I-CSMA: A Link Scheduling 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*—We propose a new randomized link scheduling algorithm for wireless networks, called I-CSMA, which is based on a modified version of the Ising model in physics. The main result is that I-CSMA is shown to be throughput-optimal. I-CSMA is a generalization of earlier Glauber-dynamics-based, throughput-optimal algorithms such as Q-CSMA in that each earlier algorithm involves a truncated Markov chain of the Markov chain in I-CSMA. The main result implies that there is no need to truncate the Markov chain in the classical way by prohibiting transitions into the subset of the excluded states, which in this case are the link configurations with interference. The Markov chain can freely move over the entire space of link configurations. Other methods of "truncation" for producing a valid schedule work equally well in terms of achieving throughput-optimality. Therefore, the main result justifies the removal of a major restriction in earlier related algorithms, and allows the exploration and exploitation of that new freedom. Our simulation experiments show that I-CSMA leads to smaller queue sizes, and therefore, gives better delay performance. The I-CSMA algorithm is still decentralized and easily implementable. We also propose a heuristic I-CSMA algorithm, which is even simpler and yet has comparable performance to I-CSMA.**

***Index Terms*—link scheduling, wireless networks, carrier-sense multiple access (CSMA), Markov chain, Ising model, Glauber dynamics**

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 realtime video streaming. 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. This group includes various maximal schedules, and in particular, the longest-queue-first (LQF) schedule [2] [3] [4] [5]. Experiments have shown that LQF has good delay performance; but this is achieved 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 scheduling algorithms has attracted much attention recently: randomized algorithms in which the link activation probabilities are dependent on the queue sizes [6] [7] [8] [9] [10] [11]. A representative one is the Q-CSMA algorithm [9]. These algorithms can be implemented similarly to the Carrier Sense Multiply Access (CSMA) scheme used in practical systems such as WIFI 802.11x. The implementation is decentralized and requires only local information and control. Interestingly, despite having simple operations, some of these algorithms are proven to be throughput-optimal. They are closely related to the proposed algorithm in this paper in that, behind the scene, they each have a Glauber dynamics, which is a special type of Markov chain.

This paper presents a new CSMA-like randomized algorithm, called I-CSMA, based on a physics model called the Ising model, which is a special type of Glauber dynamics. I-CSMA is substantially different from earlier Glauber-dynamics-based algorithms. While the Markov chain in each earlier algorithm is in the space of valid (i.e., interference-free) schedules, there is no such restriction for the Markov-chain in I-CSMA. In other words, the state space for I-CSMA is the set of all link configurations, where a configuration specifies whether each link is activated or not. When interference exists, a configuration is not a valid schedule. For actual transmission, there is a second step for conflict resolution, which converts a configuration into a valid schedule by turning off some of the interfering links. The main result of the paper is that I-CSMA is throughput-optimal, regardless which interfering links are turned off in the conflict resolution step.

Thus, I-CSMA is a generalization of earlier Glauber-dynamics-based, throughput-optimal algorithms such as Q-CSMA in that each earlier algorithm involves a truncated Markov chain of the Markov chain in I-CSMA. The main result of the paper says, there is no need to truncate the Markov chain in the classical way by prohibiting transitions into the subset of the excluded states, which in this case are the configurations with interference. The Markov chain can freely move over the entire state space of link configurations. Other methods of "truncation" for producing a valid schedule work equally well in terms of achieving throughput-optimality. Therefore, the main result of the paper justifies the removal of a major restriction in earlier related algorithms, and allows the exploration and exploitation of that new freedom.

One of the main benefits of I-CSMA that we have discovered is that it results in significantly smaller queue sizes and less delay than the closest related algorithm, Q-CSMA. Such improvement has been reliably demonstrated by our extensive simulation experiments on different networks under different traffic models. The improvement is generally over the entire range of traffic intensity; it is especially consistent in the regime of low to moderately high traffic intensity. I-CSMA works especially well, with respect to the queue-size/delay performance, under weight functions that increase very slowly with the queue size, e.g., $\log\log$ function of the queue size. Although it is difficult to give a precise analysis, the queue-size/delay improvement may have to do with the aforementioned main feature that distinguishes I-CSMA from earlier algorithms. When some neighbors of link v are activated, in earlier algorithms, link v has to passively wait for the activated neighbors to relinquish the channel, which may take a while. In I-CSMA, link v can directly compete for the channel against the activated neighbors. This feature reduces the chance that activated links monopolize the transmission opportunities for an excessive amount of time and it leads to less variability in the service processes of all the queues. As classical results on single-server queues suggest, less variability usually means a smaller queue size and less delay (see also [11] for a more rigorous argument on a related scenario).

I-CSMA has low computational complexity and is decentralized. Although it is slightly more complex than Q-CSMA, I-CSMA is still easily implementable, requiring only carrier sensing and local communications by the network devices[1]. We also propose a heuristic I-CSMA algorithm, which is yet easier to implement than I-CSMA. Simulation experiments have shown that the heuristic algorithm performs as well as I-CSMA. Finally, I-CSMA is not based on the standard Ising model, but on a generalized Ising model that we created. The generalized Ising model may have other independent applications and theoretical implications.

With respect to improving the queue-size/delay performance, we briefly review several existing studies. A related queue-based randomized algorithm is proposed in [10]. The authors show that, on a part of the capacity region, the queue dynamics is a fast-mixing Markov chain and the total queue size is bounded by a polynomial in the network size, provided the degrees of the interference graphs are bounded by a constant. It is unclear whether the algorithm is throughput-optimal. The authors of [12] consider the same class of interference graphs with a bounded degree and show that, on a part of the capacity region, the expected queue sizes under Q-CSMA-like algorithms are uniformly bounded by a constant independent of the network size. The authors of [13] introduce virtual channels and propose an CSMA-like randomized algorithm that ends up using different schedules on the different virtual channels. The objective is to avoid starvation of some links caused by active links hogging the transmission opportunity for too long. The algorithm leads to small head-of-line (HOL) delay at the wireless links. However, the model in [13] is about a closed-loop utility optimization problem with flow control and the delay is eventually pushed back to the sources. The result of low HOL delay doesn't appear to apply to an open-loop stability problem. In addition, the HOL delay is not the same as the delay experienced by a typical packet, which is what is considered in this paper. [14] [15] [16] contain interesting results about delay performance of CSMA-like algorithms but on more specialized networks.

A hybrid version of Q-CSMA is introduced together with Q-CSMA in [9] with the goal of reducing delay. In the hybrid version, when their queue weights are below a threshold, the links compete for activation in a way that emulates the greedy maximal schedule (i.e., LQF); when their queue weights exceed the threshold, the links switch to using the Q-CSMA algorithm. The same idea of switching between two algorithms can be applied to I-CSMA although we haven't pursued that opportunity yet. The authors of [11] generalize Q-CSMA in another direction. Specifically, the activation probability is parametrized by a constant β differently from the usual approach, where $\beta \in [0, 1]$. Doing so yields a family of Q-CSMA-like algorithms, which are all throughput-optimal. Through an approximate analysis, they show that each expected queue size decreases as β increases. That direction of generalization is orthogonal to the direction taken in this paper. We believe that the two can be combined into an even more general version. Some recent research challenges the existence of simple scheduling algorithms that can achieve throughput-optimality and small queue sizes simultaneously [17] [18]. Those results are about transient queue sizes. In principle, if the traffic is truly stationary, then it is possible to have a simple scheduling algorithm that results in small stationary queue sizes. For instance, we can measure the average arrival rates and find another rate vector μ in the capacity region that strictly dominates the arrival rate vector. By Carathéodory's theorem, we can express μ as a convex combination of K schedules, where K is equal to the dimension of the rate vector plus 1. The coefficients of the convex combination can be computed. After that initial stage of computation, the scheduling algorithm is to randomly select and apply one of the K schedules on each time slot with the coefficients as the probabilities for selection.

The rest of the paper is organized as follows. In Section II, we introduce the system model and notations, and review the standard Ising model and Glauber dynamics. In Section III, we present a modified Ising model and our I-CSMA algorithm. In Section IV, we prove the I-CSMA algorithm is throughput-optimal. In Section V, we discuss additional issues of I-CSMA such as control overhead and parameter tuning. In Section VI, we introduce a heuristic I-CSMA algorithm, which is a simplification of the I-CSMA algorithm. In Section VII, we present simulation results for both I-CSMA and Q-CSMA and compare their performance. Conclusions and a discussion of future work are given in Section VIII.

II. Preliminaries

A. 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. Interfering links

[1]On that point, other CSMA-like algorithms also need to listen to the transmissions from neighboring links and gather information.

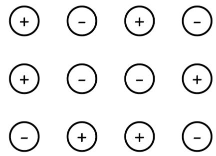

Fig. 1. Ising model and spin values

cannot successfully transmit simultaneously. Without loss of generality, we assume the graph G has at least two nodes and is connected.

We assume the links all have identical capacity and all packets have the same size. We then can 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.

A *schedule* is represented 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 links that can interfere with each other are both ON in a schedule. Thus, a schedule corresponds to an independent set of the graph G. 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 for the purpose of transmission. The set of all configurations is denoted by Ω. A scheduling algorithm is a way to choose a schedule on each time slot.

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 on 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, denoted by Λ, is the closure of the stability region [1].

B. Ising Model and Glauber Dynamics

The I-CSMA algorithm is inspired by a model in physics, called the *Ising model*. We will briefly review the Ising model.

1) Ising Model: The Ising model [19] [20] 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 function 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, i.e., $Z(\beta) = \sum_{\sigma \in \Omega} e^{-\beta H(\sigma)}$.

The Glauber dynamics corresponding to the Ising model is a discrete-time Markov chain 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 $q(-1; \sigma, v) \triangleq 1 - q(+1; \sigma, v)$, respectively, where

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

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.

2) Parallel Glauber Dynamics: The aforementioned Glauber dynamics for the Ising model does single-site update, i.e., only one vertex is selected to update its spin value in each iteration (time slot). One can speed up the Markov chain transition and shorten the time to reach the steady state by performing parallel update. For wireless scheduling algorithms based on Glabuer dynamics, how fast the Markov chain reaches the steady state, often measured by the mixing time, has a significant impact on the delay performance of the network system [21] [10].

In the parallel Glauber dynamics, on each time slot t, an independent set of vertices of the graph is chosen randomly, where no two vertices in the set are neighbors in the graph. The set is called an *updating set* and denoted as ξ. Then, every vertex in $v \in \xi$ will update its spin value according to the probabilities $q(+1; \sigma, v)$ or $q(-1; \sigma, v)$, independently from other vertices in ξ. Here, we assume σ is the configuration at time t. Under minor technical conditions, the parallel Glauber dynamics is irreducible, aperiodic and hence positive recurrent. It is reversible and, from that, it is easy to show the stationary distribution (which is also the limiting distribution) is the Gibbs distribution.

III. Modified Ising Model and I-CSMA Algorithm

A. Modified Ising Model

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 an ON or OFF state[2]. We typically desire as many ON links as possible

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

without interference. In another word, we prefer more ON-OFF neighboring link pairs. 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 the randomized I-CSMA scheduling algorithm. Similar to the existing scheduling algorithms of this family, we wish to have the (stationtary) 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 the earlier algorithms, the concentration happens when the queue sizes are sufficiently large. However, for smaller queue sizes, the algorithms in the family can have differences. 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, since not all the configurations are free of interference. Given a vector $\sigma \in \Omega$, $\sigma(v) = 1$ indicates link v is ON and $\sigma(v) = -1$ indicates link v is OFF.

Given $\sigma \in \Omega$, we associate a *spin value* with each link v and denote it by $s_\sigma(v)$.[3] 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, we set $s_\sigma(v) = A_v$; if v is OFF, we set $s_\sigma(v) = -1$. Note that, in the standard Ising model, $A_v = +1$ for all v. Hence, our modification is a generalization. In the eventual scheduling 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. In a different situation where 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 Sections III-B and III-C, has an embedded Glauber dynamics on the space of all configurations Ω with a stationary distribution that puts more probability masses on higher-energy configurations[4]. In particular, the Glauber dynamics will have the following stationary probability distribution, μ,

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

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

When the Glauber dynamics is in configuration (i.e., state) $\sigma \in \Omega$, the next configuration can differ from σ only at the vertices in the selected updating set (see Section II-B2). For ease of presentation, let us consider the Glauber dynamics with singleton updating sets. Let $\theta^+_{\sigma,v}$ be a configuration, i.e., $\theta^+_{\sigma,v} \in \Omega$, 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 $q(-1; \sigma, v) \triangleq 1 - q(1; \sigma, v)$, 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 state is configuration σ and that link v is selected for updating. 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.

B. I-CSMA Algorithm: An Overview

In the proposed I-CSMA 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. A specific example is, 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 degree of a vertex v, denoted by d_v, is the number of neighbors of v in the graph; and $\bar{d} \triangleq \max_{v\in V} d_v$. In general, to have throughout-optimality, A_v needs to satisfy certain conditions. These conditions are given in the main theorem later (see Theorem 2).

The following is a high-level description of the I-CSMA algorithm.

1) The algorithm has a Glauber dynamics in the background, which is the parallel version of the Glauber dynamics described in Section III-A with the modification that the spin values $\{A_v(t)\}_{v\in V}$ are time-varying.
2) When the state of the Glauber dynamics at time t is not feasible as a schedule, the algorithm converts the state

[3] In this modified Ising model, there is a distinction between the configuration vector σ and the spin-value vector s_σ. In the standard Ising model, the configuration vector σ is also the spin-value vector.

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

into a valid schedule by turning OFF some links in ON-ON neighboring pairs.

We next elaborate on the second step above. Let $\{\sigma(t)\}_{t\geq 1}$ be the sequence of states of the Glauber dynamics, where each $\sigma(t) \in \Omega$. The algorithm generates a sequence of schedules $\{\sigma'(t)\}_{t\geq 1}$, which are feasible by definition, by converting each $\sigma(t)$ into $\sigma'(t)$. The conversion is represented by a map $\psi : \sigma(t) \mapsto \sigma'(t)$. The map ψ is not unique; but we require it to satisfy the following conditions: (i) $\sigma'(t) \in \mathcal{M}$; (ii) $\sigma'(t)(v) \leq \sigma(t)(v)$ for all $v \in V$; and (iii) if $\sigma(t)(v) = 1$ and $\sigma(t)(w) = -1$ for every w that is a neighbor of v in the graph G, then $\sigma'(t)(v) = 1$. Condition (i) says $\sigma'(t)$ must be a valid schedule. Condition (ii) says if link v is OFF in $\sigma(t)$, then it must be OFF in the schedule $\sigma'(t)$. Condition (iii) requires that, in the conversion from $\sigma(t)$ to $\sigma'(t)$, a link v that is ON in $\sigma(t)$ should not be turned OFF if its neighbors are all OFF in $\sigma(t)$; that is, only those links that have ON neighbors are allowed to be turned OFF.

Remark 2: I-CSMA is in fact a family of algorithms. Each version of the family corresponds to a particular choice of the function ψ. For instance, one version may be to order the ON links by the queue sizes and pick a feasible subset of them in the decreasing order of the queue sizes. Another version may be to randomly select one of the links in each ON-ON pair and turn off the selected link. Different versions may provide different performance-cost tradeoffs. The alternatives offered by the family of I-CSMA algorithms are a subject of future investigation.

Remark 3: Under the above three conditions that ψ must satisfy, the most aggressive way of getting a valid schedule $\sigma'(t)$ from a Glauber dynamics state $\sigma(t)$ is to turn OFF all ON-ON neighboring link pairs in $\sigma(t)$. Here, the degree of aggressiveness is measured in terms of the number of links that are turned OFF.

C. I-CSMA Algorithm: Details of One Version

This subsection describes the details of a particular version of I-CSMA. 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 $\xi(t)$) of the interference graph; this is accomplished by using the INTENT messages. The independent set $\xi(t)$ is the updating set mentioned in Section II-B2. During control phase II, only the 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.

I-CSMA 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 $\{0, 1, \dots, W-1\}$ 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[5], 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 $q(1;\sigma, v)$, or 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. Link v sets $\sigma'(t)(v) = 0$. If v has $\sigma(t)(v) = 1$, it executes the following:
   a) Link v selects a random back-off time T_2 uniformly in $\{0, 1, \dots, W'-1\}$ and sets a timer of T_2 mini-slots.
   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 4: In control phase I, the updating set $\xi(t)$ chosen by the algorithm is an independent set. A link v can be included in the updating set $\xi(t)$ if and only if it can successfully broadcast an INTENT message. This happens when link v does not hear any INTENT message on the first T_1 control mini-slots and no other neighboring links are transmitting INTENT messages on the (T_1+1)-th control mini-slot. Furthermore, a successful INTENT message from link v prevents all its neighbors from broadcasting more control messages in the subsequent control mini-slots during phase I. Therefore, when link v manages to be in $\xi(t)$, none of its neighbors can be in $\xi(t)$. Also note that when the window size satisfies $W \geq 2$, for any link v, the probability of it being included in the updating set $\xi(t)$ is positive.

Remark 5: 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 in that neighborhood may be ON

[5] 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.

according to $\sigma(t)$. The algorithm implicitly adopts a particular function $\psi : \sigma(t) \mapsto \sigma'(t)$ mentioned in Section III-B, which is a random function.

Remark 6: It is important to note that ON-ON neighboring pairs are usually few by the design of the algorithm. Furthermore, the use of multiple (i.e., W') mini-slots and randomizing timers reduces the chance of collisions of RESERVE messages from ON-ON neighbors. Therefore, 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.

Remark 7: 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 5). Note that, in control phase II, an ON link will always transmit a RESERVE message, even if it hears other RESERVE messages before its timer expiration. A RESERVE message contains the link's queue size, which may be needed by the neighbors. A link hears the RESERVE messages with queue size information from the ON neighbors during that time slot, 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 sizes of the ON neighbors and the *number* of OFF neighbors.

D. Relationship with Q-CSMA

Q-CSMA works as follows. Suppose link v is selected for consideration of activation. When link v's neighbors are all OFF, the activation probability for link v is equal to $e^{W_v}/(1+e^{W_v})$, where the weight W_v is a slowly increasing function of link v's queue size Q_v, e.g. $W_v = \log(\alpha Q_v + 1)$ or $W_v = \log\log(\alpha Q_v + e)$. When any of link v's neighbors are ON, link v will not be turned ON.

The Markov chain in I-CSMA is in the space of all configurations Ω. In contrast, the Markov chain in Q-CSMA is in the space of valid schedules $\mathcal{M}$, where $\mathcal{M}$ is a strict subset of Ω for any connected graph with more than one node. Although the function ψ above maps a configuration $\sigma(t) \in \Omega$ to a valid schedule $\sigma'(t) \in \mathcal{M}$ and it is $\sigma'(t)$ that is in actual use in I-CSMA, the background Markov chain $\{\sigma(t)\}_{t\geq 0}$ continues to march on in the space of Ω. Thus, over time, the schedules used by I-CSMA and Q-CSMA will be very different in terms of both the sample paths and the probability distributions.

When the weight function (of the queue size) is chosen appropriately, the Markov chain in Q-CSMA can be viewed as a truncated Markov chain of the chain $\{\sigma(t)\}_{t\geq 0}$ in I-CSMA. In other words, whenever a potential transition (in Q-CSMA) leads to a state outside $\mathcal{M}$, the transition is prohibited and the chain stays at the current state. Q-CSMA is a single algorithm in the sense that there is only one version of the embedded Markov chain (under a fixed set of weight functions). I-CSMA is a family of algorithms in the sense that the embedded Markov chain is different for different ψ functions.

When some neighbors of a link v are ON[6], I-CSMA and Q-CSMA are very different. In Q-CSMA, link v has no chance to be activated. In I-CSMA, link v can directly compete for the channel against the ON neighbors. There is a non-zero probability that link v will be activated and that probability depends on the product of link v's queue size and the sum of the queue sizes of its ON neighbors. The probability is non-negligible only if all the above queue sizes are small, or when the spin value A_v is a very slowly increasing function of the queue size, e.g., $A_v = 2(\bar{d}-1) + \log\log(Q_v + e)$. Otherwise, the probability is close to 0, which means link v is unlikely to be turned ON. Thus, we expect that I-CSMA and Q-CSMA work quite differently when (i) the queue sizes are not too large; (ii) the number of neighbors is large for many links; or (iii) the spin value is a very slowly increasing function of the queue size. This helps to explain the performance differences between the two algorithms observed in the simulation results.

When link v's neighbors are all OFF, I-CSMA and Q-CSMA share similarities in the activation probability. For I-CSMA, based on (4), the activation probability for link v can be written as $q(1;\sigma, v) = \frac{\exp(\beta d_v(A_v+1))}{1+\exp(\beta d_v(A_v+1))}$, where d_v is the number of neighbors of v in the graph G. Hence, if we set the weight function W_v in Q-CSMA to be $W_v = \beta d_v(A_v+1)$, Q-CSMA and I-CSMA will have identical activation probability for link v. In the particular case where $A_v = 2(\bar{d}-1) + \log(Q_v + 1)$ in I-CSMA, an equivalent weight function in Q-CSMA would be $W_v = \beta d_v \log(Q_v + 1) + \beta d_v(2\bar{d} - 1)$. This weight function should be compared with $\log(\alpha Q_v)$, which was used in the simulation experiments for Q-CSMA in [9]. We see that, in I-CSMA, there is scaling with a node-dependent factor βd_v. The constant is generally also different.

IV. Throughput Optimality

We will establish that the I-CSMA algorithm is throughput-optimal under the time-scale separation assumption, i.e., the Glauber dynamics is in the steady state on every time slot. The same assumption was made in [9] for the Q-CSMA algorithm. The work in [7] [22] shows how to prove throughput-optimality without the time-scale separation assumption for Q-CSMA-like randomized algorithms. For our algorithm, adaptation of the method in [7] [22] has several technical difficulties. The work to overcome those difficulties is on-going, and the early results look very promising. The proofs without the time-scale separation assumption require the weight function (or A_v in the case of I-CSMA) to be a very slowly growing function of the queue size, e.g., $\log\log$ of the queue size. But, in practice, the $\log$ function usually works well enough, resulting in queue stability for up to moderately high traffic intensity.

A. Outline of the Proof

In this subsection, we outline the proof that I-CSMA is throughput-optimal. The missing details are provided in the next subsection.

We first need some definitions. Given the interference graph $G = (V, E)$, recall d_v is the degree of vertex v in G, i.e., the number of neighbors of v, and $\bar{d}$ is the maximum vertex

[6] As long as link v has at least one ON neighbor, the value of $S(\sigma, v)$ is nonnegative. This is so since $S(\sigma, v) \geq 2(\bar{d}-1) - (d_v - 1) \geq \bar{d} - 1$.

degree, i.e., $\bar{d} = \max_{v \in V} d_v$. Let A be the vector $(A_v)_{v \in V}$. Let $W_v(A)$ denote the weight of vertex v (or equivalently, of link v) under A, defined as $W_v(A) \triangleq d_v A_v$. 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.,

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

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

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

To prove throughput-optimality, we will use a theorem from [23]. Suppose $(f_v)_{v \in V}$ is a finite family of functions on $\mathbb{R}_+$ satisfying the following conditions: For each $v \in V$,
(i) f_v is non-decreasing;
(ii) $\lim_{q \to \infty} f_v(q) = \infty$;
(iii) for any $M_1 > 0$, $M_2 > 0$ and $0 < \epsilon < 1$, there exists $\bar{Q} < \infty$ such that for all $q > \bar{Q}$,

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

The functions $f_v(q) = \log(\beta_v q + 1)$ and $f_v(q) = \log\log(\beta_v q + e)$ satisfy the above conditions, where $\beta_v > 0$ is a constant.

Given such a family of functions $(f_v)_{v \in V}$, 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. Then, the weight of each link $v \in V$ at time t is $W_v(A(t)) = d_v A_v(t) = f_v(Q_v(t))$. Let $A(t) = (A_v(t))_{v \in V}$. Let $||Q(t)|| \triangleq \sqrt{\sum_{v \in V} Q_v^2(t)}$. The following theorem is proved in [23].

Therorem 1: 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 $||Q(t)|| > B$,

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

Then, the scheduling algorithm is throughput-optimal.

The next one is the main theorem of this paper.

Therorem 2: Suppose, for each link $v \in V$, $A_v(t)$ has the form $A_v(t) = f_v(Q_v(t))$, where $Q_v(t)$ is the queue length of link v at time t and each f_v is a function on $\mathbb{R}_+$ satisfying conditions (i), (ii) and (iii) above. Suppose also, for each v, $A_v(t) \ge 2(\bar{d} - 1)$ for all t. Then, every version of the I-CSMA scheduling algorithm is throughput-optimal.

Note that a version of the I-CSMA algorithm is described in remark 2. Also note that f_v satisfies conditions (i), (ii) and (iii) if and only f_v/d_v satisfies those conditions. Hence, there is no real difference in having $A_v(t) = f_v(Q_v(t))$ or $A_v(t) = f_v(Q_v(t))/d_v$ in the theorem.

We next show how to prove Theorem 2. Two important lemmas are needed. The proofs for them can be found in Section IV-B (see Lemmas 7 and 10, respectively).

Lemma 3: $W^* - |E| \le H^*$.

Lemma 4: Suppose $A_v \ge 2(\bar{d} - 1)$ for all $v \in V$. For a given $\sigma \in \Omega$, suppose $H(\sigma) \ge (1 - \epsilon) H^*$, where $0 < \epsilon < 1/(1 + |E|)$. Then, $W(\sigma') \ge (1 - \epsilon(1 + |E|)) W^* - 2|E| - |E|^2$.

We now return to the analysis of the I-CSMA algorithm. When link v is ON at time t (i.e., $\sigma_v(t) = 1$), its spin value is $A_v(t)$, which satisfies the requirement of Theorem 2. 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 1 for I-CSMA.

Under a given vector A and a scalar ϵ with $0 < \epsilon < 1$, let

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

Lemma 5: 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 9: 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.

Proof: For notational simplicity, let us tentatively suppress the dependence on A and ϵ in some of the notations. We have

$$\begin{aligned}\mu(\mathcal{X}) &= \sum_{\sigma \in \mathcal{X}} \mu(\sigma) = \sum_{\sigma \in \mathcal{X}} \frac{1}{Z} e^{\beta H(\sigma)} \\ &< \sum_{\sigma \in \mathcal{X}} \frac{1}{Z} e^{\beta(1-\epsilon)H^*} \le \sum_{\sigma \in \mathcal{X}} \frac{e^{\beta(1-\epsilon)H^*}}{e^{\beta H^*}} \\ &= \sum_{\sigma \in \mathcal{X}} e^{-\epsilon \beta H^*} \le 2^{|V|} e^{-\epsilon \beta H^*}.\end{aligned}$$

The first inequality is due to the definition of $\mathcal{X}$. The second inequality is due to the fact that $Z = \sum_{\sigma \in \Omega} e^{\beta H(\sigma)} \ge \max_{\sigma \in \Omega} e^{\beta H(\sigma)} = e^{\beta H^*}$. The last inequality is because there are at most $2^{|V|}$ items in $\mathcal{X}$.

If $2^{|V|} e^{-\epsilon \beta H^*} \le \delta$, then $\mu(\mathcal{X}) < \delta$. The condition $2^{|V|} e^{-\epsilon \beta H^*} \le \delta$ can be written as

$$H^* \ge (\log(1/\delta) + |V| \log 2)/(\epsilon \beta). \quad (10)$$

Since, for each v, $A_v \to \infty$ as $Q_v \to \infty$, we have $\lim_{||Q|| \to \infty} W^*(A) = \infty$. Since $W^*(A) - |E| \le H^*(A)$ according to Lemma 3, we see that $\lim_{||Q|| \to \infty} H^*(A) = \infty$. Hence, there exists $B(\epsilon, \delta) > 0$ such that (10) is satisfied whenever $||Q|| > B(\epsilon, \delta)$. ■

Lemma 5 says that when $||Q||$ is large enough, the stationary distribution[7] of the Glauber dynamics concentrates on the set $\mathcal{X}^c = \Omega \backslash \mathcal{X}$, i.e., $\mu(\mathcal{X}^c) \ge 1 - \delta$. Note that 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., interference-free or 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 first 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, let the conversion scheme be characterized by the map, $\phi : \sigma \mapsto \sigma'$, 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

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

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

$$\sigma'(v) = \begin{cases} \sigma(v) & \text{for } v \in F^c, \\ -1 & \text{for } v \in F. \end{cases} \tag{11}$$

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 2) Consider the ϵ and δ required by Theorem 1. For the ϵ in Lemma 5, we replace it with any ϵ_1 satisfying $0 < \epsilon_1 < \frac{\epsilon}{2(|E|+1)}$. Then, Lemma 5 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 apply Lemma 4 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 any fixed version of the I-CSMA algorithm characterized by the mapping ψ, as described in Section III-B. Suppose $\sigma''(t)$ is derived from $\sigma(t)$ by that version of I-CSMA. That is, $\psi : \sigma(t) \mapsto \sigma''(t)$ represents the conversion from $\sigma(t)$ to $\sigma''(t)$. 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 1 are satisfied by any version of I-CSMA. Every version must be throughput-optimal. ■

B. More Details for the Proof of Throughput-Optimality

Consider a fixed $\sigma \in \Omega$. Let $F \subseteq V$ contain all the vertices that are ON and each have at least one ON neighbor. In other words, all ON-ON neighboring pairs are in F. Let $F^c = V \backslash F$. Thus, $\{F, F^c\}$ is a *cut* of the graph G, and we can consider the *cut-set* $C(F, F^c)$, which contains all the edges crossing the $\{F, F^c\}$ cut. That is, $C(F, F^c) \subseteq E$ and an edge $(v, w) \in C(F, F^c)$ if and only if $v \in F$ and $w \in F^c$ or $v \in F^c$ and $w \in F$.

Remark 10: F and F^c are dependent on σ.

Fact 1: (i) $F = \emptyset$ if and only if σ is a valid schedule, i.e., $\sigma \in \mathcal{M}$. (ii) For any edge $(v, w) \in C(F, F^c)$ with $v \in F$ (and hence, $w \in F^c$), it must be that $\sigma(v) = 1$ and $\sigma(w) = -1$. (iii) Suppose a configuration σ' is derived from σ by turning OFF the vertices in F, i.e.,

$$\sigma'(v) = \begin{cases} \sigma(v) & \text{for } v \in F^c, \\ -1 & \text{for } v \in F. \end{cases} \tag{12}$$

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

Given $U \subseteq V$, the weight of U under the vector A, denoted by $W_U(\sigma, A)$, is defined as the total weight of all the vertices in U that are ON in σ, i.e.,

$$W_U(\sigma, A) \triangleq \sum_{v \in U : \sigma(v) = 1} A_v d_v. \tag{13}$$

Also define $H_U(\sigma, A)$ by

$$H_U(\sigma, A) \triangleq \sum_{(v,w) \in E : v, w \in U} s_\sigma(v) s_\sigma(w). \tag{14}$$

$H_U(\sigma, A)$ can be thought as the total energy of the subgraph of G induced by the vertices in U. For notational simplicity, we also use $W_U(\sigma)$ for $W(\sigma, A)$ and $H_U(\sigma)$ for $H_U(\sigma, A)$ when there is no confusion.

It is easy to see that the following relations hold.

Lemma 6:

$$W(\sigma) = W_{F^c}(\sigma) + W_F(\sigma). \tag{15}$$

$$H(\sigma) = H_{F^c}(\sigma) + H_F(\sigma) + \sum_{(v,w) \in C(F, F^c)} s_\sigma(v) s_\sigma(w). \tag{16}$$

Given $\sigma \in \Omega$, let $N^-(\sigma)$ be the number of OFF-OFF neighboring pairs, and let $N^+(\sigma)$ be the number of ON-ON neighboring pairs, which is equal to the number of edges in the subgraph of G induced by the vertices in F.

We will next establish that H^* is roughly equal to W^* when they are both large.

Lemma 7: For any $\sigma \in \mathcal{M}$,

$$W(\sigma) - |E| \leq H(\sigma) \leq W(\sigma) \leq W^*. \tag{17}$$

Furthermore,

$$W^* - |E| \leq H^*. \tag{18}$$

Proof: Note that, for $\sigma \in \mathcal{M}$, we have $F = \emptyset$; hence, $H_F(\sigma) = 0$ and $\sum_{(v,w) \in C(F, F^c)} s_\sigma(v) s_\sigma(w) = 0$. By (16),

$$\begin{aligned} H(\sigma) = & H_{F^c}(\sigma) = \sum_{\substack{(v,w) \in E: \\ \sigma(v)=1, \sigma(w)=-1}} A_v - \sum_{\substack{(v,w) \in E: \\ \sigma(v)=\sigma(w)=-1}} 1 \\ = & W(\sigma) - N^-(\sigma). \end{aligned} \tag{19}$$

Hence, for any $\sigma \in \mathcal{M}$,

$$W(\sigma) - |E| \leq H(\sigma) \leq W(\sigma) \leq W^*.$$

We then have

$$\max_{\sigma\in\mathcal{M}} W(\sigma) - |E| = W^* - |E| \le \max_{\sigma\in\mathcal{M}} H(\sigma) \le H^*.$$

■

The next lemma is not directly used in the proof of the main theorem. It is included for completeness in characterizing the relationship between W^* and H^*.

Lemma 8:

$$H^* \le W^* + \bar{d}|E|. \tag{20}$$

Proof: Suppose $\sigma \in \Omega$ achieves the highest energy, i.e., $H(\sigma) = H^*$. Suppose there exists an edge $(v,w) \in E$ such that $\sigma(v) = \sigma(w) = 1$.[8] Consider $\hat{\sigma} \in \Omega$, which is the same as σ except that $\hat{\sigma}(w) = -1$. In other words, $\hat{\sigma}$ is obtained from σ by turning OFF w. We will consider the change from $H(\sigma)$ to $H(\hat{\sigma})$. The gain at (v,w) is $A_v + A_vA_w$. Corresponding to each of the other ON neighbors of w, there is an additional gain from turning OFF w. Corresponding to each of w's OFF neighbors, there is a loss of $A_w + 1$. Hence,

$$\begin{aligned} H(\hat{\sigma}) - H(\sigma) \ge& A_v + A_vA_w - (A_w+1)(d_w-1) \\ =& (A_w+1)(A_v - d_w + 1). \end{aligned}$$

Since σ achieves the maximum energy H^*, we have $H(\hat{\sigma}) \le H(\sigma)$, which implies that $A_v \le d_w - 1 \le \bar{d} - 1$. By switching the roles of v and w, we also get $A_w \le d_v - 1 \le \bar{d} - 1$. Recall that a vertex u is in F if and only if u is ON and u has at least one ON neighbor in σ. Hence, we have established that, if σ achieves H^*, then $A_u \le \bar{d} - 1$ for any $u \in F$.

Let σ' be derived from σ by turning OFF the ON-ON neighboring pairs in σ, i.e., by turning OFF all $u \in F$. Specifically, σ' is given by (12). By Fact 1 (iii), σ' is a valid schedule. By (16), Fact 1 (ii) and that $H_F(\sigma) \le 0$, we have

$$\begin{aligned} H(\sigma) =& H_{F^c}(\sigma) + H_F(\sigma) + \sum_{(v,w)\in C(F,F^c)} s_\sigma(v)s_\sigma(w) \\ \le& H_{F^c}(\sigma) + (\bar{d}-1)|C(F,F^c)|. \end{aligned} \tag{21}$$

Since $\sigma'(v) = \sigma(v)$ for $v \in F^c$, we have $H_{F^c}(\sigma') = H_{F^c}(\sigma)$. Also $H_F(\sigma') = -N^+(\sigma)$, where $N^+(\sigma)$ is the number of ON-ON neighboring pairs in σ. We get

$$\begin{aligned} H(\sigma') =& H_{F^c}(\sigma') + H_F(\sigma') + \sum_{(v,w)\in C(F,F^c)} s_{\sigma'}(v)s_{\sigma'}(w) \\ =& H_{F^c}(\sigma) - N^+(\sigma) - |C(F,F^c)|. \end{aligned} \tag{22}$$

By (21) and (22)

$$H^* = H(\sigma) \le H(\sigma') + \bar{d}|C(F,F^c)| + N^+(\sigma).$$

Since $\sigma' \in \mathcal{M}$, we have $H(\sigma') \le W(\sigma') \le W^*$ according to Lemma 7. Then, we get

$$H^* \le W^* + \bar{d}|C(F,F^c)| + N^+(\sigma).$$

Since $|C(F,F^c)| + N^+(\sigma) \le |E|$ and $\bar{d} \ge 1$, the conclusion of the lemma follows. ■

By combining Lemma 7 and Lemma 8, we get

$$W^* - |E| \le H^* \le W^* + \bar{d}|E|. \tag{23}$$

[8]If there is no such an edge, then H^* is achieved by some $\sigma \in \mathcal{M}$. $H^* = H(\sigma) \le W(\sigma)$ by Lemma 7. Hence, $H^* \le W^*$ and the lemma holds.

The next lemma says, when the energy of a configuration σ is high, every vertex in F has a small spin value.

Lemma 9: Suppose $A_v \ge 2(\bar{d}-1)$ for all $v \in V$. For a given $\sigma \in \Omega$, suppose $H(\sigma) \ge (1-\epsilon)H^*$, where $0 < \epsilon < 1$. Then, for any $v \in F$, $A_v \le \epsilon H^* + |E|$.

Proof: If $F = \emptyset$, there is nothing to show. Subsequently, we assume $F \ne \emptyset$. Suppose we generate a new $\hat{\sigma} \in \Omega$ by turning OFF all the ON-ON neighboring pairs in σ except a particular vertex $u \in F$. That is,

$$\hat{\sigma}(v) = \begin{cases} \sigma(v) & \text{for } v \in F^c \text{ or } v = u \in F, \\ -1 & \text{for } v \in F \text{ and } v \ne u. \end{cases} \tag{24}$$

For a vertex $v \in V$ and a set $U \subseteq V$, let $d_v(U)$ denote the number of neighbors of v in the set U.

We will expand $H(\sigma)$ and $H(\hat{\sigma})$ using (16). First,

$$\begin{aligned} H_F(\sigma) =& -\sum_{(v,w)\in E: v,w\in F} A_vA_w \\ =& -\frac{1}{2}\sum_{v\in F} A_v \sum_{w\in F:(v,w)\in E} A_w. \end{aligned} \tag{25}$$

Using Fact 1 (ii), we have

$$\sum_{(v,w)\in C(F,F^c)} s_\sigma(v)s_\sigma(w) = \sum_{v\in F} A_v d_v(F^c). \tag{26}$$

By (16),

$$\begin{aligned} H(\sigma) =& H_{F^c}(\sigma) - \frac{1}{2}\sum_{v\in F} A_v \sum_{w\in F:(v,w)\in E} A_w + \sum_{v\in F} A_v d_v(F^c) \\ =& H_{F^c}(\sigma) + \sum_{v\in F} A_v\big(d_v(F^c) - \frac{1}{2}\sum_{w\in F:(v,w)\in E} A_w\big). \end{aligned}$$

When $A_w \ge 2(\bar{d}-1)$ for all $w \in V$, $\frac{1}{2}\sum_{w\in F:(v,w)\in E} A_w \ge (\bar{d}-1)d_v(F)$. Also, $d_v(F^c) + d_v(F) = d_v$ for all $v \in F$. If $v \in F$, then $d_v(F) \ge 1$; that is, v has at least one neighbor in F. Hence, for $v \in F$,

$$d_v(F^c) - \frac{1}{2}\sum_{w\in F:(v,w)\in E} A_w \le d_v - \bar{d}d_v(F) \le d_v - \bar{d} \le 0.$$

We arrive at the following result:

$$H(\sigma) \le H_{F^c}(\sigma). \tag{27}$$

Next, we consider $H(\hat{\sigma})$. First, note that σ has $N^+(\sigma)$ ON-ON neighboring pairs, each of which is a pair of vertices in F. But, all such pairs except those of the form $(u,w) \in E$, where $w \in F$, become OFF-OFF pairs in $\hat{\sigma}$. Hence,

$$H_F(\hat{\sigma}) = A_u d_u(F) - (N^+(\sigma) - d_u(F)).$$

Also,

$$\sum_{(v,w)\in C(F,F^c)} s_{\hat{\sigma}}(v)s_{\hat{\sigma}}(w) = A_u d_u(F^c) - \sum_{v\in F, v\ne u} d_v(F^c).$$

By (16),

$$\begin{aligned}&H(\hat{\sigma})\\=&H_{F^c}(\hat{\sigma})+A_u d_u(F)-(N^+(\sigma)-d_u(F))\\&+A_u d_u(F^c)-\sum_{v\in F, v\neq u} d_v(F^c)\\=&H_{F^c}(\hat{\sigma})+A_u d_u-(N^+(\sigma)+\sum_{v\in F,v\neq u} d_v(F^c)\\&+d_u(F^c)-d_u(F^c)-d_u(F))\\=&H_{F^c}(\hat{\sigma})+A_u d_u-(N^+(\sigma)+\sum_{v\in F} d_v(F^c)-d_u). \quad (28)\end{aligned}$$

The last equality uses the fact that $d_u(F)+d_u(F^c)=d_u$.

By (27) and (28),

$$H(\hat{\sigma})\geq H(\sigma)+A_u d_u-(N^+(\sigma)+\sum_{v\in F} d_v(F^c)-d_u).$$

But, $H(\hat{\sigma})\leq H^*$ and $H(\sigma)\geq(1-\epsilon)H^*$. Hence,

$$\begin{aligned}\epsilon H^*\geq&A_u d_u-(N^+(\sigma)+\sum_{v\in F} d_v(F^c)-d_u)\\\geq&A_u d_u-(|E|-d_u)\geq A_u-|E|.\end{aligned}$$

The second inequality above uses the fact that $N^+(\sigma)+\sum_{v\in F} d_v(F^c)\leq|E|$. Finally, we get $A_u\leq\epsilon H^*+|E|$. ■

Suppose $H(\sigma)\geq(1-\epsilon)H^*$. Suppose we generate a new $\sigma'\in\mathcal{M}$ by turning OFF all the ON-ON neighboring pairs in σ, i.e., σ' is given by (12). The next lemma addresses the question whether the weight of σ' is close to W^*.

Lemma 10: 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 $0<\epsilon<1$. Then,

$$W(\sigma')\geq(1-\epsilon(1+|E|))W^*-2|E|-|E|^2. \quad (29)$$

Proof: If $F=\emptyset$, then $\sigma\in\mathcal{M}$ and, by (17), $W(\sigma)\geq H(\sigma)$. Hence, by (18), $W(\sigma)\geq(1-\epsilon)H^*\geq(1-\epsilon)W^*-(1-\epsilon)|E|\geq(1-\epsilon(1+|E|))W^*-2|E|-|E|^2$.

Next, suppose $F\neq\emptyset$. Since $\sigma'\in\mathcal{M}$, by (17),

$$W(\sigma')-|E|\leq H(\sigma')\leq W(\sigma').$$

Then, by (15) and by noticing $W_{F^c}(\sigma)=W(\sigma')$, we have

$$W(\sigma)\geq W(\sigma')\geq H(\sigma').$$

We will expand $H(\sigma)$ and $H(\hat{\sigma})$ using (16). First, note that $H_{F^c}(\sigma)=H_{F^c}(\sigma')$. Since all the vertices in F are OFF in the configuration σ', we have $H_F(\sigma')=-N^+(\sigma)$. Moreover, if $(v,w)\in C(F,F^c)$, it must be that both v and w are OFF in the configuration σ'. Hence,

$$\sum_{(v,w)\in C(F,F^c)} s_{\sigma'}(v)s_{\sigma'}(w)=-|C(F,F^c)|.$$

Therefore,

$$H(\sigma')=H_{F^c}(\sigma)-N^+(\sigma)-|C(F,F^c)|. \quad (30)$$

Next, we consider $H(\sigma)$. By Lemma 9, $A_v\leq\epsilon H^*+|E|$ for every $v\in F$. Hence,

$$\sum_{(v,w)\in C(F,F^c)} s_\sigma(v)s_\sigma(w)\leq(\epsilon H^*+|E|)\,|C(F,F^c)|.$$

By (16) and the fact that $H_F(\sigma)\leq 0$, we have $H(\sigma)\leq H_{F^c}(\sigma)+(\epsilon H^*+|E|)\,|C(F,F^c)|$. Replacing $H_{F^c}(\sigma)$ by using (30), we get

$$\begin{aligned}H(\sigma)\leq&H(\sigma')+N^+(\sigma)+|C(F,F^c)|\\&+(\epsilon H^*+|E|)\,|C(F,F^c)|.\\=&H(\sigma')+N^+(\sigma)+(|E|+1)|C(F,F^c)|\\&+\epsilon|C(F,F^c)|H^*\\\leq&H(\sigma')+\epsilon|C(F,F^c)|H^*+|E|+|E|^2.\end{aligned}$$

The final inequality is because $|C(F,F^c)|\leq N^+(\sigma)+|C(F,F^c)|\leq|E|$.

Since by assumption $H(\sigma)\geq(1-\epsilon)H^*$, we have

$$\begin{aligned}H(\sigma')\geq&(1-\epsilon(1+|C(F,F^c)|))H^*-|E|-|E|^2\\\geq&(1-\epsilon(1+|C(F,F^c)|))(W^*-|E|)-|E|-|E|^2\\=&(1-\epsilon(1+|C(F,F^c)|))W^*\\&-(1-\epsilon(1+|C(F,F^c)|))|E|-|E|-|E|^2\\\geq&(1-\epsilon(1+|E|))W^*-2|E|-|E|^2.\end{aligned}$$

The second inequality is due to (18). Finally, since $\sigma'\in\mathcal{M}$, by (17), $H(\sigma')\leq W(\sigma')$. ■

In summary, when σ has nearly the maximum energy, the derived schedule σ' has nearly the maximum weight.

V. Further Discussions about I-CSMA

A. Control Overhead

The control overhead can be made small by enlarging the data slot size. Suppose the data slot size is equivalent to D control mini-slots. The algorithm has an efficiency ratio of $D/(D+W+W')$, which is the fraction of time used for transmitting data packets. In step 6, the transmitted data packet during a data slot may consist of multiple link-layer frames to improve the efficiency. When the duration of a data slot is extended sufficiently long, each INTENT or RESERVE message is relatively small compared with a large data packet.

Let us consider how large W needs to be. Suppose a neighborhood has d links, all interfering with each other, where $d>1$. Control phase I says each link independently chooses a number uniformly at random from $0,1,\ldots,W-1$. Let $X=(X_0,X_1,\ldots,X_{W-1})$ be a random vector, where each X_i is the number of links that have chosen number i. The random vector X has a multinomial distribution: If a set of integers $k_0,k_1,\ldots,k_{W-1}$ satisfies the property that $k_i\in\{0,1,\ldots,W-1\}$ for each i and $\sum_{i=0}^{W-1} k_i=d$, then

$$\begin{aligned}&\mathrm{P}(X_0=k_0,X_1=k_1,\cdots,X_{W-1}=k_{W-1})\\=&\frac{d!}{k_0!k_1!\cdots k_{W-1}!}(\frac{1}{W})^{k_0}(\frac{1}{W})^{k_1}\cdots(\frac{1}{W})^{k_{W-1}}\\=&\frac{d!}{k_0!k_1!\cdots k_{W-1}!}\frac{1}{W^d};\end{aligned}$$

otherwise, the probability is zero. Let us consider the probability that there exists a number in $\{0,1,\ldots,W-1\}$ chosen by exactly one link. When such an event happens, control phase I will be able to select exactly one link in the neighborhood to be included in the updating set $\xi(t)$. The probability is $\sum_{k_i\neq 1,\forall i}\frac{d!}{k_0!k_1!\cdots k_{W-1}!}\frac{1}{W^d}$. It is possible to compute the probability exactly for only very small values of W and d. For

TABLE I
PROBABILITY BY POISSON APPROXMIATION

d/W	$W=10$	$W=20$	$W=40$	$W=80$
4	0.5327*	0.7817*	0.9523	0.9977
3	0.8016*	0.9607	0.9985	>0.9999
2	0.9574	0.9982	>0.9999	>0.9999
1	0.9898	0.9999	>0.9999	>0.9999
1/2	0.9730	0.9993	>0.9999	>0.9999
1/4	0.8853*	0.9868	0.9998	>0.9999

other cases, we consider the Poisson approximation. Suppose, for each $i \in \{0, 1, \ldots, W-1\}$, the number of links choosing the number i is a Poisson random variable with a mean d/W, and the random variable is denoted by Y_i. Suppose the random variables $Y_0, Y_1, \ldots, Y_{W-1}$ are i.i.d. Then,

$$\mathrm{P}(Y_i = 1 \text{ for some } i) = 1 - (1 - \frac{d}{W} e^{-d/W})^W. \tag{31}$$

For $d/W = 4$, 3, 2, 1, 1/2, and 1/4, the values of $1 - \frac{d}{W} e^{-d/W}$ are 0.9267, 0.8506, 0.7293, 0.6321, 0.6967 and 0.8053, respectively. For these values of d/W, the probability expressed in (31) approaches 1 very rapidly as W increases. As seen from Table I[9], the probability becomes very close to 1 when W is fairly small. For instance, for $W = 10$ and $d = 20$, the probability is about 0.9574; for $W = 20$ and $d = 40$, the probability is about 0.9982; for $W = 40$ and $d = 160$, the probability is about 0.9523. Table I does not show the cases where d/W is much less than 1, since, first, the Poisson approximation is no longer accurate in those cases, and second, there is no need of them. In control phase I, when W is much greater than d, the probability of having at least one number chosen by exactly one link should be extremely close to 1. To summarize, having $W = 20$ is enough to handle up to 60 links in any neighborhood.

Next, the parameter W' can be made small because of the rare occurrence of the ON-ON neighboring pairs in σ. In our simulation, $W' = 8$ is sufficient. Finally, the number of control messages is limited by our design. Consider a neighborhood with d mutually interfering links. In phase I, at most one link transmits an INTENT message if randomization successfully resolves collision. In phase II, a link v is allowed to transmit a RESERVE message only if $\sigma(v) = 1$. The number of such links is small.

B. *Effect of* β

As can be observed from (3), under a fixed set of queue sizes, when β increases, the distribution μ is increasingly biased towards higher-energy states and the I-CSMA algorithm increasingly resembles the max-weight scheduling algorithm, which is known to have a small stationary total queue size [24]. Thus, we expect the total queue size is smaller when β is larger. Furthermore, expression (5) suggests that a larger β value leads to a higher activation probability for a 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 activation, and consequently, smaller stationary queue sizes. However, our simulation experiments have shown that a larger β may increase the mixing time, i.e., the time taken for the Glauber dynamics to reach stationarity. Because of that, a larger β value may increase the total queue size. When β is sufficiently large, the algorithm may not even be throughput-optimal when the time-scale separation assumption is significantly violated (see the beginning of Section IV for the assumption, which we rely on for the proof of throughput-optimality). Thus, there is tension in choosing a larger or smaller β. The effect of β on the total queue size can be seen in the simulation results in Section VII. We may further explore this issue in future work.

VI. HEURISTIC I-CSMA SCHEDULING ALGORITHM

As discussed in Remark 4, the I-CSMA algorithm in Section III-C selects an independent set $\xi(t)$ for parallel update during control phase I on each time slot. Parallel update speeds up the Markov chain transitions and potentially reduces the mixing time and queue sizes. Requiring the updating set $\xi(t)$ to be an independent set is for the theoretical reason that one can still easily compute the stationary distribution of the Glauber dynamics. However, much of the complexity of control phase I is due to the requirement to select an independent updating set. Removing that requirement will simplify the algorithm.

We next propose a much simpler heuristic I-CSMA algorithm. In this algorithm, we allow *every* link to update its ON-OFF status on each time slot based on its neighbors' information during the *previous* time slot. On time slot t, each link v with a non-empty queue runs the following procedure.

Heuristic I-CSMA Algorithm (at link v)

---

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).
2. Link v sets $\sigma(t)(v) = 1$ (chooses ON) with probability $q(1; \sigma, v)$, or it sets $\sigma(t)(v) = -1$ (chooses OFF) with probability $q(-1; \sigma, v)$.
3. Link v sets $\sigma'(t)(v) = 0$. If $\sigma(t)(v) = 1$, it executes the following:
   a) Link v selects a random back-off time T_v uniformly in $\{0, 1, \ldots, W-1\}$ and sets a timer of T_v mini-slots.
   b) When the T_v 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 $\sigma'(t)(v) = 1$.
4. In the data slot, if $\sigma'(t)(v) = 1$, link v transmits a packet.

Compared with the original I-CSMA algorithm in Section III-C, line 2 above replaces all the steps in control phase I and line 3 is the same as control phase II. The heuristic I-CSMA algorithm bypasses the selection of an independent set $\xi(t)$ as the updating set.

[9] We have verified the accuracy of the Poisson approximation by comparing its results with simulation-based sampling experiments. All entries except those marked with * are within a 3% error margin. For the four cases with larger error margins, the sampling results are: 0.37 for $W = 10$ and $d = 40$, 0.69 for $W = 10$ and $d = 30$, 0.99 for $W = 10$ and $d = 3$, and 0.72 for $W = 20$ and $d = 80$.

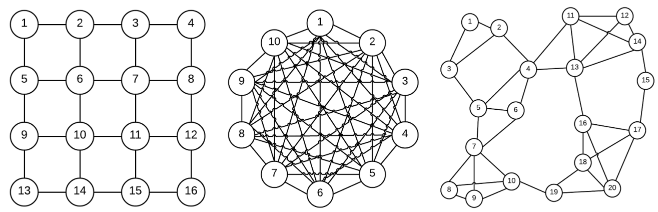

Fig. 2. Interference graphs: 16-link grid, 10-link clique and 20-link random topology. The vertices correspond to wireless links in the network.

We expect the heuristic algorithm to be throughput-optimal (or nearly so) under some mild technical conditions. But, we have no analytical proofs yet. Our simulation results show that its performance is similar to that of the original algorithm (see Section VII-D).

VII. Simulation Results

In this section, we show simulation-based performance evaluation of the I-CSMA algorithm. We compare I-CSMA with Q-CSMA, which is both popular and quite related to I-CSMA. Having proved throughput-optimality, our focus here is the average total queue length, which, by Little's law, is related to the average delay experienced by packets.

We used three interference graphs to evaluate the algorithms in different situations. One is a 4×4 grid with 16 vertices; another is a clique with 10 vertices, also known as an 10-clique; and the last one is a random network with 20 links (see Fig. 2). Recall that the vertices represent wireless links.

The incoming traffic to each link v follows an i.i.d. stochastic process with an average 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, giving enough time for the queues to become steady, and we recorded the sum of the long-time average queue sizes of all the links as the performance measure.

We used two weight functions for Q-CSMA: $W_v(t) = \log(\alpha Q_v(t)+1)$ and $W_v(t) = \log\log(\alpha Q_v(t)+e)$. For our I-CSMA algorithm, we also tried two functions: $A_v(t) = 2(\bar{d}-1)+\log(Q_v(t)+1)$ and $A_v(t) = 2(\bar{d}-1)+\log\log(Q_v(t)+e)$.[10] The window sizes used were $W = 32$ for Q-CSMA and for control phase I of I-CSMA, and $W' = 8$ for control phase II of I-CSMA. Since the α and β values significantly affect the queue sizes in the algorithms[11], we explored a range of values to make appropriate choices.

A. Network Traffic

1) Probability Distributions: For the traffic models, we tried the Bernoulli, Poisson and Pareto distributions for the number of arrivals on each time slot. While the Bernoulli and Poisson models are used in many studies, the random traffic generated under a bounded Pareto distribution can better simulate the bursty nature of LAN traffic [25]. For our experiments, we used a bounded Pareto distribution with the shape factor $\gamma = 1.5$ and the upper bound $H = 1000$. Specifically, the cumulative distribution function is $F(x) = \frac{1-(L/x)^\gamma}{1-(L/H)^\gamma}$, where $x \in [L, H]$. The lower bound L is determined from the mean arrival rate used for an experiment, and therefore, it varies.

2) Arrival Rate Vectors: We used the following scheme to sample the arrival rate vectors for different networks.

a) Grid Network: For the 16-link grid network, we picked 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 s^1 and s^2 each be a 16-dimensional vector representation of L_1 and L_2: For each $i \in \{1, 2\}$, set $s_v^i = 1$ if link $v \in L_i$; otherwise, set $s_v^i = 0$. We chose arrival rate vectors that can be represented as:

$$\lambda = \rho \sum_i t_i s^i, \quad \text{with} \sum_i t_i = 1 \text{ and } t_i \geq 0, \forall i. \tag{32}$$

If $t_1 = t_2 = 0.5$, all the links have equal arrival rates. It can be shown that $\sum_{i=1}^2 t_i s^i$ lies on the boundary of the capacity region for any fixed (t_1, t_2). Hence, λ of the form in (32) is in the capacity region Λ if and only if $0 \leq \rho \leq 1$. The parameter ρ can be interpreted as the traffic intensity or load. We required $0 < \rho < 1$ in the simulation experiments.

b) Clique Network: For the 10-clique network, each link interferes with all other links. We chose 10 different link sets, each containing only one link. We let $s^1 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)'$, $s^2 = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0)'$, ..., $s^{10} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)'$. The arrival rate vectors were set as in (32), with $t_i = 0.1$ for each i. Again, $\sum_{i=1}^{10} t_i s^i$ lies on the boundary of the capacity region, and λ is in Λ if and only if $0 \leq \rho \leq 1$.

c) Random Network: The random topology was created by placing vertices randomly in a unit square, and connecting them by edges based on the distance between each pair of vertices. For the topology shown in Fig. 2, we chose four sets of links:

$$L_1 = \{1, 4, 7, 12, 15, 20\}, L_2 = \{2, 5, 10, 11, 17\},$$
$$L_3 = \{3, 6, 9, 13, 15, 18\}, L_4 = \{1, 5, 8, 14, 16, 19\}.$$

Each set is a maximal schedule for the network. We set the 20-dimensional vectors s^1, s^2, s^3, s^4 accordingly. We chose arrival rate vectors of the form in (32). It can be checked that, for any set of weights t_1, t_2, t_3, t_4 with $\sum_i t_i = 1$, ρ must be no more than 1 for λ to in the capacity region. For the experiments, we set $t_i = 1/4$ for each i and we required $\rho \in (0, 1)$.

3) Uneven Traffic: We also studied the algorithms' performance under uneven arrival rates for different links on the grid network, e.g, using $(t_1, t_2) = (0.6, 0.4)$ and $(t_1, t_2) = (0.7, 0.3)$ in addition to $(t_1, t_2) = (0.5, 0.5)$ in (32).

B. Choices of α and β Values

We first studied the performance of both algorithms under different α (for Q-CSMA) or β (for I-CSMA) values. The α and β values we have tested with are: $0.01, 0.03, 0.1, 0.3, 1, 3$. Fig. 3 and Fig. 5 show the simulation results for Q-CSMA on the 16-link grid network with the $\log$ and $\log\log$ weight

[10] In our experiments, $A_v(t) = 1+\log\log(Q_v(t)+e)$ leads to even smaller queue sizes while maintaining throughput-optimality. Since our theoretical result for throughput-optimality requires $A_v(t) \geq 2(\bar{d}-1)$, we chose to report the cases with $A_v(t) = 2(\bar{d}-1) + \log\log(Q_v(t)+e)$.

[11] β shows up in (4) for I-CSMA.

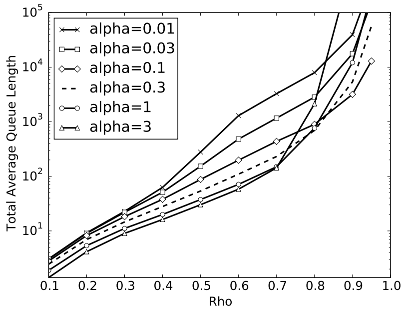

Fig. 3. 16-link grid, Q-CSMA using log function with different α's.

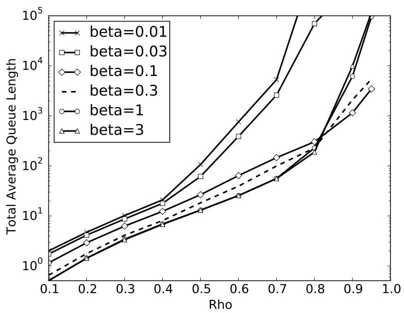

Fig. 4. 16-link grid, I-CSMA using log function with different β's.

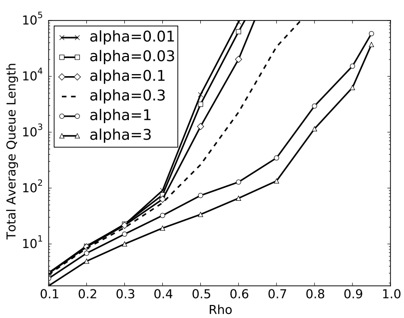

Fig. 5. 16-link grid, Q-CSMA using $\log\log$ function with different α's.

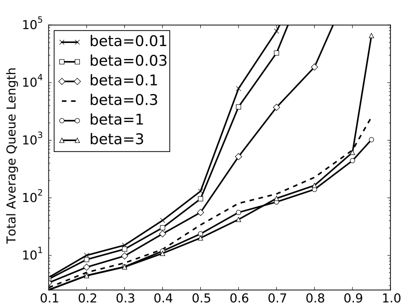

Fig. 6. 16-link grid, I-CSMA using $\log\log$ function with different β's.

functions, respectively. The traffic model is Poisson. Fig. 4 and Fig. 6 are for I-CSMA under the same respective scenarios. The plots are on semi-log scales.

The plots show that the α or β values have large impact

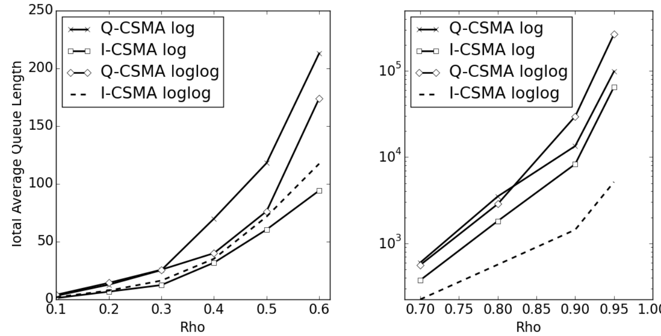

Fig. 7. 16-link grid, Pareto traffic.

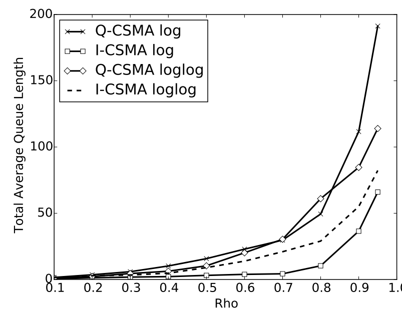

Fig. 8. 10-link clique, Poisson traffic.

on the performance of Q-CSMA or I-CSMA. For I-CSMA, the results confirm our earlier discussion about the effect of β. A very small β tends to result in large queues, since the activation probabilities are not large enough until the queue sizes become large. On the other hand, if β is too large, the queues can also be large, especially when the arrival rate vector is close to the boundary of the capacity region. The reason may be that the activated links seize the channel for too long, causing slow convergence of the Glauber dynamics to the steady state and ultimately violating the time-scale separation assumption. To choose the best α or β values, we experimented on different network topologies and each with different arrival traffic patterns: Poisson, Bernoulli, and Pareto. The values that yielded best results were $\alpha = 0.1$ and $\beta = 0.1$ for Q-CSMA and I-CSMA, respectively, under the $\log$ weight functions, $\alpha = 3$ for Q-CSMA under the $\log\log$ weight function, and $\beta = 1$ for I-CSMA under the $\log\log$ function. We used those best values for the remaining simulation results.

C. Performance Comparison

We compared I-CSMA with Q-CSMA under various network and traffic scenarios. For brevity, we have only included a subset of the results in this paper, which are representative. Note that the vertical axis is sometimes in the linear scale and sometimes in the log scale, depending on the range of queue sizes we wish to show.

Fig. 7 shows the simulation results on the grid network for both algorithms, with the $\log$ and $\log\log$ functions. In order to better examine the differences, we use a semi-log plot only for the high traffic intensity segment $\rho \geq 0.7$. For $\rho \leq 0.6$,

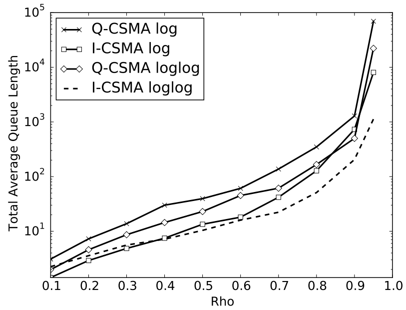

Fig. 9. 20-link random graph, Pareto traffic.

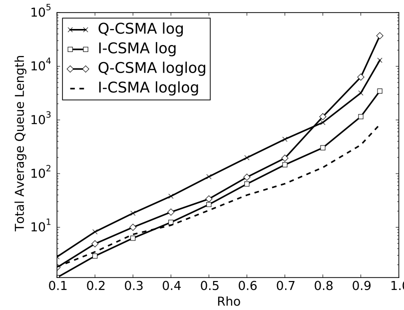

Fig. 10. 16-link grid, uneven Poisson traffic.

we use the linear scale to zoom in and see the queue length differences. The results for the 10-clique network and the random network are shown in Fig. 8 and Fig. 9, respectively. We can observe the following from these figures.

- With a well-chosen β, our I-CSMA algorithm generally leads to smaller queues than Q-CSMA under a well-chosen α throughout the entire feasible region of the traffic intensity $\rho \in [0, 1)$. The queue size reduction over Q-CSMA is between a factor of 2 to 10 at low to medium traffic intensity. At higher traffic intensity, the improvement can be more significant.
- Both algorithms achieve stability for the entire feasible region of the traffic intensity. This is a strong evidence for throughput-optimality.
- For low to medium traffic intensity, the total queue backlog in the network under I-CSMA falls in the range of 10 to 100 packets, which are quite small and bode well for practical applications of the algorithm.
- I-CSMA with the $\log\log$ function works especially well, often better than with the $\log$ function. The reason may have to do with the fact that the $\log\log$ function changes very slowly with the queue size so that the Glauber dynamics is given time to converge to the steady state (See also Section V-B). It may also be because the probability of having ON-ON pairs is higher, allowing more frequently rotations of link transmissions and reducing the chance that some links monopolize the transmission opportunities for too long (see also the discussion in Section III-D).
- I-CSMA and Q-CSMA seem to differ more when the queue sizes are not too large, when the number of neighbors is large on average, and when the traffic is bursty.

The different traffic models have some impact on the simulation results. The Bernoulli and Poisson arrivals are smooth, while the Pareto arrivals are much more bursty, occasionally having a large number of arrivals on a single time slot. The bursty traffic causes the curves to be less smooth than the other two traffic patterns. However, the long-term trend of the results is not altered and I-CSMA is still able to reduce the queue size by at least $1/2$ in most of the cases.

Fig. 10 shows the simulation results on the grid network under uneven Poisson traffic, with $(t_1, t_2) = (0.6, 0.4)$. We observe that the unevenness of the arrival rates has little impact on the results, compared with the case of even arrival rates.

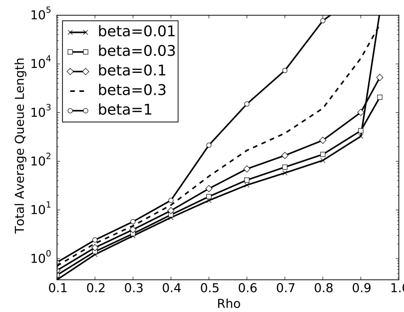

Fig. 11. 16-link grid, Pareto traffic, heuristic I-CSMA algorithm using log function.

Though each individual queue length can be quite different due to the uneven arrival rates, the total queue sizes are similar between the even and uneven cases. I-CSMA still out-performs Q-CSMA by a factor of 2 to 10.

D. Heuristic I-CSMA Algorithm

The simulation experiments for the heuristic I-CSMA algorithm used the same set of interference graphs and the Pareto traffic. We experimented with the $\log$ function with $\beta = 0.01, 0.03, 0.1, 0.3, 1$. The results are shown in Fig. 11.

The figure shows that the heuristic I-CSMA algorithm with an appropriate β value is able to stabilize the queues for ρ up to nearly 1, giving strong indication of throughput-optimality. Even with respect to the total queue size, the heuristic algorithm performs as well as I-CSMA, which can be seen, for example, by comparing Fig. 11 with Fig. 4. In practice, the heuristic algorithm may be preferred due to its simplicity and less control overhead.

VIII. Conclusions and Future Work

In this paper, we propose a randomized link scheduling algorithm, I-CSMA, based on a modified Ising model and the associated Glauber dynamics. The main result is that all versions of I-CSMA are throughput-optimal. The result allows the removal of a major restriction in earlier related algorithms regarding how the Markov chain in the Glauber dynamics is truncated. Consequently, I-CSMA is more flexible than earlier algorithms for achieving secondary objectives after achieving throughput-optimality. Our simulation experiments show that the version of I-CSMA presented in the paper gives better queue-size/delay performance than the popular Q-CSMA. The improvement is significant and consistent, particularly in the regime of low to moderately high traffic intensity. I-CSMA is easily implementable. We also propose a heuristic I-CSMA algorithm, which is observed to have similar performance in simulation studies and is even simpler to implement.

We briefly describe possible future work. First, the proof for throughput-optimality relies on the time-scale separation assumption. We are currently attempting to remove that assumption and find conditions under which the algorithm is still throughput-optimal. The preliminary results are very promising. Second, one may ask whether or when the embedded Glauber dynamics is fast mixing, how the mixing time is related to the parameter β, and what the effects of the mixing time are on throughput-optimality and queue-size/delay performance. Third, the current I-CSMA algorithm is a link-based algorithm in that the algorithm steps are executed by each link. The link-based algorithm description is useful for its conceptual simplicity and ease of presentation; it is sufficient for theoretical analysis and performance evaluation. However, in reality, a wireless link is associated with two wireless devices that can transmit and receive packets. The control of the link is implemented at the two devices. To make the algorithm more practically useful, we have on-going work that converts it into a device-based version. This introduces a different set of challenges. Fourth, the performance-cost tradeoffs of different versions of I-CSMA are worth further study. Fifth, it may be interesting to explore other algorithms or variations from the same class that uses neighboring queue sizes. The new approach appears to be powerful and information-rich, when compared with earlier algorithms' approach of only using a link's own queue size. Its performance benefits are far from being fully explored. Finally, theoretical analysis of the queue-size/delay performance will be valuable; but it is expected to be challenging.

References

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

[2] G. Sharma, C. Joo, N. B. Shroff, and R. R. Mazumdar, "Joint congestion control and distributed scheduling for throughput guarantees in wireless networks," *ACM Transactions on Modeling and Computer Simulation*, vol. 21, no. 1, p. 5, 2010.

[3] B. Li, C. Boyaci, and Y. Xia, "A refined performance characterization of longest-queue-first policy in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1382–1395, 2011.

[4] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, 2007.

[5] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1132–1145, 2009.

[6] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[7] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, 2009, pp. 133–144.

[8] L. Jiang and J. Walrand, "Convergence and stability of a distributed CSMA algorithm for maximal network throughput," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC'09)*, 2009.

[9] 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/ACM Transactions on Networking*, vol. 20, no. 3, pp. 825–836, 2012.

[10] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6541–6555, 2012.

[11] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi, "From Glauber dynamics to Metropolis algorithm: Smaller delay in optimal CSMA," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2012.

[12] V. Subramanian and M. Alanyali, "Delay performance of CSMA in networks with bounded degree conflict graphs," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2011.

[13] P. K. Hwang and X. Lin, "Improving the delay performance of CSMA algorithms: A virtual multi-channel approach," in *Proceedings of IEEE INFOCOM*, 2013.

[14] B. Li and A. Eryilmaz, "A fast-CSMA algorithm for deadline-constrained scheduling over wireless fading channels," *arXiv preprint arXiv:1203.2834*, 2012.

[15] M. Lotfinezhad and P. Marbach, "Throughput-optimal random access with order-optimal delay," in *Proceedings IEEE INFOCOM*, 2011.

[16] K.-K. Lam, C.-K. Chau, M. Chen, and S.-C. Liew, "Mixing time and temporal starvation of general CSMA networks with multiple frequency agility," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2012.

[17] C. Boyaci, B. Li, and Y. Xia, "An investigation on the nature of wireless scheduling," in *Proceedings of IEEE INFOCOM*, March 2010.

[18] D. Shah, D. N. C. Tse, and J. N. Tsitsiklis, "Hardness of low delay network scheduling," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7810–7817, December 2011.

[19] P. Bremaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer Science & Business Media, 1999, vol. 31.

[20] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[21] J. Ghaderi and R. Srikant, "Effect of access probabilities on the delay performance of Q-CSMA algorithms," in *Proceedings of IEEE INFOCOM*, 2012.

[22] ——, "On the design of efficient CSMA algorithms for wireless networks," in *IEEE Conference on Decision and Control (CDC'10)*, 2010.

[23] 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, 2005.

[24] G. R. Gupta and N. B. Shroff, "Delay analysis for wireless networks with single hop traffic and general interference constraints," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 393–405, 2010.

[25] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71–86, 1997.