A Novel Medium Access Control Algorithm for Ad Hoc Networks Based on Ising Model

TAHA AMEEN UR RAHMAN, MOHAMED S. HASSAN, AND MAHMOUD H. ISMAIL, (Senior Member, IEEE)

1Department of Electrical Engineering, American University of Sharjah, Sharjah, United Arab Emirates
2Department of Electronics and Communications Engineering, Faculty of Engineering, Cairo University, Giza 12613, Egypt

Corresponding author: Mohamed S. Hassan (mshassan@aus.edu)

This work was supported by the American University of Sharjah OAP Program.

ABSTRACT Medium Access Control (MAC) scheduling in ad hoc networks is a challenging task due to the trade-offs between fairness, delay and throughput. In this work, we propose a distributed link scheduling algorithm based on the Ising Model from statistical mechanics, by associating a novel Hamiltonian measure with the network such that its optimization yields a feasible schedule. This work overcomes the shortcomings of previous Ising Model based algorithms by incorporating queuing and servicing dynamics as well as fairness measures for improving aggregate throughput and network latency. Our simulations show considerable improvement in performance compared to existing benchmarks over a variety of traffic arrival patterns and network topologies.

INDEX TERMS MAC layer scheduling, multiple access, communication networks, wireless networks, ad hoc networks, Ising model.

I. INTRODUCTION

The IEEE 802 standards employ Medium Access Control (MAC) algorithms in the link layer of the OSI Model to access and interact with the hardware that interfaces with the transmission medium. However, the ever-increasing number of devices that share the transmission medium, coupled with the volume of data they transmit and receive, necessitates fast and efficient algorithms to allow these devices to share the transmission medium concurrently without interfering with each other. The primary objective of such algorithms is to determine and control which links in the network should be active at any given time for optimum performance. For instance, a good scheduler allocates schedules to not only avoid collisions, but also maximize the aggregate network throughput and minimize network delay without compromising fairness or allowing some links to monopolize resources [1]. As networks grow larger and more interconnected, the need for efficient scheduling algorithms is greater now than ever before. Subsequently, MAC protocols for a variety of networks have been extensively studied in literature over the past decade. To provide some context, we review recent developments in this area. A table of abbreviations is provided in Table 1 for reference.

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

A. RELATED WORK

MAC layer scheduling has been studied extensively over the years, but continues to be a hot topic of research due to its extensive utility in a variety of networks. For example, Internet-of-Things (IoT) networks consist of a large number of connected devices that must communicate without interference [2]. Hence, MAC-layer scheduling protocols to avoid network congestion and minimize delay for IoT networks have been studied, for example, in [3], [4]. Another example of networks requiring MAC scheduling algorithms is Wireless Sensor Networks (WSNs), where large networks of sensors communicate with each other [5]. In addition to the aforementioned constraints on throughput and delay, WSN algorithms must also consume low power so as to not deplete the network resources [6], and thus, efficiency is paramount in this application. Algorithms to achieve collision-free scheduling and minimize the aggregate network latency have been studied, for example, in [7], [8], and a survey of approaches to MAC scheduling in WSNs can be found in [9]. Yet another paradigm where efficient resource allocation is crucial is Cognitive Radio Networks (CRNs), where unlicensed Secondary Users (SUs) opportunistically access the spectrum of available frequencies in the absence of Primary Users (PUs) [10], [11]. In addition to the usual objectives of avoiding collisions, maximizing throughput and minimizing latency, MAC scheduling algorithms for CRNs must
also be spectrum-aware [12], and account for mobility-the process by which SUs hop between frequency bands depending on spectrum availability [13]. A recent survey of MAC scheduling algorithms for CRNs can be found in [14], where the authors also provide a classification of MAC protocols based on their mechanism of medium access. Specifically, recent trends in MAC layer protocols focus on a combination of random access protocols and discrete-time protocols. A simple approach to scheduling involves variants of greedy algorithms - such as the Longest-Queue-First (LQF) algorithm, in which links with the longest queue of packets in their buffer are given priority to transmit [15]. However, networks such as WSNs and CRNs are often ad hoc, i.e. they do not have a centralized controller. Hence, a link has no way of knowing about the state of other links - except those in its vicinity that it can directly communicate with. These relationships can be modeled using interference graphs, so that the objective is then to find the Maximum-Weighted Independent Set (MWIS) of the graph, where the weights on the links are functions of queue length. This is the famous MWIS algorithm, which is not only throughput optimal, but can also stabilize queues for all arrival rates as long as they do not exceed the network capacity [16]. However, it requires solving an NP-hard combinatorial problem using global information at each time step and thus is not implementable in practice. Consequently, distributed algorithms that attempt to achieve the MWIS objective have gained considerable attention in recent years [15], [17]. For example, the Distributed Greedy Maximal Scheduling (D-GMS) algorithm attempts to approximate the longest-queue-first algorithm in a distributed manner, through localized communication between neighboring links [18]. Although D-GMS procedures work well when network traffic is relatively low, their performance is rather limited from the perspective of stabilizing network queues [19]. This motivated a new class of MAC protocols that are based on Carrier Sense Multiple Access (CSMA) and its variants, which have gained popularity due to their simplicity of implementation [20]. For example, queue-length information based approaches have garnered considerable attention [19], [21], [22]. Subsequently, contention resolution protocols such as Queue length based CSMA (Q-CSMA) employ random access protocols when queue lengths are high, and switch to D-GMS-like procedures when the link queue length falls below a specified threshold. Such approaches have resulted in improved overall network performance [19].

Despite these algorithms being easily implementable and provably throughput-optimal, CSMA often exhibits a large delay [23]. This is attributed to the tendency of some links to monopolize the resources at the expense of others, thereby compromising fairness. Subsequently, there have been investigations into the delay performance of such algorithms [23]–[25]. Most of these algorithms use a Markov chain to sample the independent sets of a graph to heuristically attain the MWIS objective, using Glauber dynamics based approaches [1]. However, a more generalized approach was proposed in [26] where the authors provide a framework to interpolate from the Glauber dynamics approach to the Metropolis-Hastings (MH) Algorithm [27], citing improved delay performance. The close relationship between the Ising Model and the Metropolis Algorithm then motivated the use of the Ising formulations in [31] to introduce a simple energy function, which is optimized using simulated annealing [32] to obtain a feasible schedule for transmission. However, the authors do not explicitly account for the queue lengths in the network and they only study the volume of data transmission and information delay without taking any fairness measures into consideration. Furthermore, their system model assumes an infinite transmission rate so that each link can send all its backlog at once as soon as it is allowed to transmit. Finally, the authors only consider random graphs instead of common network topologies. Although their algorithm achieves low delay in small networks, simulations show that it is not scalable for larger networks.

B. CONTRIBUTIONS

Inspired by the Q-CSMA and MH algorithms, we propose a novel MAC scheduling algorithm and implement it to study its performance in terms of network throughput, delay and fairness. Our algorithm associates a Hamiltonian measure to a given ad hoc network, such that minimizing it yields a
schedule that not only avoids collisions, but also reduces network latency and increases aggregate throughput. The construction of the Hamiltonian is such that it can be optimized using only local communication in a distributed manner, making it applicable for any ad hoc network. Furthermore, our Hamiltonian accounts for the dynamics of both queuing and servicing, considers queue lengths at the buffers and inactivity time for the links, and incorporates this information in the decision-making process. It is easy to implement in a distributed manner, requiring only local communication and channel sensing capabilities for the links in the network. It also requires much lesser local information exchange between links compared to the annealing process in [30], making it practical for implementation in ad hoc networks. Our simulations show that this framework performs better than commonly used algorithms over a variety of network topologies and traffic arrival patterns. We also believe that the flexibility of this approach allows analogous Hamiltonian measures for other specific ad hoc networks such as WSNs and CRNs, on which our algorithm can be applied.

The rest of this paper is organized as follows. Section II introduces the Ising Model. It is followed by Section III where the system model is discussed, and the relevance of the Ising Model to our work is described. Based on this, we propose a distributed scheduling algorithm in Section IV. The results of simulating the algorithm and comparing it with existing benchmarks are discussed in Section V. Finally, Section VI draws some conclusions.

II. THE ISING MODEL

In statistical mechanics, the Ising Model of ferromagnetism uses a graph, \( G(\mathcal{V}, \mathcal{E}) \), to model the atomic structure of a magnetic material so that the vertices, \( \mathcal{V} \), represent atoms and the edges, \( \mathcal{E} \), represent the interaction between them. Each vertex, \( v \in \mathcal{V} \) has a spin, \( \sigma_v \in \{-1, +1\} \) that represents its magnetic dipole moment, and \((v_i, v_j) \in \mathcal{E}\) if and only if \( \sigma_v \) influences \( \sigma_j \). For example, the Ising Model defined on a \( 4 \times 4 \) lattice is shown in Fig. 1, where positive and negative spins are arbitrarily assigned. The lattice structure indicates that each spin influences its spatial neighbors as represented by the edges. The state of a system with \( N \) spins can then be described by the spin configuration \( \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N) \).

Finally, the energy of the system is measured by the Hamiltonian [33]:

\[
H(\sigma) = - \sum_{(i,j) \in \mathcal{E}} J_{ij} \sigma_i \sigma_j - \mu \sum_{j \in \mathcal{V}} h_j \sigma_j,
\]

where \( J_{ij} \) is a measure of inter-atomic interaction (influence of \( v_i \) on \( v_j \)), \( h_j \) is a measure of interaction of \( v_j \) with the external magnetic field and \( \mu \) is the magnetic moment [33]. Most of the existing literature on the Ising Model assumes that interaction between spins is uniform and that no external magnetic field is present. Thus, the energy is then given by

\[
H(\sigma) = -J \sum_{(i,j) \in \mathcal{E}} \sigma_i \sigma_j.
\]

The probability that the system is in a particular state \( \sigma \) depends on the inverse thermodynamic temperature, \( \beta \). Let \( P_\beta(\cdot) \) be the probability measure defined on the space of configurations, \( \Omega = \{\sigma \mid \sigma_i = \pm 1 \ \forall \ i = 1, 2, \ldots, N\} \), at inverse temperature \( \beta \). Then, \( P_\beta(\sigma) \) is given by the Boltzmann distribution [33]:

\[
P_\beta(\sigma) = \frac{e^{-\beta H(\sigma)}}{\sum_{\sigma \in \Omega} e^{-\beta H(\sigma)}}.
\]

Clearly, when \( \beta = 0 \), each configuration is equally likely. The key observation is that at higher values of \( \beta \), configurations with lower energy are favorable, as evident from Equation (3). However, the direct computation of this probability is intractable, due to the large sum over \( |\Omega| = 2^N \) configurations, except under very specific graph topologies. Consequently, Monte-Carlo Markov Chain (MCMC) methods such as the MH algorithm are used in the literature to sample from the distribution. The MH algorithm considers a Markov Chain over the state space \( \Omega \) and permits transitions between spin configurations through spin-flips so that in the steady state, the Markov chain converges to the sample drawn from the distribution. This result is exploited in our work in Section III-C, by viewing spin flips as a time evolution of the system, knowing that the inverse temperature parameter determines the type of configuration to which the system converges in time.

III. SYSTEM MODEL

The Ising Model is relevant to our work because of the similarities in the underlying graph structure between the magnetic system in Section II, and the interference graph of an ad hoc network. In particular, one can treat the atoms as network links, and their magnetic spins as a binary variable that determines whether or not the link is allowed to transmit. The interactions between atomic spins is manifested in our model as interference between links. We have the liberty to associate an energy measure with this network, and constructing a careful choice that can enable the use
of the MH algorithm to obtain a sample that optimizes this energy measure. The MH algorithm is useful because it can be implemented in a distributed sense and is practical for ad hoc networks where there is no centralized controller. The remainder of this section elaborates the system model.

A. INTERFERENCE GRAPH
Consider a network with $N$ links, such that the interference graph, $G(V, E)$ is defined so that $v_i \in V$ is the $i$-th link and $(v_i, v_j) \in E$ if and only if $v_i$ and $v_j$ cannot simultaneously transmit. Thus, interfering links are connected with an edge. The spin value associated with each link is a time varying binary variable which determines whether or not the link is allowed to transmit. Since only links with positive spin can transmit, the configuration $\sigma$ is said to be a feasible schedule if the vertices with positive spin form an independent set of the graph, and hence no collisions can occur. Figure 2 shows a 5-link network where packets can only exit the buffer of link $v$ if $\sigma_v = +1$ during that time slot, and packets arrive at buffers independent of the spin value, as per the queuing dynamics described in Section III-B. Here, the inward arrows represent packet arrivals, outward arrows represent a transmission, and the buffer occupancy is shaded. Note that $\sigma = \{-1, -1, +1, +1, -1\}$ as shown is feasible.

![Figure 2. Interference graph with feasible schedule for 5-link network.](image)

B. QUEUING DYNAMICS
We consider a discrete-time (slotted) queuing model where packets are of identical size, and can only be transmitted at predefined instants in time slots. Let $Q_v(t)$ be a vector containing queue lengths, so that $Q_v(t)$ is its $v$-th entry, which represents the queue length at the transmitter buffer of link $v$ at time $t$. Further, let $A_v(t)$ represent the stochastic process that counts the number of packet arrivals at link $v$ for time slot $t$. Note that $A_v(t)$ and $A_w(t)$ are allowed to follow different arrival processes, and are not required to be i.i.d. Further, let $R$ be the number of packets that a link can transmit in one time slot as a proportion of the maximum possible number of packet arrivals, i.e. $R$ is a normalized transmission rate. Then, without loss of generality, we can assume that $A_v(t) \in [0, 1]$ and $R$ packets are transmitted at time slot $t$ if $\sigma_v(t) = +1$. Hence, the number of packets in the buffer at time slot $(t + 1)$ is given as

$$Q_v(t + 1) = \max \left(0, Q_v(t) + A_v(t) - R \times \frac{\sigma_v(t) + 1}{2} \right), \quad (4)$$

where $\sigma_v(t) + 1 \in [0, 1]$ is an indicator function that indicates whether or not link $v$ was allowed to transmit at time slot $t$. The max($\cdot$, $\cdot$) function accounts for the fact that queue length is non-negative and thus, no transmission can occur when the buffer is empty, even if the spin associated with the link is positive.

C. NETWORK HAMILTONIAN
Let $\eta = \eta(t)$ be a vector so that its $v$-th entry, $\eta_v(t)$, denotes the number of time slots that have elapsed since the last transmission at link $v$, until time slot $t$. Observing that $\eta_v(t) = 0$ if and only if $\sigma_v(t) = +1$, we propose the following Hamiltonian to associate with a given network configuration $\sigma$ at time $t$. Note that the time dependence is implicit as $\sigma, Q$ and $\eta$ are all functions of $t$.

$$H(\sigma, Q, \eta) = A \sum_{(i,j) \in E} \frac{\ln(1 + Q_i) \times \ln(1 + Q_j)}{\ln(R_{b_{max}})^2} \sigma_i \sigma_j$$

$$- \sum_{v \in V} \left[ B \frac{\ln(1 + Q_v) \sigma_v}{\ln(R_{b_{max}})} - C \frac{\ln(1 + \eta_v)}{\ln(b_{max})} \right],$$

where $A, B$ and $C$ are weights that balance the trade-offs between delay and fairness, and $b_{max}$ is the maximum possible buffer occupancy before it overflows. In the rest of this discussion, we explain the rationale behind the choice, referring to the individual terms in the Hamiltonian as the $A, B,$ and $C$ terms respectively. For notational simplicity, we will refer to the above Hamiltonian as $H(\sigma)$, where the dependence on queue length, inactivity, and time is understood to be inherent in the spin configuration $\sigma$.

Our claim is that network latency, throughput, and fairness are improved if the spin configuration at each time slot is chosen such that $H(\sigma)$ is minimized. The intuition is that since $H(\sigma)$ is increased whenever $\sigma_v \sigma_w = 1$ for an interfering pair of links, $(v, w)$, minimizing $H(\sigma)$ entails minimizing the ON-ON pairs, which result in collisions, and also the OFF-ON pairs, which result in inefficient utilization of network resources. In other words, the $A$ term is minimized when the number of ON-OFF pairs is maximized, which is required to approximate the largest independent set in the graph. Further, despite ON-OFF pairs being desirable, it is often impossible to only have ON-OFF pairs in a feasible schedule, as is the case in Fig. 2. Scaling the $A$ term (the sum over the edges) by a product of a slowly increasing function of the queue lengths ensures that whenever an OFF-OFF pair must exist, the links that are switched OFF are those with smaller queue lengths, as doing so increases $H(\sigma)$ by only a small amount. Next, the sum over the vertices consists of two separate terms. The $B$ term imposes that in an ON-OFF pair, the link with higher queue length is ON. Note that the contribution of a given interfering pair $(v, w) \in E$ to the $A$ term is the same regardless of which of the two links is ON. However, the $B$ term mandates that $H(\sigma)$ is lesser when the link with lesser queue length is OFF. This locally imposes a “Longest-Queue-First” criterion. Finally, the $C$ term prevents links from
monopolizing the network resources by imposing a penalty for every consecutive transmission. Thus, if a link that has been transmitting consecutively for a long time is switched OFF, \( H(\sigma) \) is significantly reduced. Finally, each term is normalized so that their magnitudes are comparable. Note that the underlying process is not a Markov chain due to the dependence on \( \eta_i(t) \). This provides an alternative approach to the higher order Markov chain method pursued in [34].

IV. PROPOSED SCHEDULING ALGORITHM

In this section, we propose a scheduling algorithm based on the above Hamiltonian optimization problem. Our formulation allows us to minimize \( H(\sigma) \) in a distributed manner. The idea is for a link to decide whether to “flip” its spin based on whether doing so reduces \( H(\sigma) \). This is possible because although the link has no way of knowing the Hamiltonian as an absolute value (which requires global information), the difference caused to the Hamiltonian due to a spin flip can be calculated using local information, as will be shown later. In particular, if the flip reduces \( H(\sigma) \), it is kept, whereas if it increases \( H(\sigma) \), it is kept with a certain probability. This probabilistic approach reduces the chance of converging at a configuration that is only a local minimum. We now propose an algorithm to minimize our Hamiltonian inspired by the MH and Q-CSMA algorithms. In our algorithm, a time slot is divided into four phases: Local Communication (P1), Control I (P2), Control II (P3), and Data Transmission (P4). P2 and P3 are further divided as shown in Fig. 3.

P1 - Local Communication: At the beginning of each time slot, link \( i \) sends the queue length in its buffer, \( Q_i(t) \), and the number of time slots for which it has been inactive, \( \eta_i(t) \), to its neighbors \( \mathcal{N}(i) \).

P2 - Control I: Each link sets its spin value to minimize \( H(\sigma) \) in a distributed manner, using a procedure that depends on whether or not its queue length is above a preset threshold, \( Q_i \). For small queue lengths, the well-known D-GMS procedure is adopted, similar to the hybrid Q-CSMA procedure (see [19] for D-GMS and hybrid Q-CSMA). This threshold is required because at low queue lengths, D-GMS is found to perform better [19]. Nevertheless, it is when queue lengths are high that the requirement for efficient scheduling is highest. For queue lengths above \( Q_i \), we propose the following procedure. For each mini-slot, the parameter \( \beta \), is gradually increased as per a predefined “cooling schedule” [35]. From (3), we know that at high \( \beta \), we attain a schedule with low Hamiltonian - the benefits of which are evident from Section III-C. Although the calculation of \( H(\sigma) \) requires global information, the difference in Hamiltonian caused due to a spin flip, \( \sigma_i \mapsto -\sigma_i \) only requires local information as seen in (5). Let \( \sigma^{(o)} \) and \( \sigma^{(n)} \) be the spin configurations before and after flipping \( \sigma_i \). Then, we calculate the difference in Hamiltonian due to a spin flip, \( \Delta_i H = H(\sigma^{(n)}) - H(\sigma^{(o)}) \) for a fixed time \( t \) as:

\[
\Delta_i H = \frac{2A \ln(1 + Q_i)\sigma_i^{(n)}}{\ln(Rb_{\text{max}})^2} \left[ \sum_{j \in \mathcal{N}(i)} \ln(1 + Q_j)\sigma_j \right] - \frac{2B \ln(1 + Q_i)\sigma_i^{(n)}}{\ln(b_{\text{max}})} - \frac{C}{\ln(b_{\text{max}})} \ln \left( \frac{1 + \delta_i^{(o)}}{1 + \eta_i^{(o)}} \right), \tag{5}
\]

where \( \mathcal{N}(i) \) is the set of neighbors of link \( i \), \( \sigma_i^{(o)} \) is the spin at node \( i \) after the flip, \( \eta_i^{(o)} \) is the inactivity time before the flip, and \( \delta[\cdot] \) is the discrete delta function, defined as \( \delta[x] = 1 \) if \( x = 0 \), and 0 otherwise. It appears because flipping a spin from OFF to ON resets \( \eta_i \) to zero regardless of the inactivity time, and conversely, flipping a spin from ON to OFF sets \( \eta_i \) to one.

P3 - Control II: The preliminary spin configuration \( \sigma \), obtained at end of P2 need not be feasible. However, since \( H(\sigma) \) is near-minimum, we expect many ON-OFF pairs so that only a few links have to be turned OFF to achieve a feasible schedule. The objective of P3 is to map \( \sigma \) to a feasible schedule, either by switching one of the links OFF in an ON-ON pair uniformly at random (u.a.r.), or by imposing that the link with the shorter queue is switched OFF. Our simulations show that the choice of the map does not noticeably influence the algorithm performance as the map does not significantly alter the configuration. A similar result is proved in [28].

P4 - Data Transmission: If \( \sigma_v = +1 \), link \( v \) transmits its data packets in this sub-slot with normalized transmission rate \( R \). The complete process with all the aforementioned steps is summarized in Algorithm 1.

V. SIMULATION

To verify the performance of our algorithm, we implement it in MATLAB and compare simulation results under a variety of network conditions with existing algorithms such as Distributed Greedy Maximal Scheduling (D-GMS), Q-CSMA, I-CSMA and Longest-Queue-First (LQF). All these algorithms are described in Section I-A, where references to their respective original sources are also provided. Our simulations compare the aggregate network throughput, average network delay and network fairness for different network topologies and traffic arrival patterns. Note that LQF is a centralized algorithm that uses global knowledge to sort queue lengths.
Algorithm 1: Proposed Scheduling Algorithm at Time $t$

**PHASE 1:**

$\forall i, \forall j \in \mathcal{N}(i)$, link $i$ sends $[Q_i, \eta_i]^T$ to $j$;

**PHASE 2:**

for $v \in \mathcal{V}$ such that $Q_v < Q_i$ do

(D-GMS): Link $v$ selects back-off time:

$T_v = W \lfloor B - \log_b(Q_v(t) + 1) \rfloor + \text{Uniform}[1, W]$;

if Link $v$ hears an INTENT message from $w \in \mathcal{N}(i)$ before mini-slot $T_v$ then

$\sigma_v \mapsto -1$;

else

Link $v$ transmits INTENT message to $w \in \mathcal{N}(v)$ at mini-slot $T_v$;

if INTENT message of link $v$ has a collision at mini-slot $T_v$ then

$\sigma_v \mapsto -\sigma_v$;

else

$\sigma_v \mapsto +1$;

end

end

for $j = \{1, 2, \ldots, |\mathcal{B}|\}$ do

for $v \in \mathcal{V}$ such that $Q_v > Q_j$ do

Link $v$ calculates $\Delta H_a(\sigma)$ using (5);

if $\Delta H_a(\sigma) < 0$ or $e^{-\delta} \Delta H_a(\sigma) > \text{random number generated u.a.r. between 0 and 1}$ then

$\sigma_v \mapsto -\sigma_v$;

else

$\sigma_v \mapsto +1$;

end

end

**PHASE 3:** for $v \in \mathcal{V}$ such that $\sigma_v = +1$ do

for $w \in \mathcal{N}(v)$ such that $\sigma_w = +1$ do

if $Q_v \leq Q_w$ then

$\sigma_v \mapsto -1$;

end

end

end

**PHASE 4:** if $\sigma_v = +1$ then

Link $v$ transmits with transmission rate $R$;

end

D-GMS is set at $Q_i = 100$. Since both hybrid Q-CSMA and our proposed algorithm switch to D-GMS when $Q < Q_i$ (and therefore become equivalent), we are interested in comparing the performance of these algorithms when $Q > Q_i$. Hence, the initial queue lengths are chosen to be sufficiently high enough. All reported figures are an average from several runs of the algorithms.

The first performance metric we are interested in is the network delay, which is directly related by Little’s law to the average queue length [29], computed as follows:

$$Q_{avg}(t) = \frac{1}{N} \sum_{i=1}^{N} Q_i(t). \quad (6)$$

Secondly, we are also interested in the aggregate throughput [30] measured as follows:

$$D(t) = \sum_{i=1}^{N} D_i(t), \quad (7)$$

where $D_i(t)$ is the cumulative amount of data transmitted by link $i$ from initial time till slot $t$. Another measure of delay that we track is the average time since last transmission:

$$\eta_{avg}(t) = \frac{1}{N} \sum_{i=1}^{N} \eta_i(t). \quad (8)$$

which is the time evolution of the average waiting time for the network links [30]. Lastly, we measure the fairness of the network by Jain’s index [36],

$$J(x_1, \ldots, x_N) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2}, \quad (9)$$

where $x_i$ is the ratio of $D_i(t)$ to the cumulative amount of data that link $i$ would have transmitted had it been ON the entire time. This takes both link occupancy and transmitted data into consideration.
We begin by confirming that our algorithm does indeed minimize the network Hamiltonian, by tracking $H(\sigma)$ as a function of the sequential spin flips (or lack thereof) during the control phase of our algorithm. Figure 4 shows the network Hamiltonian evolution for a randomly chosen time slot. Clearly, although the flips are done using local information, the impact is global in nature. Note that there are instances during the evolution when a spin flip increases the Hamiltonian, so that $H(\sigma)$ is not a non-increasing function. This is due to the tendency of the MH algorithm to avoid local minima during the optimization process.

We now simulate our algorithm in MATLAB for different network topologies and traffic patterns. Figure 5 shows the time evolution of $Q_{\text{avg}}(t)$, $D(t)$, and $\eta_{\text{avg}}(t)$ for various scheduling algorithms. In particular, it studies the performance of these algorithms for different traffic arrival patterns while assuming a lattice topology for the underlying interference graph. The first row in Fig. 5 shows the results obtained for a $6 \times 6$ lattice assuming Poisson arrivals. Here, the Poisson arrival rate for each node is different and chosen randomly, such that they are distributed uniformly between 0 and 0.8. The initial queue length is also chosen uniformly at random, between 400 and 500 packets per link, and $R = 2$. It is clear from Figs. 5(a) and (b) that in terms of stabilizing network queues and maximizing aggregate throughput, our algorithm...
performs better than all the distributed algorithms. Further, Fig. 5(c) shows the average waiting time for a link before it receives the opportunity to transmit, $\eta_{avg}(t)$. Clearly, it is the least for our proposed algorithm as compared to the other distributed schemes. Only LQF performs better but it has the exclusive advantage of global information. Finally, the fairness index calculated is $[0.931, 0.880, 0.879, 0.977, 0.959]$ listed in order of the figure legends. I-CSMA has the best fairness ratio due to its inherent design which places a link activation probability of 0.5 even for a link with empty buffer [29]. However, this is achieved at the cost of reduced throughput and larger delay. Our algorithm overcomes this problem.

A more realistic scenario is captured in Figs. 5(d), 5(e) and 5(f), where ON-OFF bursty traffic is simulated. In particular, the traffic pattern assumes a Pareto distributed inter-arrival time between packets, with mean of 8 slots and shape parameter 1.5. The simulation is carried for a large network with 144 links, such that the interference graph forms a $12 \times 12$ lattice, and the transmission rate is assumed to be $R = 2$. The results obtained are very similar to the case of Poisson arrivals, and it is observed that our algorithm performs best among the distributed schemes in terms of reducing the queue length at the buffers, and maximizing the volume of transmitted data. Further, the fairness indices for this simulation are $[0.9977, 0.9965, 0.9952, 0.9985, 0.9849]$. Evidently, LQF results in links with high queue length monopolizing the network when the variance in queue lengths at the links is high.

Next, we consider a Bernoulli random process as the traffic arrival pattern. Here, at each time slot, a packet either arrives or does not arrive at the link buffer with equal probability, for all the links in the network. We consider a $10 \times 10$ lattice topology for the interference graph, and we assume $R = 2$. Figures 5(g), 5(h) and 5(i) show similar performance results as compared to the previous cases. Finally, the fairness indices of $[0.9977, 0.9965, 0.9952, 0.9985, 0.9849]$ are comparable to each other all cases. This suggests that our proposed algorithm can improve the delay performance, as well as overcome the throughput limitations of other distributed algorithms. The results also indicate that the performance of our algorithm is not affected by the size of the network or the the traffic arrival patterns. This is expected as the algorithm uses $Q_v(t)$ and $\eta_v(t)$ for link $v$ at time $t$, regardless of the arrival pattern at the link.

Next, we implement our algorithm for different network topologies. Figure 6 shows the ring and random topologies on which the algorithm was tested. In particular, Fig. 6(a) shows a 25-link ring, whereas Fig. 6(b) considers an underlying interference graph of 25-links such that each link interferes with all other links with probability 0.2. We simulate a 25-link ring with Poisson arrival rates distributed uniformly across $[0, 0.8]$. Further, the initial queue lengths are uniformly distributed between 300 and 1000 packets per link, and we assume that $R = 1$. Clearly, our algorithm responds much faster to the queues and achieves near-global level performance as seen in Fig. 7(a) and Fig. 7(b). It is particularly interesting to observe the large initial waiting time.
for LQF. This effect is more pronounced in smaller networks, and is a consequence of the large variance in initial queue length. The links with large queue length initially monopolize the network until the queue length at other links exceed theirs. Consequently, the transmission schedule remains static until queue lengths become comparable, thereby accounting for the large average waiting time during the initial phase of the simulation. Our algorithm manages to achieve throughput and delay performance comparable to LQF, but without the link monopolizing problem.

Finally, we implement a random graph as the topology of the network. We consider a random graph with 25 links in the network, where each edge in the graph is present with a probability \( p = 0.2 \). Figure 7(d) shows the average queue length of the network as a function of time. Here, D-GMS is found to perform better than our algorithm in terms of delay, but the performance in terms of throughput is comparable among D-GMS, Q-CSMA and our algorithm as seen in Fig. 7(e). Finally, despite the many interfering links, our algorithm prevents links from monopolizing the network as seen in Fig. 7(f) thereby overcoming an inherent limitation of LQF.

VI. CONCLUSION

In this work, we proposed a novel Hamiltonian measure inspired by the Ising Model and associated it with the underlying interference graph of an ad hoc network. The Hamiltonian accounts for the queue length and waiting time at each link and is such that its minimization yields a schedule that results in low network delay, and increases aggregate throughput and fairness. It is constructed such that the change in Hamiltonian due to a spin flip only requires local information to compute. These characteristics allowed us to develop a scheduling algorithm that is distributed and easy to implement, requiring only local communication between links. Simulations show that the performance of our algorithm under different traffic patterns and network architectures is better than existing distributed algorithms such as D-GMS, Q-CSMA and I-CSMA in terms of aggregate throughput, network delay and fairness. This approach may be extended to other resource allocation problems - particularly scheduling for Cognitive Radio Networks. One approach is to include a priority measure in the network Hamiltonian based on whether the link is occupied by a Primary or Secondary user, and allow for adaptive transmission rates. This, and other resource allocation problems will be the focus of future work.

REFERENCES

[1] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, “Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling,” IEEE Trans. Inf. Theory, vol. 58, no. 10, pp. 6541–6555, Oct. 2012.

[2] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, “Decentralized traffic-aware scheduling for multi-hop low power lossy networks in the Internet of Things,” in Proc. IEEE 14th Int. Symp. “A World Wireless, Mobile Multimedia Networks” (WoWMoM), Jun. 2013, pp. 1–6.

[3] T. Chaturvedi, K. Li, C. Yuen, A. Sharma, L. Dai, and M. Zhang, “On the design of MAC protocol and transmission scheduling for Internet of Things,” in Proc. IEEE Region 10 Conf. (TENCON), Nov. 2016, pp. 2000–2003.

[4] K. Choi and S.-H. Chung, “Enhanced time-slotted channel hopping scheduling with quick setup time for industrial Internet of Things networks,” Int. J. Distrib. Sensor Netw., vol. 13, no. 6, Jun. 2017, Art. no. 155014771771362.

[5] M. V. Ngo, Q. D. La, D. Leong, T. Q. S. Quek, and H. Shin, “User behavior driven MAC scheduling for body sensor networks: A cross-layer approach,” IEEE Sensors J., vol. 19, no. 17, pp. 7755–7765, Sep. 2019.

[6] M. Zareei, C. Vargas-Rosas, R. V. Hernandez, and E. Arciplicita, “Efficient transmission power control for energy-harvesting cognitive radio sensor network,” in Proc. IEEE 30th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC Workshops), Sep. 2019, pp. 1–5.

[7] W. Osamy, A. A. El-sawy, and A. M. Khedr, “SATC: A simulated annealing based tree construction and scheduling algorithm for minimizing aggregation time in wireless sensor networks,” Wireless Pers. Commun., vol. 108, no. 2, pp. 921–938, Sep. 2019.

[8] S. Kumar and H. Kim, “Energy efficient scheduling in wireless sensor networks for periodic data gathering,” IEEE Access, vol. 7, pp. 11410–11426, 2019.

[9] M. Zareei, A. K. M. M. Islam, C. Vargas-Rosas, N. Mansoor, S. Goudarzi, and M. H. Rehmani, “Mobility-aware medium access control protocols for wireless sensor networks: A survey,” J. Netw. Comput. Appl., vol. 104, pp. 21–37, Feb. 2018.

[10] J. Mitola and G. Q. Maguire, “Cognitive radio: Making software radios more personal,” IEEE Pers. Commun., vol. 6, no. 4, pp. 13–18, 1999.

[11] Y. Aborahama and M. S. Hassan, “On the stochastic modeling of the holding time of SUs to PU channels in cognitive radio networks,” IEEE Trans. Cognit. Commun. Netw., vol. 6, no. 1, pp. 282–295, Mar. 2020.

[12] N. Mansoor, A. K. M. M. Islam, M. Zareei, and C. Vargas-Rosas, “RARE: A spectrum aware cross-layer MAC protocol for cognitive radio ad-hoc networks,” IEEE Access, vol. 6, pp. 22210–22227, 2018.

[13] M. Zareei, A. K. M. Muzahidul Islam, N. Mansoor, S. Baharun, E. M. Mohamed, and S. Saempei, “CMCS: A cross-layer mobility-aware MAC protocol for cognitive radio sensor networks,” EURASIP J. Wireless Commun. Netw., vol. 2016, no. 1, pp. 1–15, Dec. 2016.

[14] M. Zareei, A. Islam, S. Baharun, C. Vargas-Rosas, L. Arzpilicueta, and N. Mansoor, “Medium access control protocols for cognitive radio ad hoc networks: A survey,” Sensors, vol. 17, no. 9, p. 2136, Sep. 2017.

[15] A. Dimakis and J. Walrand, “Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits,” Adv. Appl. Probab., vol. 38, no. 02, pp. 505–521, Jun. 2006.

[16] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” in Proc. 29th IEEE Int. Conf. Decis. Control, Dec. 1990, pp. 2130–2132.

[17] S. T. Maguluri, B. Hajek, and R. Srikant, “The stability of longest-queue-first scheduling with variable packet sizes,” IEEE Trans. Autom. Control, vol. 59, no. 8, pp. 2295–2300, Aug. 2014.

[18] Q. Li and R. Negi, “Greedy maximal scheduling in wireless networks,” in Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Dec. 2010, pp. 1–5.

[19] N. Ii, 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 Trans. Netw., vol. 20, no. 3, pp. 825–836, Jun. 2012.

[20] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” IEEE/ACM Trans. Netw., vol. 18, no. 3, pp. 960–972, Jun. 2010.

[21] S. Rajagopalan, D. Shah, and J. Shin, “Network adiabatic theorem: An efficient randomized protocol for contention resolution,” ACM SIGMETRICS Perform. Eval. Rev., vol. 37, no. 1, pp. 133–144, Jun. 2009.

[22] J. Ghaderi and R. Srikant, “On the design of efficient CSMA algorithms for wireless networks,” in Proc. 49th IEEE Conf. Decis. Control (CDC), Dec. 2010, pp. 954–959.

[23] J. Ghaderi and R. Srikant, “The impact of access probabilities on the delay performance of Q-CSMA algorithms in wireless networks,” IEEE/ACM Trans. Netw., vol. 21, no. 4, pp. 1063–1075, Aug. 2013.

[24] D. Shah and J. Shin, “Delay optimal queue-based CSMA,” ACM SIGMETRICS Perform. Eval. Rev., vol. 38, no. 1, pp. 373–374, 2010.

[25] V. G. Subramanian and M. Alanyali, “Delay performance of CSMA in networks with bounded degree conflict graphs,” in Proc. IEEE Int. Symp. Inf. Theory Proc., Jul. 2011, pp. 2373–2377.
[26] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi, “From glauber dynamics to metropolis algorithm: Smaller delay in optimal CSMA,” in Proc. IEEE Int. Symp. Inf. Theory Proc., Jul. 2012, pp. 2681–2685.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” J. Chem. Phys., vol. 21, no. 6, pp. 1087–1092, Jun. 1953.

[28] Y. Wang and Y. Xia, “I-CSMA: A link-scheduling algorithm for wireless networks based on ising model,” IEEE Trans. Control Netw. Syst., vol. 5, no. 3, pp. 1038–1050, Sep. 2018.

[29] Y. Wang and Y. Xia, “A distributed CSMA algorithm for wireless networks based on ising model,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2013, pp. 170–176.

[30] X. Li, P. Tolmachev, M. Pauley, and J. H. Manton, “A distributed transmission scheduling algorithm for wireless networks based on the ising model,” in Proc. IEEE Stat. Signal Process. Workshop (SSP), Jun. 2018, pp. 6–10.

[31] A. Lucas, “Ising formulations of many NP problems,” Frontiers Phys., vol. 2, p. 5, Feb. 2014.

[32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671–680, 1983.

[33] B. M. McCoy and T. T. Wu, The Two-Dimensional Ising Model. Chelmsford, MA, USA: Courier Corporation, 2014.

[34] J. Kwak, C.-H. Lee, and D. Y. Eun, “A high-order Markov-chain-based scheduling algorithm for low delay in CSMA networks,” IEEE/ACM Trans. Netw., vol. 24, no. 4, pp. 2278–2290, Aug. 2016.

[35] B. Hajek, “Cooling schedules for optimal annealing,” Math. Oper. Res., vol. 13, no. 2, pp. 311–329, May 1988.

[36] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer system,” Digit. Equip. Corp., Hudson, MA, USA, Tech. Rep. DEC-TR-301, 1984.

TAHA AMEEN UR RAHMAN received the B.Sc. degree (summa cum laude) in electrical engineering and mathematics from the American University of Sharjah, Sharjah, United Arab Emirates, in 2019. He is currently a Research Associate with the American University of Sharjah. His research interests include communication networks, dynamical systems, and discrete mathematics.

MOHAMED S. HASSAN received the M.Sc. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 2000, and the Ph.D. degree in electrical and computer engineering from The University of Arizona, USA, in 2005. He is currently a Full Professor of electrical engineering with the American University of Sharjah. He was involved with multiple projects related to free space optical communications, electromagnetic shielding, demand response and smart grids, anti-static flooring, and fiber optic sensors for infrastructure health monitoring applications. His research interests include multimedia communications and networking, wireless communications, cognitive radios, resource allocation, performance evaluation of wired networks, and next generation wireless systems.

MAHMOUD H. ISMAIL (Senior Member, IEEE) received the B.Sc. degree (Hons.) in electronics and electrical communications engineering and the M.Sc. degree in communications engineering from Cairo University, Egypt, in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering from The University of Mississippi, Southaven, MS, USA, in 2006. From 2000 to 2002, he was a Research and Teaching Assistant with the Department of Electronics and Electrical Communications Engineering, Cairo University. From 2004 to 2006, he was a Research Assistant with the Center for Wireless Communications, The University of Mississippi. He was also a Systems Engineering Consultant with the Egypt Design Center, Newport Media Inc. (Microchip), Cairo, from 2006 to 2014. He is currently a Full Professor (on leave) with the Department of Electronics and Electrical Communications Engineering, Cairo University, and a Full Professor with the American University of Sharjah, Sharjah, United Arab Emirates. His research interests include wireless communications with emphasis on performance evaluation of next-generation wireless systems and communications over fading channels. He is a member of Sigma Xi and Phi Kappa Phi. He was a recipient of The University of Mississippi Summer Assistantship Award in 2004 and 2005, The University of Mississippi Dissertation Fellowship Award in 2006, The University of Mississippi Graduate Achievement Award in electrical engineering in 2006, and the Best Paper Award from the Tenth IEEE Symposium on Computers and Communications, La Manga del Mar Menor, Spain, in 2005. He served as a reviewer for several refereed journals and conferences.

* * *

VOLUME 8, 2020