy: (i) $-A_v A_w$ if they are both ON; (ii) A_v if v is ON and w is OFF; (iii) A_w if w is ON and v is OFF; and (iv) -1 when both are OFF. When A_v or A_w is large, an ON-ON pair (v, w), which corresponds to two interfering links, contributes a negative value with a large magnitude to the total energy. Hence, a high-energy configuration tends to have few such ON-ON pairs, but many ON-OFF pairs. When A_v is close to 1 for all v, OFF-OFF pairs are also discouraged in a high-energy configuration. For instance, in a highest-energy configuration, a link cannot be OFF when its neighbors are all OFF.

Glauber Dynamics: The proposed randomized scheduling algorithm, which will be described in Section III-C, has an embedded Glauber dynamics on the space Ω with a stationary distribution that puts more probability masses on higher-energy configurations[2]. In particular, the Glauber dynamics will have the following stationary probability distribution, μ,

$$\mu(\sigma) = \frac{1}{Z(\beta)} e^{\beta H(\sigma)}, \ \sigma \in \Omega, \tag{3}$$

[2]Note that this objective is the opposite to the Ising model, which puts more probability masses on lower-energy configurations.

where β is a positive parameter and the normalizing constant $Z(\beta)$ is given by $Z(\beta) = \sum_{\sigma \in \Omega} e^{\beta H(\sigma)}$.

Given that the Glauber dynamics is in configuration (i.e., state) $\sigma \in \Omega$, the next configuration can differ from σ in at most one entry. Let $\theta^+_{\sigma,v}$ be a configuration in Ω such that $\theta^+_{\sigma,v}(v) = 1$ and $\theta^+_{\sigma,v}(w) = \sigma(w)$ for any $w \neq v$. Let $\theta^-_{\sigma,v} \in \Omega$ be such that $\theta^-_{\sigma,v}(v) = -1$ and $\theta^-_{\sigma,v}(w) = \sigma(w)$ for any $w \neq v$.

For determining the next configuration, first, a link is chosen uniformly at random; second, given link v is chosen, it will be turned ON with probability $q(1; \sigma, v)$ and OFF with probability $q(-1; \sigma, v) = 1 - q(1; \sigma, v)$, where

$$q(1; \sigma, v) = \frac{\mu(\theta^+_{\sigma,v})}{\mu(\theta^+_{\sigma,v}) + \mu(\theta^-_{\sigma,v})} = \frac{e^{-A_v \beta S(\sigma,v)}}{e^{\beta S(\sigma,v)} + e^{-A_v \beta S(\sigma,v)}}. \tag{4}$$

$$= \frac{1}{2}(1 - \tanh(\frac{A_v + 1}{2} \beta S(\sigma, v))). \tag{5}$$

In the above, $S(\sigma, v) \triangleq \sum_{w:(v,w) \in E} s_\sigma(w)$, which is the sum of the spin values of link v's neighbors.

The quantity $q(1; \sigma, v)$ is called the *activation probability* for link v given that the system is in configuration σ and that link v is selected for consideration. It only depends on the spin values of link v's neighboring links. Such a property of locality makes the scheduling protocol simple, since only local information needs to be collected.

C. Proposed Distributed Scheduling Algorithm

In the proposed scheduling algorithm, the spin value of an ON link is an increasing function of the link's queue size and hence will vary with time. Specifically, for each ON link v at time t, its spin value is $A_v(t) = 2(\bar{d} - 1) + \log(Q_v(t) + 1)$, where $Q_v(t)$ is link v's queue size at time t and $\bar{d}$ is the maximum vertex degree in the graph G.

The proposed algorithm has a Glauber dynamics in the background, which is the one described in Section III-B with the modification that the spin values are time-varying. When the state of the Glauber dynamics at time t is not feasible as a schedule, the algorithm converts the state into a valid schedule by turning OFF some links. Specifically, let $\{\sigma(t)\}_{t \geq 1}$ be the sequence of states of the Glauber dynamics; there is a sequence of schedules $\{\sigma'(t)\}_{t \geq 1}$, which are feasible by definition, generated by the algorithm. If $\sigma(t)$ is feasible, then $\sigma'(t) = \sigma(t)$. Otherwise, $\sigma'(t)$ is a restriction of $\sigma(t)$ in the sense that $\sigma'(t)(v) \leq \sigma(t)(v)$ for all $v \in V$; that is, if link v is OFF in $\sigma(t)$, then it must be OFF in the schedule $\sigma'(t)$.

A time slot is divided into a control slot and a data slot. For efficiency, the data slot size should be much larger than the control slot size. The control slot is further divided into $W + W'$ mini-slots where W and W' are chosen constant integers. The first W mini-slots form *control phase I* and the goal is to collectively set the vector $\sigma(t)$. The second W' mini-slots form *control phase II* and the goal is to collectively set the vector $\sigma'(t)$. During control phase I, the links that attempt to change their entry values in $\sigma(t)$ must correspond to an independent set, denoted by $\xi(t)$, in the interference graph; this is accomplished using the INTENT messages. During control phase II, only ON links will compete for the channel and RESERVE messages are used for the competition.

Each RESERVE message also contains the current queue size information of the link.

On each time slot t, each link v with a non-empty queue runs the following steps.

Distributed Scheduling Algorithm (at Link v)

Initialization:

1. At the beginning of the time slot, link v calculates $S(\sigma(t-1), v)$ based on the neighboring links' ON-OFF status (in $\sigma(t-1)$) and queue sizes that it learned during the previous time slot. Link v calculates the probability $q(1;\sigma(t-1), v)$ based on the expressions in (4).

Control Phase I - W Mini-Slots: Set $\sigma(t)(v)$

2. Link v selects a random back-off time T_1 uniformly in $\{0, 1, \ldots, W-1\}$ and sets a timer of T_1 control mini-slots.
3. If link v hears an INTENT message from any of its neighboring links before the T_1 timer expires, it sets $\sigma(t)(v) = \sigma(t-1)(v)$ and it will not transmit an INTENT message (v will not be included in $\xi(t)$).
4. Otherwise, when the T_1 timer expires, link v broadcasts an INTENT message at the beginning of the $(T_1 + 1)$-th mini-slot.
   a) If link v's INTENT message has a collision[3], link v sets $\sigma(t)(v) = \sigma(t-1)(v)$ (v is not included in $\xi(t)$).
   b) Otherwise, link v sets $\sigma(t)(v) = 1$ (chooses ON) with probability $q(1;\sigma, v)$ and it sets $\sigma(t)(v) = -1$ (chooses OFF) with probability $q(-1;\sigma, v)$.

Control Phase II - W' Mini-Slots: Set $\sigma'(t)(v)$

5. If link v has $\sigma(t)(v) = 1$, it executes the following:
   a) Link v selects a random back-off time T_2 uniformly in $\{0, 1, \ldots, W'-1\}$ and sets a timer of T_2 control mini-slots. It sets $\sigma'(t)(v) = 0$.
   b) When the T_2 timer expires, v broadcasts a RESERVE message containing its current queue size.
   c) If link v has not heard any RESERVE messages from its neighboring links before the timer expiration and if its own RESERVE message does not have a collision, link v sets $\sigma'(t)(v) = 1$.

Data Slot:

6. If $\sigma'(t)(v) = 1$, link v transmits a packet.

Remark 2: The information used to compute $S(\sigma(t-1), v)$ in step 1 is obtained from the broadcast of the RESERVE messages in the previous time slot (see step 5b)). A link hears the RESERVE messages with queue sizes from those neighbors that are ON in the previous time slot (according to $\sigma(t-1)$), but hears nothing from the OFF links. This does not pose a problem for computing $S(\sigma(t-1), v)$, since the computation only requires the queue size information of the ON neighbors and the total *number* of OFF neighbors.

Control phase II is for conflict resolution. At the end of the phase, a valid schedule $\sigma'(t)$ is produced based on the state of the Glauber Dynamics $\sigma(t)$. The objective is to allow at most one link to transmit in each link's neighborhood, even if multiple links may be ON (according to $\sigma(t)$) in that neighborhood. It is important to note that ON-ON neighboring pairs are usually few by the design of the algorithm. Furthermore, the use of multiple (W') mini-slots and randomizing timers further reduce the chance of collisions of RESERVE messages from ON-ON neighbors. Hence, collisions among RESERVE messages will be rare. If a collision still occurs on a mini-slot, all links that sent the colliding messages will keep $\sigma'(t)(v)$ at 0. In the event that some required control information, such as the current queue length, is not updated due to collisions, old information from an earlier time slot can be used without affecting the proper functioning of the algorithm; the impact on efficiency will also be negligible.

The control overhead can be made small. Each INTENT or RESERVE message is relatively small compared with a data packet, if the duration of a data slot is extended sufficiently long. Furthermore, the number of control messages is limited by our design. The parameter W' can be small because of the rare occurrence of the ON-ON neighboring pairs in σ. In our simulation, $W' = 4$ is usually sufficient.

Effect of β: As can be observed from (3), as β increases, the distribution μ is increasingly biased towards higher-energy states, which tend to have more ON-OFF neighboring link pairs and fewer ON-ON or OFF-OFF pairs. As β decreases, the probabilities for configurations of different energy levels become more and more equalized. Expression (5) suggests that a larger β value leads to a higher activation probability for link v, if its neighbors are OFF (thus giving a negative $S(\sigma, v)$). Hence, a larger β tends to result in more aggressive link activations, and consequently, smaller queue sizes. However, our study has shown that a larger β may increase the mixing time, i.e., the time taken for the Glauber dynamics to reach stationarity. Thus, choosing a relatively small β may be essential in having and proving the fast-mixing property when we do not assume time-scale separation. We will explore these issues in future work.

Activation Probability and Performance: As mentioned in the introduction section, the delay performance of the queue-based CSMA algorithm is non-ideal in part because the queue of a link has to build up sufficiently in order for the link to have a substantial activation probability. Our algorithm naturally avoids this unfavorable situation. From (5), we see that the activation probability is a hyperbolic tangent function with scaling and horizontal translation. When all the neighbors of link v are OFF (the factor $S(\sigma, v)$ is negative), the activation probability is at least 0.5 even for very small A_v (and hence, very small queue size). More concretely, suppose link v has two neighbors and they are both OFF, which results in $S(\sigma, v) = -2$. Suppose $\beta = 0.1$ and $A_v = 2(\bar{d} - 1) + \log(Q_v + 1)$. Suppose the maximum vertex degree of the interference graph G is $\bar{d} = 3$, and consequently, $A_v \geq 4$. The activation probability is equal to 0.73, 0.80, 0.86, 0.90 for $A_v = 4$, 6, 8, 10, which corresponds to $Q_v = 0$, 7.4, 54.6, 403.4, respectively. Hence, the activation probability starts at a relatively high value and can increase rapidly towards 1 as the queue size increases. In short, our

[3] The receiver of a link transmits an acknowledgement in response to a INTENT or RESERVE message from the sender of the link. A collision is detected by the absent of an acknowledgement.

algorithm reacts faster to the queue build-up and has lower delay than the queue-based CSMA, especially when the queue sizes are small.

On the other hand, if one or more neighbors of link v are ON, resulting in a positive $S(\sigma, v)$, the activation probability can be very close to 0 for even a small queue size at link v. Hence, ON-ON neighboring pairs are strongly discouraged.

D. Throughput Optimality

For the proposed scheduling algorithm, we can establish its throughput-optimality. In this section, we outline the proof of throughput-optimality under the time-scale separation assumption, i.e., the Glauber dynamics is in the steady state in every time slot. We refer the reader to [16] for more details about the proof. A proof without the time-scale separation assumption is left to future work.

Therorem 1: The proposed randomized scheduling algorithm is throughput-optimal.

We first need some definitions. Given the interference graph $G = (V, E)$, let d_v denote the degree of vertex v in G and let $\bar{d}$ be the maximum vertex degree, i.e., $\bar{d} = \max_{v \in V} d_v$. Let A be the vector $(A_v)_{v \in V}$. Let the weight of vertex v (or link v) be $W_v(A) = d_v A_v$. For a configuration $\sigma \in \Omega$, define the weight of σ under A to be the total weight of all the vertices that are ON in σ, i.e.,

$$W(\sigma, A) = \sum_{v \in V : \sigma(v)=1} W_v(A) = \sum_{v \in V : \sigma(v)=1} d_v A_v. \quad (6)$$

If $\sigma \in \mathcal{M}$, i.e., σ is a valid schedule, then $W(\sigma, A)$ is the schedule weight.

Let $W^*(A)$ be the maximum schedule weight under A, i.e., $W^*(A) = \max_{\sigma \in \mathcal{M}} W(\sigma, A)$. Let $H^*(A)$ be the maximum energy under A, where the maximization is taken over all possible configurations, i.e., $H^*(A) = \max_{\sigma \in \Omega} H(\sigma, A)$.

Remark 3: When there is no ambiguity, we use the simplified notations $W(\sigma)$, W_v, W^*, and H^* instead.

To prove Theorem 1, we will use a theorem from [11]. Consider a finite family of non-decreasing functions f_v on $\mathbb{R}_+$, where $v \in V$, with the property that $\lim_{q \to \infty} f_v(q) = \infty$ for each v. In addition, suppose the following holds for each v: For any $M_1 > 0$, $M_2 > 0$ and $0 < \epsilon < 1$, there exists $\bar{Q} < \infty$ such that for all $q > \bar{Q}$,

$$(1-\epsilon) f_v(q) \leq f_v(q - M_1) \leq f_v(q + M_2) \leq (1+\epsilon) f_v(q). \quad (7)$$

Given such a family of function $(f_v)_{v \in V}$, let $A_v(t) = f_v(Q_v(t))/d_v$, where $Q_v(t)$ is the queue size of link v at time t. Hence, the weight of each link $v \in V$ at time t is $d_v A_v(t) = f_v(Q_v(t))$. Let $A(t) = (A_v(t))_{v \in V}$. Let $||Q(t)|| = \sqrt{\sum_{v \in V} Q_v^2(t)}$. The following theorem is proved in [11].

Therorem 2: Consider a scheduling algorithm. Suppose for any ϵ and δ, $0 < \epsilon, \delta < 1$, there exists a $B > 0$ such that in any time slot t, with probability greater than $1 - \delta$, the scheduling algorithm chooses a schedule $\sigma(t) \in \mathcal{M}$ satisfying the following: Whenever $||Q(t)|| > B$,

$$W(\sigma(t), A(t)) \geq (1 - \epsilon) W^*(A(t)). \quad (8)$$

Then, the scheduling algorithm is throughput-optimal.

We need some supporting lemmas.

Lemma 3: Suppose $A_v \geq 2(\bar{d} - 1)$ for all $v \in V$. For a given $\sigma \in \Omega$, suppose $H(\sigma) \geq (1 - \epsilon) H^*$, where $0 < \epsilon < 1/(1 + |E|)$. Then, $W(\sigma') \geq (1 - \epsilon(1 + |E|)) W^* - 2|E| - |E|^2$.

We now return to the analysis of the proposed scheduling algorithm. In our case, $A_v(t) = 2(\bar{d} - 1) + \log(Q_v(t) + 1)$ at time t and the link weight at time t is $d_v A_v(t)$. Hence, the function $f_v(q)$ is given by $f_v(q) = d_v(2(\bar{d} - 1) + \log(q + 1))$, which satisfies (7). When link v is ON at time t, its spin value is $A_v(t)$. The total weight of any configuration $\sigma \in \Omega$ is given by (6) with $A(t) = (A_v(t))_{v \in V}$ replacing A. Now, we only need to check the conditions of Theorem 2 for our algorithm.

Under a given vector A, let

$$\mathcal{X}(A, \epsilon) = \{\sigma \in \Omega : H(\sigma, A) < (1 - \epsilon) H^*(A)\}, \quad (9)$$

where $0 < \epsilon < 1$.

Lemma 4: For any δ, where $0 < \delta < 1$, there exists $B(\epsilon, \delta) > 0$ such that when $||Q|| > B(\epsilon, \delta)$, $\mu(\mathcal{X}(A, \epsilon)) < \delta$.

Remark 4: For fixed ϵ and δ, each particular Q satisfying $||Q|| > B(\epsilon, \delta)$ determines A, which in turn determines $\mathcal{X}(A, \epsilon)$, $H^*(A)$, $W^*(A)$, etc. The distribution μ is a conditional distribution given that A is known.

Lemma 4 says that when $||Q||$ is large enough, the stationary distribution[4] of the Glauber dynamics concentrates on the set $\mathcal{X}^c = \Omega \backslash \mathcal{X}$, i.e., $\mu(\mathcal{X}^c) \geq 1 - \delta$. Each element $\sigma \in \mathcal{X}^c$ has nearly the maximum energy.

Suppose the state of the Glauber dynamics at time t is $\sigma(t) \in \Omega$, which may or may not be a valid schedule (i.e., feasible). If $\sigma(t)$ is not feasible, the proposed scheduling algorithm converts it into a valid schedule in $\mathcal{M}$ by turning OFF some of the links. We consider the most aggressive version of such a conversion scheme in which all the ON links with ON neighbors in $\sigma(t)$ are turned OFF. The resulting valid schedule is denoted by $\sigma'(t)$. More precisely, the conversion scheme is characterized by a mapping, $\phi : \sigma \mapsto \sigma'$, defined as follows. Given σ, let us define a subset $F \subseteq V$: A vertex v is in F if and only if v is ON and v has at least one ON neighbor in σ. In other words, all ON-ON neighboring pairs of vertices are in F. Let $F^c = V \backslash F$. Then, $\sigma' = \phi(\sigma)$ is given by

$$\sigma'(v) = \begin{cases} \sigma(v) & \text{for } v \in F^c, \\ -1 & \text{for } v \in F. \end{cases} \quad (10)$$

Note that σ' is a valid schedule, i.e., $\sigma' \in \mathcal{M}$.

The key is to show that if σ has near maximum energy, then σ' has near maximum weight.

Proof: (of Theorem 1) Consider the ϵ and δ required by Theorem 2. For the ϵ in Lemma 4, we replace it with any ϵ_1 satisfying $0 < \epsilon_1 < \frac{\epsilon}{2(|E|+1)}$. Then, Lemma 4 says that there exists B_1, which depends on ϵ_1 and δ, such that when $||Q(t)|| > B_1$, $\mu(\mathcal{X}(A(t), \epsilon_1)) < \delta$.

Let $\mathcal{Y}(A(t), \epsilon_1) = \{\phi(\sigma) | \sigma \in \mathcal{X}^c(A(t), \epsilon_1)\}$. Then, when $||Q(t)|| > B_1$, $P(\sigma'(t) \in \mathcal{Y}(A(t), \epsilon_1)) = \mu(\mathcal{X}^c(A(t), \epsilon_1)) \geq 1 - \delta$. We only need to show that $W(\sigma'(t), A(t)) \geq (1 - \epsilon) W^*(A(t))$ for $\sigma'(t) \in \mathcal{Y}(A(t), \epsilon_1)$. For that purpose, we

[4] This is the stationary distribution conditional on holding the queue sizes Q, and hence, A unchanged.

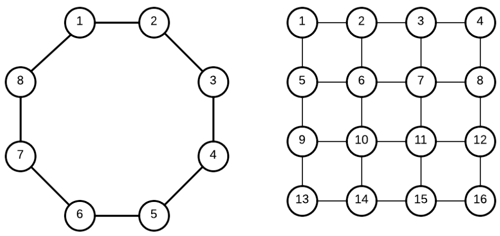

Fig. 2. Interference graphs: 8-link ring (8-cycle) and 16-link grid. The vertices correspond to wireless links in the network.

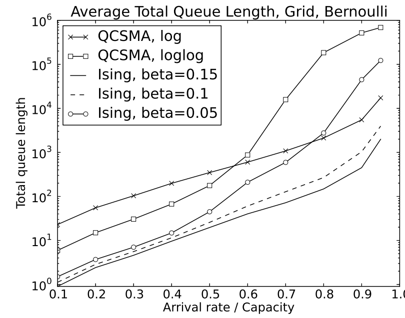

Fig. 3. 16-link grid interference graph, Bernoulli arrival

apply Lemma 3 with ϵ replaced by ϵ_1.

$$\begin{aligned} & W(\sigma'(t), A(t)) \\ \geq & (1-\epsilon_1(1+|E|))W^*(A(t)) - 2|E| - |E|^2 \\ > & (1-\epsilon/2)W^*(A(t)) - 2|E| - |E|^2 \\ = & (1-\epsilon)W^*(A(t)) + \epsilon W^*(A(t))/2 - 2|E| - |E|^2 \\ \geq & (1-\epsilon)W^*(A(t)), \text{ when } W^*(A(t)) \geq (4|E|+2|E|^2)/\epsilon \end{aligned}$$

It is easy to see that there exists some $B_2 > 0$ such that when $||Q(t)|| > B_2$, $W^*(A(t)) \geq (4|E|+2|E|^2)/\epsilon$. Finally, we can choose $B = \max\{B_1, B_2\}$. For each $Q(t)$ such that $||Q(t)|| > B$, which determines $A(t)$, we have $P(\sigma'(t) \in \mathcal{Y}(A(t), \epsilon_1)) \geq 1-\delta$ and $W(\sigma'(t), A(t)) > (1-\epsilon)W^*(A(t))$.

Let us now return to the actual schedule, $\sigma''(t)$, derived from $\sigma(t)$ by the proposed algorithm in Section III-C. Let $\psi : \sigma(t) \mapsto \sigma''(t)$ represent the conversion from $\sigma(t)$ to $\sigma''(t)$ by the proposed algorithm, and let $\mathcal{Z}(A(t), \epsilon_1) = \{\psi(\sigma) | \sigma \in \mathcal{X}^c(A(t), \epsilon_1)\}$. Then, $P(\sigma''(t) \in \mathcal{Z}(A(t), \epsilon_1)) = \mu(\mathcal{X}^c(A(t), \epsilon_1)) \geq 1-\delta$.

Next, for every $\sigma''(t) \in \mathcal{Z}(A(t), \epsilon_1)$, we can find $\sigma(t) \in \mathcal{X}^c(A(t), \epsilon_1)$ such that $\sigma''(t) = \psi(\sigma(t))$; with such $\sigma(t)$, let $\sigma'(t) = \phi(\sigma(t))$. Then, $\sigma'(t) \in \mathcal{Y}(A(t), \epsilon_1)$. Being both converted from the same $\sigma(t)$, the actual schedule $\sigma''(t)$ has at least as much weight as $\sigma'(t)$, according to the rules of the two conversion schemes. We have just shown that $W(\sigma''(t), A(t)) > (1-\epsilon)W^*(A(t))$ for all $\sigma''(t) \in \mathcal{Z}(A(t), \epsilon_1)$.

Hence, the conditions of Theorem 2 are satisfied by the proposed algorithm, which then must be throughput-optimal. ■

IV. Simulation Results

In this section, we evaluate the performance of the proposed scheduling algorithm by simulation and compare it with the Q-CSMA algorithm [6]. Our focus is on the average queue length, which, by Little's law, is directly related to the average delay experienced by the packets.

We used two interference graphs to evaluate the algorithms in different situations. One is a 4×4 grid with 16 vertices; the other is a ring with 8 vertices, also known as an 8-cycle (see Fig. 2). The vertices represent wireless links and the edges represent the interference relations between the links.

For the first part of the simulation, we assumed that the incoming traffic to each link v follows a Bernoulli process with rate λ_v. The arrival processes for different links are independent. For each set of traffic rates, we ran the simulation 10 times and took the average. Each simulation run lasted for 10^6 time slots, and we recorded the sum of the final queue sizes of all the links as the performance measure.

We used two weight functions for Q-CSMA: $W_v(t) = \log(0.1 * Q_v(t) + 1)$ and $W_v(t) = \log\log(Q_v(t) + e)$, and the contention window is set to $W = 32$. For our distributed algorithm, we tried three different β values 0.15, 0.1, and 0.05 to evaluate the impact of β on performance, and the selected window sizes are $W = 32$ and $W' = 4$.

a) Grid and Bernoulli Arrival: For the experiments on the 16-link grid, we used the following scheme to sample the arrival rate vectors. Consider two sets of links:

$$L_1 = \{1, 3, 6, 8, 9, 11, 14, 16\}, L_2 = \{2, 4, 5, 7, 10, 12, 13, 15\}.$$

Each set is a maximal schedule for the network. Let e^1 and e^2 each be a 16-dimensional vector representation of L_1 and L_2: For each $i \in \{1, 2\}$, set $e^i_v = 1$ if link $v \in L_i$; otherwise, set $e^i_v = 0$. We chose arrival rate vectors that can be represented as conical combinations of e^1 and e^2.

$$\lambda = \rho \sum_{i=1}^{2} t_i e^i, \quad \text{with } t_1 + t_2 = 1. \tag{11}$$

Since $\sum_{i=1}^{2} t_i e^i$ lies on the boundary of the capacity region, the parameter ρ, where $\rho \geq 0$, is a measure of the traffic intensity or load.

Fig. 3 shows the simulation results for the 16-link grid. We used $(t_1, t_2) = (0.5, 0.5)$, $(0.6, 0.4)$, $(0.7, 0.3)$ for the experiments and they gave very similar results. Each of the curves in Fig. 3 is the average of the three sets of experiments corresponding to different (t_1, t_2) values. Note that the y-axis is in log scale. We can observe the following:

- Our algorithm leads to smaller queues than Q-CSMA throughout the entire feasible region of the traffic intensity, $[0, 1)$. At low to medium traffic intensity, our algorithm has close to zero queue size, while Q-CSMA generally has $10^2 - 10^3$ total packets in the network.
- Both algorithms achieve stability for the entire feasible region of the traffic intensity. This is a strong evidence for throughput-optimality.
- β affects the performance of our algorithm. As predicted, a larger β generally leads to a smaller queue size due to more aggressive link activation.
- With a well-chosen β, the improvement by our algorithm is up to two orders of magnitude in these cases.

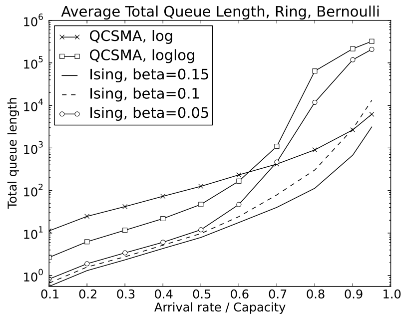

Fig. 4. 8-link ring interference graph, Bernoulli arrival

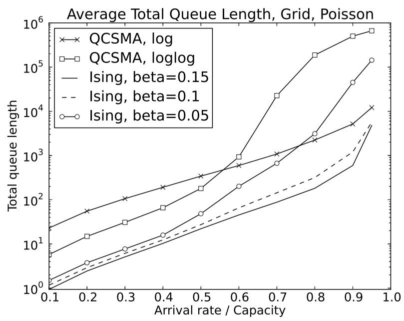

Fig. 5. 16-link grid interference graph, Poisson arrival

b) Ring and Bernoulli Arrival: For the 8-cycle, we chose the following two maximal schedules: $L_1 = \{1, 3, 5, 7\}$, $L_2 = \{2, 4, 6, 8\}$. We let $e^1 = (1, 0, 1, 0)'$ and $e^2 = (0, 1, 0, 1)'$. The arrival rate vector was set as in (11). The simulation results are shown in Fig. 4. We observe similar results as for the grid case. Our algorithm can lead to much smaller queue sizes, especially under low to medium traffic intensity. The improvement is not as large as it is in the grid topology, perhaps because there are fewer links and less interference in the 8-cycle.

c) Other Traffic Models: We have tested the algorithms under other traffic models. One is the I.I.D. process with a Poisson distribution, and the other is on-off bursty traffic. The traditional Bernoulli and Poisson traffic models give smooth traffic over a large time interval, and these types of traffic may be less representative than bursty traffic or self-similar traffic for local area networks [14][15]. For the on-off model, we use Pareto distributions for the on and off durations. The traffic for each link is still generated according to a Bernoulli process. During the off period, the generated packets are stored in a buffer at the source instead of being released into the network. During the on period, the generated and previously stored packets are injected into the network. The simulation results for the grid topology under the two traffic models are shown in Fig. 5 and Fig. 6. We see that even under these very different traffic patterns, the results show similar trends in the queue sizes as in the previous cases, with our algorithm out-performing Q-CSMA.

V. Conclusions

In this paper, we propose a randomized link scheduling algorithm based on a modified Ising model and the associated Glauber dynamics. The algorithm gives better delay performance than other queue-based CSMA-like algorithms. It is throughput-optimal and easily implementable. We validated the performance of the proposed algorithm and compared it with Q-CSMA by simulation. The proof for throughput-optimality relies on the time-scale separation assumption. In the future work, we will attempt to remove that assumption. We may also further explore the effect of β and whether or when the algorithm leads to fast-mixing Markov chains.

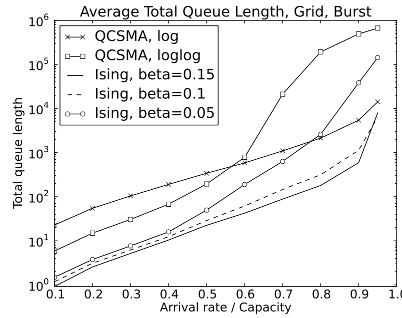

Fig. 6. 16-link grid interference graph, on-off bursty traffic

References

[1] L. Tassiulas and A. Ephremides. “Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multi-hop radio networks”, *IEEE Trans. Automatic Control*, Dec. 1992.

[2] L. Jiang and J. Walrand. “A distributed CSMA algorithm for throughput and utility maximization in wireless networks”, in *Proc. 46th Annual Allerton Conf. on Communication, Control and Computing*, Sept. 2008

[3] S. Rajagopalan, D. Shah, J. Shin. “Network adiabatic theorem: an efficient randomized protocol for contention resolution”, in *Proc. of ACM Sigmetrics*, June 2009.

[4] L. Jiang and J. Walrand. “Convergence and stability of a distributed CSMA algorithm for maximal network throughput”, in *IEEE Conference on Decision and Control*, Dec. 2009.

[5] D. Shah, D.N.C. Tse, and J.N. Tsitsiklis. “Hardness of low delay network scheduling”, in *IEEE Trans. on Information Theory*, Dec. 2011.

[6] J. Ni and R. Srikant. “Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks”, in *Proc. of Info. Theory and App. Workshop*, Jan. 2009.

[7] G. Sharma, N.B. Shroff, and R.R. Mazumdar, “Joint congestion control and distributed scheduling for throughput guaranttee in wireless networks”, in *Proc. IEEE INFOCOM*, 2007.

[8] B. Li, C. Boyaci, and Y. Xia. “A refined performance characterization of longest-queue-first policy in wireless networks”, in *Proc. ACM MobiHoc*, pp. 65-74, May 2009.

[9] L. Jiang, M. Leconte, J. Ni, R. Srikant and J. Walrand, “Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling”, in *Proc. IEEE INFOCOM*, 2011.

[10] David A. Levin, Yuval Peres and E.L. Wilmer, *Markov Chains and Mixing Times*, 2009

[11] A. Eryilmaz, R. Srikant and J.R. Perkins. “Stable Scheduling Policies for Fading Wireless Channels”, in *IEEE/ACM Transaction on Networking*, 13(2):411-424, April 2005.

[12] K. Binder, “Ising model”, in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, 2001

[13] C. Boyaci, B. Li, Y. Xia, “An investigation on the nature of wireless scheduling”, in *Proc. INFOCOM 2010*

[14] W.E. Leland, M.S. Taqqu, W. Willinger, “On the self-similar nature of Ethernet traffic”, in *IEEE/ACM Transactions on Networking*, Feb. 1994.

[15] J. Shin, T.M. Pinkston, “The performance of routing algorithms under bursty traffic loads”, in *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications*, Jun. 2003.

[16] Yi Wang and Ye Xia “A distributed CSMA algorithm for wireless networks based on Ising model” (long version), https://www.cise.ufl.edu/~yx1/publication.html.