# Spatial CSMA: A Distributed Scheduling Algorithm for the SIR Model with Time-varying Channels

Peruru Subrahmanya Swamy, Radha Krishna Ganti, Krishna Jagannathan

Department of Electrical Engineering, IIT Madras, Chennai, India 600036

{p.swamy, rganti, krishnaj}@ee.iitm.ac.in

*Abstract*—**Recent work has shown that adaptive CSMA algorithms can achieve throughput optimality. However, these adaptive CSMA algorithms assume a rather simplistic model for the wireless medium. Specifically, the interference is typically modelled by a conflict graph, and the channels are assumed to be static. In this work, we propose a distributed and adaptive CSMA algorithm under a more realistic signal-to-interference ratio (SIR) based interference model, with time-varying channels. We prove that our algorithm is throughput optimal under this generalized model. Further, we augment our proposed algorithm by using a parallel update technique. Numerical results show that our algorithm outperforms the conflict graph based algorithms, in terms of supportable throughput and the rate of convergence to steady-state.**

## I. INTRODUCTION

A central problem in wireless networks is the design of efficient link scheduling algorithms in the presence of interference. In the design of scheduling algorithms, there are three key performance metrics of interest. The first among them is the achievable *throughput region*. The throughput performance of a scheduling algorithm is characterized by the largest set of arrival rates under which the algorithm can stabilize the queues in the network. Secondly, the *average delay* incurred by the packets in the queue should be small. The third metric of interest is the computational and communication *complexity* involved in implementing the algorithm. Scheduling algorithms with low computational complexity and low communication overheads are preferable.

### A. Related Work

A large part of the existing literature on scheduling is based on the maximum weight scheduling algorithm [1], which is known to be throughput optimal under fairly general conditions. However, maximum weight scheduling generally requires solving an NP-hard problem during each scheduling instant, and is difficult to implement in practice. Further, it is not directly amenable to a distributed implementation. Several low complexity alternatives [2] have been proposed but they achieve only a fraction of the capacity region, and are hence not throughput optimal.

On the other hand, there are simple random access techniques such as Aloha, CSMA (Carrier Sense Multiple Access) which can be implemented in a distributed manner. A distributed algorithm was developed in [3] to adaptively choose the CSMA parameters so as to achieve throughput optimality. Central to this algorithm is the so called *Glauber dynamics*,

This work has been presented at NCC-2015, held at Mumbai, India.

which is a Monte Carlo Markov Chain sampling technique [4], [5]. Specifically, it is a Gibbs sampler [5] based algorithm.

A main shortcoming of the existing papers on adaptive CSMA is that the results are derived based on rather simplistic models for the wireless channels and the interference. Typical modelling assumptions used include:

a. *Conflict graph interference model*: The interference is modelled by a conflict graph or protocol model [3], where the transmissions from two links fail, if the links share an edge in the conflict graph. In reality however, the success or failure of a link depends on the aggregate interference from all the active links in the interference range. In other words, the complex nature of wireless interference is not adequately captured by a conflict graph. On the other hand, the SIR-based interference model can be used to overcome this limitation.

b. *Channel model*: It is assumed that the wireless channel is either to be static (*i.e.*, not time-varying), or that the instantaneous CSI (Channel state information) at each time slot is available for scheduling collision free transmissions. However, wireless channels are seldom static due to fading, and the availability of CSI at each transmitter is not necessarily realistic in an adhoc setting.

We present a brief summary of the assumptions made in the existing literature in the following table:

| Ref. | Interference model | Channel model | CSI | Throughput Optimality |
|---|---|---|---|---|
| [3], [6] | graph | static | - | ✓ |
| [7] | graph | varying | Inst. | ✓ |
| [8] | graph | varying | Inst. | |
| [9] | SIR | varying | Inst. | |
| [10] | SIR | varying | Stat. | |
| This work | SIR | varying | Stat. | ✓ |

- *Inst.* - Instantaneous channel gains are assumed to be known at each time slot.
- *Stat.* - Channel statistics (such as average channel gains or distribution) are assumed to be known.

A time-varying channel is considered between the transmitter of a link and its corresponding receiver in [7]. However, the channel gains between the interfering links are assumed to be static. In [8], time varying channels are considered among all the links, and the interference is modelled by a conflict graph. However, the algorithm [8] can support only a fraction of the achievable rate region. A SIR model is considered in [9] to a propose conservative algorithm that is suboptimal. An adaptive Aloha based algorithm is proposed in [10] under time-

varying channels. However the algorithm can only maximize some utility functions and is not throughput optimal.

### B. Our Contributions

In this work, we consider a single-hop wireless network and propose a distributed scheduling algorithm.

- We consider time-varying channels among all the links in the network. Further, the interference is modelled using the SIR model which is more realistic.
- A key contribution of this paper is in the design of a Gibbs sampler [5] based throughput optimal scheduling algorithm (*Algorithm 1*). In the algorithm we propose, each link only requires the average channel gains from its neighbouring links (defined later). In particular, instantaneous channel gains are not required, which makes our algorithm practical in a fast fading scenario, where the channel gains vary rapidly within a data slot.
- We augment *Algorithm 1*, which allows only single link updates, and propose *Algorithm 2*, which performs parallel link updates and converges faster.

The remainder of the paper is organised as follows. In Section II, the network model is described. In Section III, the spatial CSMA algorithm is presented and its throughput optimality is proved. Numerical results are presented in Section IV, and we conclude in Section V.

## II. NETWORK MODEL

We consider a single-hop ad-hoc wireless network, and model the links using a bipole model introduced in [10]. In a bipole model, each transmitter is associated with a receiver that is at a distance $R$ in some arbitrary direction. A transmitter and its corresponding receiver is referred to as a link. We assume that there are $N$ links in the network. We use the set $\mathcal{N}$ to denote all the links in the network. We assume that the time is slotted.

We assume that the link distance $R$ is much smaller than the distances of the transmitter and receiver to the other links. With this assumption, we can think of links as points in the Euclidean space (The results in this paper are not limited by this assumption. The assumption is taken to keep the expressions simple). Let $r_{ji}$ denote the distance between the links $i, j$. We consider a standard path-loss model $\|x\|^{-\alpha}, \alpha > 2$.
*Channel model:* The small-scale fading (power) between any pair of nodes is modeled by a unit power Rayleigh distribution and is assumed to be i.i.d across time and space. The channel gain between the transmitter of a link $i$ and the receiver in link $j$ is denoted by $h_{ji}$. Since $h_{ij}$ is Rayleigh distributed, $|h_{ij}|^2$ is exponentially distributed with unit mean.
*Interference model:* A receiver successfully receives the packet of the corresponding transmitter if the received SIR is above a pre-determined threshold $T$. We consider interference limited networks, where the impact of thermal noise is negligible as compared to interference. Suppose $\mathcal{M} \subset \mathcal{N}$, be the set of links that are transmitting in the current slot. The SIR of a link $i \in \mathcal{M}$ denoted by $\gamma_{i,\mathcal{M}}$ is given by,

$$\gamma_{i,\mathcal{M}} = \frac{|h_{ii}|^2 R^{-\alpha}}{I(\mathcal{M} \setminus \{i\})}.$$

Here $|h_{ii}|^2 R^{-\alpha}$ is the received power at the receiver in link $i$ from its intended transmitter and

$$I(\mathcal{M} \setminus \{i\}) = \sum_{j \in \mathcal{M} \setminus \{i\}} |h_{ij}|^2 r_{ij}^{-\alpha},$$

is the interference power from other concurrent transmissions.
*Queuing Dynamics:* Each link has a separate arrival process and maintains its own buffer. $[a_i]_{i=1}^N$ denote the arrival rates of the links, $[q_i(t)]_{i=1}^N$ denote the queue lengths of the links in time slot $t$.
*Assumptions on channel state information:* We assume that each link knows the distances to its neighbours (defined later), the path loss exponent and the SIR threshold $T$.

We now compute the probability that a transmission is successful in the presence of interference.

### A. Probability of successful link

The probability of success for a link $i \in \mathcal{M}$ denoted by $\mu_i(\mathcal{M})$ is

$$\begin{aligned}
\mu_i(\mathcal{M}) &= \mathbb{P}\left(\gamma_{i,\mathcal{M}} \geq T\right), \\
&= \mathbb{P}\left(|h_{ii}|^2 \geq R^\alpha T I(\mathcal{M} \setminus \{i\})\right), \\
&\stackrel{(a)}{=} \mathbb{E}_{\{h_{ij}\}} \exp\left(-TR^\alpha \sum_{j \in \mathcal{M} \setminus \{i\}} |h_{ij}|^2 r_{ij}^{-\alpha}\right), \\
&\stackrel{(b)}{=} \prod_{j \in \mathcal{M} \setminus \{i\}} \mathbb{E}_{h_{ij}} \exp\left(-R^\alpha T |h_{ij}|^2 r_{ij}^{-\alpha}\right), \\
&\stackrel{(c)}{=} \prod_{j \in \mathcal{M} \setminus \{i\}} \frac{1}{1 + \left(\frac{R}{r_{ij}}\right)^\alpha T}, \qquad \forall i \in \mathcal{M},
\end{aligned}$$

where $(a)$ and $(c)$ follow from the exponential distribution of $|h_{ii}|^2$, $|h_{ij}|^2$ and $(b)$ follows from the independence of the fading variables. Let $f(r_{ij}) := \frac{1}{1+\left(\frac{R}{r_{ij}}\right)^\alpha T}$. Then the probability of success can be written as,

$$\mu_i(\mathcal{M}) = \prod_{j \in \mathcal{M} \setminus \{i\}} f(r_{ij}), \quad \forall i \in \mathcal{M}. \tag{1}$$

For convenience, the probability of success is set to zero for the links that are not in the currently active set $\mathcal{M}$, *i.e.*, $\mu_i(\mathcal{M}) = 0, \quad \forall i \notin \mathcal{M}$. Note that (1) is calculated, assuming all the active links in the network can contribute to the interference of a receiver. However, from the studies on statistical distribution of co-channel interference, the aggregate interference from the links beyond a certain distance can be safely neglected [11], [12]. The radius beyond which the interference can be neglected is referred to as close-in radius, and is denoted by $R_I$. Hence for a link $i$, the interference from the active links outside a ball of radius $R_I$ around $i$ can be neglected. Let $\mathcal{N}_i$ denote the set of links that are potential interferers of link $i$, *i.e.*, the set of links within the ball of radius $R_I$ around link $i$. The links in $\mathcal{N}_i$ are referred to as the *neighbours* of link $i$. Thus, from (1) the probability of success is,

$$\mu_i(\mathcal{M}) = \prod_{j \in \mathcal{M}_i} f(r_{ij}), \quad \forall i \in \mathcal{M}, \tag{2}$$

where $\mathcal{M}_i := \mathcal{N}_i \cap \mathcal{M}$ is set of active links that are within the close-in radius of link $i$. Also note that, if none of the potential interferers of a link $i$ are active, then it succeeds with probability one, *i.e.*,

$$\mu_i(\mathcal{M}) = 1, \quad \text{if} \ \ \mathcal{M} \cap \mathcal{N}_i = \emptyset.$$

From (2), we can observe that the probability of success of a link depends only on the distances from its *active* neighbours $\mathcal{M}_i$. This allows for computation of $\mu_i(\mathcal{M})$ by a simple neighbour discovery algorithm [13].

We now characterize the capacity region in terms of the link success probabilities $\mu_i(\mathcal{M})$.

### B. Capacity Region

Every subset of the links $\mathcal{M} \subset \mathcal{N}$, is associated with a $N$-dimensional vector $\mu(\mathcal{M}) = [\mu_i(\mathcal{M})]_{i \in \mathcal{N}}$ whose $i$-th element correspond to the probability of success of the link $i$ (when $\mathcal{M}$ is the set of links that are transmitting). $\mu(\mathcal{M})$ can also be interpreted as the long-term rates that can be supported when the subset $\mathcal{M}$ is active. We refer to these vectors as rate vectors.

The capacity region of the network is the set of all the arrival rate vectors for which there exists a scheduling algorithm that can stabilize the queues. It is known that the capacity region is given by

$$\Lambda = \{a \in \mathbb{R}_+^N \mid \exists \epsilon > 0, \ a(1 + \epsilon) \in \mathcal{C}o(\mu)\},$$

where, $\mathcal{C}o(\mu)$ is the convex hull of $\{\mu(\mathcal{M})\}_{\mathcal{M} \subset \mathcal{N}}$.

An arrival rate vector $y \in \mathbb{R}^n$ is said to be feasible if $y \in \Lambda$. A scheduling algorithm is said to be *throughput optimal*, if the algorithm can stabilize the network for any feasible arrival rate. A maximum weight scheduling algorithm is known to be throughput optimal. In each time slot, the algorithm picks the schedule, $\mathcal{M}(t) = \arg\max\limits_{\mathcal{M} \subset \mathcal{N}} \sum\limits_{j \in \mathcal{M}} \mu_j(\mathcal{M}) q_j(t)$.

Some of the notations used so far, are summarized below.

| | |
|---|---|
| $\mathcal{N} -$ | *Set of all the links in the network* |
| $\mathcal{M}(t) -$ | *Set of links that are active in slot $t$* |
| $\mathcal{N}_i -$ | *Set of potential interferers of link $i$.* |
| $\mathcal{M}_i(t) -$ | *Set of active interferers of link $i$ in slot $t$.* |
| $\mu_i(\mathcal{M}) -$ | *Rate of link $i$ when the set of active links is $\mathcal{M}$.* |

### III. SPATIAL CSMA

In this Section, our distributed algorithm, *Spatial CSMA* is presented and its throughput optimality is proved. The key idea is to sample subsets (of links) so that sampled subsets provide a good approximation to the Maximum weight algorithm [3], [6]. Let $g(x)$ be a real valued function of queue length. The details of the function $g(x)$ are discussed subsequently.

---

*Algorithm1*: **Spatial CSMA**

---

**Intialization:** Each link $i \in \mathcal{N}$ pre-computes $f_{ij} := f(r_{ij})$ for all its neighbours $j \in \mathcal{N}_i$ .

**Control slot:**

- *Decision schedule-* A link $i \in \mathcal{N}$, is picked uniformly at random.

- *Neighbour discovery-* Each link $j \in \{i\} \cup \mathcal{N}_i$ executes a neighbour discovery [13] algorithm to compute the set of its active interferes in the previous slot, *i.e.*, $\mathcal{M}_j(t-1)$.

- *Inactive weights-* Each link $j \in \mathcal{N}_i$ computes $\mu_j(\mathcal{M}(t-1) \setminus \{i\})$ from (2) and subsequently computes the inactive weight

$$w_j^0 := g(q_j(t)) \, \mu_j(\mathcal{M}(t-1) \setminus \{i\}). \qquad (3)$$

- *Active weights-* Link $i$ obtains the inactive weights from its neighbours and computes the active weights as defined below.

$$w_j^1 := w_j^0 f_{ij}, \quad \forall j \in \mathcal{N}_i,$$
$$w_i^1 := g(q_i(t)) \, \mu_i(\mathcal{M}(t-1) \cup \{i\}).$$

- *Update Probability-* Link $i$ computes its update probability $p(t)$ as,

$$p(t) = \frac{\exp(w_i^1)}{\exp\left(\sum\limits_{j \in \mathcal{M}_i(t-1)} (w_j^0 - w_j^1)\right) + \exp(w_i^1)}. \qquad (4)$$

Link $i$ chooses to transmit with probability $p(t)$ and chooses not to transmit with probability $1 - p(t)$, *i.e.*,

$$\mathcal{M}(t) = \begin{cases} \mathcal{M}(t-1) \cup \{i\} & \text{w.p.} \quad p(t), \\ \mathcal{M}(t-1) \setminus \{i\} & \text{w.p.} \quad 1 - p(t). \end{cases}$$

**Data slot:** In the data slot, all the links $j \in \mathcal{M}(t)$ will transmit.

---

In *Algorithm1*, each time slot is divided into a control slot and a data slot. In the control slot, a link $i$ is chosen at random (the implementation of this step is discussed later), and only this link is allowed to change its status (on/off) in this time slot. All other links will retain their status of the previous time slot. Link $i$ and its neighbours execute a neighbour discovery algorithm to identify all their active neighbours. For example, the compressed neighbour discovery scheme [13] is a fast and efficient neighbour discovery algorithm which jointly detects all the active neighbours by allowing them to simultaneously report their identity.

All the neighbours of the link $i$, use their neighbourhood information $\mathcal{M}_j(t-1)$ computed in the previous step to calculate their rate vectors. Note that, all the links in $\mathcal{M}_j(t-1)$ retain their status except for the possible change of the status for link $i$. Hence, to account for this possible change of the status of link $i$, we define two sets of weights namely *inactive weights* and *active weights*. The contribution of interference from link $i$ is excluded for computing the inactive weights but included for computing the *active weights*. Link $i$ uses these weights to compute its update probability $p(t)$, and updates its status accordingly. In the data slot, all the active links transmit.

### A. Throughput Optimality

**Lemma 1.** *If the queue lengths are fixed at $q = [q_i]_{i=1}^N$, then Algorithm1 corresponds to a Glauber dynamics Markov chain on the subsets $\mathcal{M} \subset \mathcal{N}$ with a stationary distribution given by,*

$$\Pi(\mathcal{M}) = \frac{1}{Z} \exp\left(\sum\limits_{j \in \mathcal{M}} \mu_j(\mathcal{M}) g(q_j)\right), \quad \forall \mathcal{M} \subset \mathcal{N}, \quad (5)$$

*where $Z$ is the normalizing constant.*

   *Proof:* Proof can be found in Appendix-VI-A ∎

If the queue lengths were indeed fixed (say at $q$) as required in Lemma 1, *Algorithm1* provides a good approximation for the maximum weight scheduler [6]. This can be observed from (5), as the stationary distribution $\Pi$ on the set of subsets, places the largest mass on the set $\mathcal{M}$ that maximizes $\sum_{j \in \mathcal{M}} \mu_j(\mathcal{M}) g(q_j)$, which is precisely the max-weight scheduler except for $q_j$ being replaced by $g(q_j)$. However, this replacement can be justified if an appropriate function $g(x)$ is chosen [14].

Lemma 1 assumes that the queue lengths are fixed. However, the queue lengths are time-varying. Moreover, the time required for the Glauber dynamics to reach steady-state can be very long in general to assume that the queue lengths do not change. However, if appropriate slowly varying functions like $\log(0.1x), \log\log(x+e)$ are used as $g(x)$, it can be shown [6], [15] that *Algorithm1* does approximate the maximum weight scheduler in each time slot with a high probability and is hence throughput optimal.

**Lemma 2.** *If $g(x) = \log(0.1x)$ or $\log\log(x+e)$, the proposed spatial CSMA algorithm is throughput optimal.*

   *Proof:* Follows from our Lemma 1, and Theorem 1, Proposition 2 in [6]. ∎

   *Remarks:* While the techniques used are standard, the key contribution of this paper is the application of these techniques to design a throughput optimal scheduling algorithm for the SIR model with time-varying channels.

Although *Algorithm1* is proposed for a Rayleigh fading model, it can be easily extended to other fading models without any additional effort.

In [7], CSMA algorithm is considered on a conflict graph with time-varying link capacities. The authors of [7] show that the back-off parameter should have a exponential form of the channel gain. They obtain this by solving a maximum entropy problem. However, as we see from Lemma 1, the exponential form follows naturally from the max-weight formulation.

*B. Parallel Updates*

In the control slot of *Algorithm1*, it is assumed that a link (decision schedule) can be randomly picked at each time slot to update its status. However, the algorithm does not explicitly describe how to implement that step. Moreover, only one link is allowed to update its status in a given time slot. In *Algorithm2*, we relax the limitation of single-update, and provide a distributed algorithm to pick a decision schedule. The limitation of single-update can be relaxed by considering a block (parallel update) Gibbs sampler based algorithm, which allows for parallel updates and also converges faster. However, to ensure a distributed implementation of the (parallel update) Gibbs sampler, the set of links that can do parallel updates has to satisfy the following constraint. (See Lemma 3 for a formal proof.)
*If link $i$ updates its status in a given slot, all the links whose current status information is being used in the computation of the update probability $p(t)$ of link $i$, cannot update in the same slot.*

A set of links, which satisfy the above constraint is referred to as a decision schedule. The formal definition is as follows:

**Definition 1.** *Decision Schedule*
*A set of links $\mathcal{D} \subset \mathcal{N}$, is said to be a decision schedule if,*

$$\mathcal{N}_i \cup \left( \bigcup_{j \in \mathcal{N}_i} \mathcal{N}_j \setminus \{i\} \right) \subset \mathcal{N} \setminus \mathcal{D}, \qquad \forall i \in \mathcal{D}. \quad (6)$$

The intuition for this definition is as follows. From (4), one can observe that the update probability $p(t)$ of a link $i$, depends only on the weights of active links in $\mathcal{N}_i$. Further, the weight of each link $j \in \mathcal{N}_i$, depends on the the status of its neighbours $\mathcal{N}_j$. (See the computation of incative weights in *Algorithm1*.) Hence, the required constraint translates to (6).

*Remark:* This constraint is only on the set of links that can do parallel updates in a given slot. However, there are no hard constraints on the set of links than can transmit in a given slot. This is a key difference of this model compared to conflict graph based model.

Generating a decision schedule $\mathcal{D}$ can be done in two steps.
*Step 1:* Generate a subset of links $\mathcal{S}$, such that no two links in $\mathcal{S}$ are within the close-in radius of each other.
*Step 2:* Initialize $\mathcal{D}$ to $\mathcal{S}$, and update $\mathcal{D}$ by removing some links from $\mathcal{D}$ as follows. Each link $k \notin \mathcal{S}$ checks if any of its neighbours are in $\mathcal{S}$. If more than one of its neighbours are present in $\mathcal{S}$, then the neighbours are removed from $\mathcal{D}$.

The first step ensures $\mathcal{N}_i \subset \mathcal{N} \setminus \mathcal{D}$, while the second step ensures $\left( \bigcup_{j \in \mathcal{N}_i} \mathcal{N}_j \setminus \{i\} \right) \subset \mathcal{N} \setminus \mathcal{D}$ so that (6) is satisfied.

In [6], a distributed algorithm is suggested for generating the subset $\mathcal{S}$ (*step 1*). We extend that algorithm to generate the subset $\mathcal{D}$. For the sake of completeness, we present the *step 1* from [6]. In *Algorithm2*, the control slot is divided into $(W+2)$ control mini-slots for some $W \geq 2$. (This bound on $W$ is to ensure that each link has a non zero probability of being selected in the decision schedule.) In each time slot, all the links in the network will execute *Algorithm2* independently.
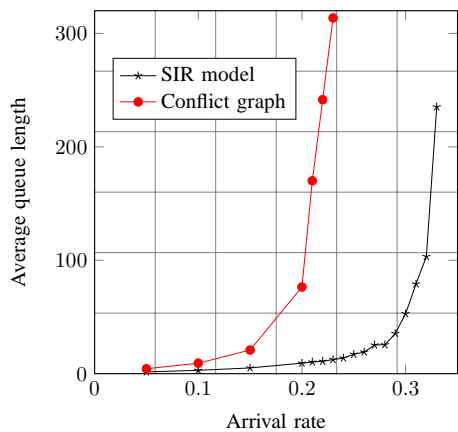
---

*Algorithm2*: **Decision Schedule Algorithm (at link $i$)**
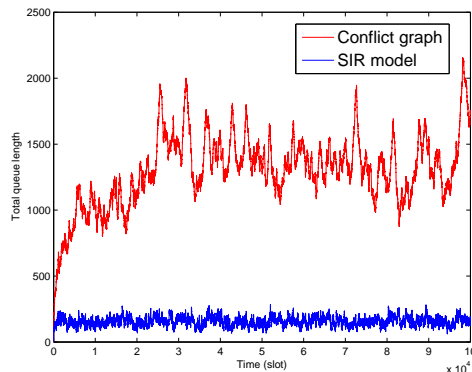
---

*Step1:* **Generating $\mathcal{S}$**
- Link $i$ selects a random (integer) backoff time $T_i$ uniformly in $[0, W-1]$ and waits for $T_i$ control mini-slots.
- If link $i$ hears an INTENT message from a link in $\mathcal{N}_i$ before the $(T_i + 1)$th control mini-slot, $i$ will not be included in $\mathcal{S}$ and will not transmit an INTENT message.
- If link $i$ does not hear an INTENT message from any link in $\mathcal{N}_i$, before the $(T_i + 1)$th control mini-slot, it will send (broadcast) an INTENT message to all links in $\mathcal{N}_i$ at the beginning of the $(T_i + 1)$th control mini-slot.
  - If there is a collision (*i.e.*, if there is another link in $\mathcal{N}_i$ transmitting an INTENT message in the same mini-slot), link $i$ will not be included in $\mathcal{S}$.
  - If there is no collision, link $i$ will be included in $\mathcal{S}$.

*Step2:* **Generating $\mathcal{D}$ from $\mathcal{S}$**
- If link $i \in \mathcal{S}$, it sends (broadcasts) an INTENT message to all links in $\mathcal{N}_i$ at the beginning of $(W+1)$th control mini-slot.

(a) Throughput Comparison



(b) Convergence rate comparison

Fig. 1: Numerical Results

- If link $i \notin \mathcal{S}$, it senses the channel for a possible collision (*i.e.*, if more than one of its neighbours are in $\mathcal{S}$, then all of them send INTENT messages which result in a collision) in $(W+1)$th mini-slot.
    - If there is a collision, link $i$ will broadcast a DETECT message to all its neighbours in $(W+2)$th control mini-slot.
- If link $i \in \mathcal{S}$ and it doesn't hear a DETECT message in $(W+2)$th control mini-slot, it will be included in $\mathcal{D}$.

**Lemma 3.** *If all the links in a decision schedule $\mathcal{D}$ selected from* Algorithm2*, simultaneously update their schedules using* Algorithm1*, then the stationary distribution of the resulting (parallel update) Glauber dynamics is given by* (5).

*Proof:* Proof can be found in Appendix VI-B. ∎

## IV. NUMERICAL RESULTS

In this Section, we evaluate the performance of the spatial CSMA algorithm. The results are compared to conflict graph based CSMA. This comparison requires the generation of an equivalent conflict graph for a given set of locations (of the links) as described below.

*Construction of conflict graph:* In a conflict graph based interference model, each link in the network is represented by a vertex in a graph. Two vertices are connected by an edge, if their concurrent transmissions can possibly end up in a collision. Concurrent transmissions from any two links that are with in the close-in radius of each other can be unsuccessful (depending on the channel conditions). Hence, a pair of vertices are connected by an edge if they are in the close-in distance of each other.

*A. Simulation settings*

We consider a two dimensional square plane with side length 13. A homogeneous Poisson point process of density 0.1 is generated. The generated points correspond to the locations of the transmitters. Each transmitter has its receiver at a distance of 0.25 in a random direction. The path loss exponent $\alpha$ is set to 2.5, the close-in radius $R_I$ is set to 4, and

the threshold SIR is set to 17 dB. The function $g(x)$ (used in *Algorithm1*) is set to $\log(0.1x)$. In each time slot, the channel gains corresponding to unit power Rayleigh distribution are generated.

*B. Throughput performance*

In Figure 1a, we illustrate the throughput performance of *Algorithm1*, by plotting the average queue lengths to see for which arrival rates the system is stable. If the algorithm cannot stabilize the network for a given arrival rate, the queue length blows up. We consider homogeneous arrival rates for all the links. It can be observed that the SIR based algorithm supports a larger set of arrival rates compared to the graph based algorithm. This is because, in a conflict graph model, concurrent transmissions from two neighbouring links are strictly prohibited irrespective of the exact distance between them. However, in SIR model, the links make a better choice by considering the exact distances from its neighbouring links (thereby taking into account the severity of interference) while computing the update probabilities.

*C. Convergence rate*

In Figure 1b, we compare the convergence rate of the spatial CSMA with the graph model by plotting the total queue evolution as a function of time. We consider a homogeneous arrival rate of 0.2 which is in the stable region of both these models. It can be observed that the queue reaches steady state much faster in the SIR model as compared to the conflict graph model.

## V. CONCLUDING REMARKS

In this paper, we considered the SIR model with time-varying channels, and proposed a distributed CSMA algorithm. We further proved that the proposed algorithm is throughput optimal. We also proposed a parallel update algorithm with a better convergence rate. Using simulations, we observed that the SIR model supports a larger set of arrival rates, and converges much faster than the conflict graph based model.

## VI. Appendix

### A. Proof of Lemma - 1

The Glauber dynamics (section 3.3.2 of [4]) corresponding to the distribution $\Pi$, is a reversible Markov chain with state space $\{\mathcal{M} \mid \mathcal{M} \subset \mathcal{N}\}$, and stationary distribution $\Pi$. The transition probabilities of that Markov chain are described here. From a given state $\mathcal{M}(t-1)$, the chain moves to a new state as follows. A link $i$ is chosen uniformly at random from $\mathcal{N}$ and a new state is chosen according to the measure $\Pi$ conditioned on the set of states in which status (on/off state) of all the links other than link $i$ remain the same as in $\mathcal{M}(t-1)$. In other words, the chain can only move to $\mathcal{M}(t-1) \cup \{i\}$ or $\mathcal{M}(t-1) \setminus \{i\}$ according to the conditional distribution

$$\Pi(\mathcal{M}(t) \mid \mathcal{M}(t) \in \{\mathcal{M}(t-1) \cup \{i\}, \mathcal{M}(t-1) \setminus \{i\}\}). \quad (7)$$

From (5), it can be easily verified that the update probability $p(t)$ in *Algorithm1* correponds to the conditional distribution in (7). Hence, *Algorithm1* corresponds to a Glauber dynamics Markov chain with stationary distribution given by (5).

### B. Proof of Lemma - 3

Let us represent a state $\mathcal{M}$ of the Glauber dynamics Markov chain with a $N-$ dimensional binary vector $\sigma$ whose elements are given by $\sigma_i = \mathbf{1}\{i \in \mathcal{M}\}, \quad i \in \mathcal{N}$. Thus, the equivalent state space of the Markov chain is $\{\sigma \mid \sigma \in \{0,1\}^N\}$. $\sigma(t)$ can be shown as a Random field (chapter 7 in [5]) on $\mathcal{N}$ with $\{\sigma_i(t)\}_{i \in \mathcal{N}}$ being the underlying random variables. We construct an undirected graph $G(V, E)$ with $V = \mathcal{N}$ and edges given by the following definition of the neighbourhood on $G$.

$$N_G(i) := \left( \bigcup_{k \in \mathcal{N}_i} \mathcal{N}_k \setminus \{i\} \right) \cup \mathcal{N}_i,$$

where, $N_G(i)$ is the set of neighbouring vertices of vertex $i$ in the graph $G$. Let us denote, $\sigma(\mathcal{S}) = [\sigma_i]_{i \in \mathcal{S}}$ for $\mathcal{S} \subset \mathcal{N}$. Observe that the conditional distribution, $\sigma_i(t)$ given $\sigma(\mathcal{N} \setminus \{i\})(t)$ corresponds to the distribution in (7) (which corresponds to the update probability $p(t)$ as discussed in the proof of lemma 1). Hence,

$$\mathbb{P}(\sigma_i(t) = 1 \mid \sigma(\mathcal{N} \setminus \{i\})(t)) = p(t). \quad (8)$$

From (4), one can observe that the update probability $p(t)$, depends only on the weights of active links in $\mathcal{N}_i$. Further, the weight of each link $k \in \mathcal{N}_i$ given by (3), depends on the the status of its neighbours $\mathcal{N}_k$. In other words, the update probability $p(t)$ of link $i$, depends only on the status of the links in $N_G(i)$. Using this observation in (8) gives,

$$\mathbb{P}\left( \sigma_i(t) \mid \sigma(\mathcal{N} \setminus \{i\})(t) \right) = \mathbb{P}\left( \sigma_i(t) \mid \sigma(N_G(i))(t) \right). \quad (9)$$

In other words, the random variable $\sigma_i(t)$ is independent of all other random variables given the random variables corresponding to its neighbours in $G$, *i.e.*, $\sigma(t)$ is a Markov random field [5] with respect to $G$.

In a parallel update Glauber dynamics corresponding to $\Pi$, if $\mathcal{D}$ is the set of links that are selected for update, then the update rule is specified by the following joint law [5],

$$\mathbb{P}(\sigma(\mathcal{D})(t) \mid \sigma(\mathcal{N} \setminus \mathcal{D})(t)) = \Pi(\sigma(t) \mid \sigma(\mathcal{N} \setminus \mathcal{D})(t)). \quad (10)$$

From (6), $i \in \mathcal{D} \Rightarrow N_G(i) \subset \mathcal{N} \setminus \mathcal{D}$. Using this property along with (9), we can write (10) as a product of single-site update probabilities as follows.

$$\mathbb{P}(\sigma(\mathcal{D})(t) \mid \sigma(\mathcal{N} \setminus \mathcal{D})(t)) = \prod_{i \in \mathcal{D}(t)} \mathbb{P}\left( \sigma_i(t) \mid \sigma(N_G(i))(t) \right).$$

In other words, we can use the same update rule as in *Algorithm1* for all the links in $\mathcal{D}(t)$ simultaneously and still converge to the required stationary distribution given by (5).

## References

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

[2] 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 (TON)*, vol. 17, no. 4, pp. 1132–1145, 2009.

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

[4] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Soc., 2009.

[5] P. Bremaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. springer, 1999, vol. 31.

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

[7] S.-Y. Yun, J. Shin, and Y. Yi, "CSMA over time-varying channels: optimality, uniqueness and limited backoff rate," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2013, pp. 137–146.

[8] C. Joo, X. Lin, J. Ryu, and N. B. Shroff, "Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2013, pp. 89–98.

[9] D. Qian, D. Zheng, J. Zhang, and N. Shroff, "CSMA-based distributed scheduling in multi-hop MIMO networks under SINR model," in *IN-FOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[10] F. Baccelli and C. Singh, "Adaptive spatial Aloha, fairness and stochastic geometry," in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), 2013 11th International Symposium on*. IEEE, 2013, pp. 7–14.

[11] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 2–13.

[12] P. C. Pinto and M. Z. Win, "Communication in a Poisson field of interferers," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 432–437.

[13] J. Luo and D. Guo, "Compressed neighbor discovery for wireless ad hoc networks: the Rayleigh fading case," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 308–313.

[14] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 2, pp. 411–424, 2005.

[15] 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. ACM, 2009, pp. 133–144.