Low Complexity Stable Link Scheduling for Maximizing Throughput in Multihop Wireless Networks

Xiang-Yang Li [*] XiaoBing Wu [†] XuFei Mao ShaoJie Tang YanWei Wu Ping Xu XiaoHua Xu

ABSTRACT

This paper presents novel distributed algorithms for scheduling transmissions in multi-hop wireless networks. Our algorithms generate new schedules in a distributed manner via simple local changes to existing schedules. Two classes of algorithms are designed: one assumes known location information of all wireless nodes, and the other does not. Both classes of algorithms are parameterized by an integer k (called algorithm-k). We show that algorithm-k of our class that uses geometry location achieves $(1-2/k)^2$ of the capacity region, for every $k \geq 3$; algorithm-k of our class that did not use geometry location achieves $1/\rho$ of the capacity region, for every $k \geq 3$ and a constant ρ depending on k. Our algorithms have small worst-case overheads. Both classes of algorithms can generate a new schedule by requiring communications within $\Theta(k)$ hops for every node. The parameter k explicitly captures some tradeoffs between control overhead and the throughput performance of any scheduler. Additionally, the class of algorithms with known geometry location of nodes can find a new schedule in time $\Theta(k^2\Delta)$, where Δ is the minimum mini-time-slots such that each of the n nodes can communicate with its neighbors once, which is clearly the minimum time-slots required by *any* scheduling algorithm.

1. INTRODUCTION

Link scheduling that maximizes the network throughput has been extensively studied in the literature. Recently, a number of scheduling algorithms with theoretical performance guarantees [2, 10, 14, 23, 25, 31] and/or of practical efficiency have been added to an already rich body knowledge [3,12,18,24,27,28] of general scheduling problems. Scheduling algorithms in networking community often want to maximize the network throughput [25, 31], or achieve a certain fairness among all requesting network flows [16, 17, 29, 30]. The task of wireless scheduling (or medium access control) is challenging due to the simultaneous presence of two characteristics: interference between simultaneous transmissions, and the need for practical distributed implementation with small communication overhead and time complexity. It will be more challenging to guarantee the performance of scheduler when the traffic flows will arrive based on certain random process, not in a constant fixed known data rate. In this paper, we will focus on the scenario when data packets will arrive randomly (with a bounded variance). Unlike the wired networks, the signal interference casts significant effect on the fundamental limit on the data throughput that any scheduling algorithms (centralized or distributed) can achieve. It is well-known that a number of scheduling problems (*e.g.*, maximum throughput scheduling) become NP-hard when considering wireless interference, while their counter-parts are solvable in polynomial time for wired networks. Thus, the scheduling algorithms for wireless networks (even the benchmark performances obtained by centralized scheduling approaches) often rely on heuristics that approximately optimize the throughput.

A linear time centralized scheduling policy that achieves the maximum attainable throughput region has been presented by Tassiulas and Ephremides [28] and Tassiulas [27]. For arbitrary interference models, it is well-known [4] that maximum throughput scheduling problem is NP-complete and not approximable within $m^{\frac{1}{3}-\epsilon}$ for any arbitrarily small $\epsilon > 0$ for a network of m links, unless NP=ZPP. In [26], it is shown that one can obtain centralized $(1+\epsilon)$-approximation algorithms for max-throughput scheduling under all K-hop interference models in the case of geometric graphs. They also showed that the *maximal* scheduling policy (that chooses a maximal independent set, instead of maximum weighted independent set of links) will achieve an efficiency ratio[1] $\frac{1}{49}$ for networks modeled as unit disk graphs under k-hop interference model. However, the lack of central control in wireless networks calls for the design of distributed scheduling algorithms. Such distributed algorithms should achieve the maximum throughput or at least a guaranteed fraction of the maximum throughput, on the other hand, it will incur only a small communication overhead.

The distributed scheduling algorithms, to the best of our knowledge, are mainly based on the *pick and compare* approach that was first developed and analyzed in [27]. It was shown in [27] we can provide maximum possible throughput if we can pick an *optimum* solution for the current time-slot with at least a constant probability. In this approach, a set of links satisfying the interference constraints is picked at each time slot (with a constant probability it has maximum weight) and its weight (typically the total queue size) is compared with the weight of the set of links chosen to transmit during the previous time-slot; and the one with maximum weight is chosen for transmission during the current time-slot. However, there are two challenges in using *pick and compare* approach: 1) it is often difficult to find an optimum solution with at least a constant

[*]Dept. of Comp. Sci., Illinois Inst. of Tech.

[†]Dept. of Comp. Sci., NanJing University, NanJing, China

[1]The efficiency ratio is defined as the largest number γ such that any rate vector $\lambda \in \gamma\mathcal{C}$ can be stablized. Here $\mathcal{C}$ is the capacity region of the network.

probability, and 2) comparing the weights of two given schedulings (*i.e.*, matchings when consider primary-interference) often requires network-wide computation and message exchange and may incur substantial overhead in terms of time, even under the simplest of interference models, such as the 1-hop interference model.

Recently, a number of *distributed* scheduling algorithms [4, 10, 14,20,22,25] for multihop wireless networks have been proposed in the literature for (approximately) maximizing the attainable throughput. Results in [10,20,25] only considered the primary-interference model. Sharma *et al.* [4] proposed maximal matching policy for 1-hop (*i.e.*, primary-interference interference model) or 2-hop interference model (without using pick and comparing approach) that runs in time $\log^3 |V|$ and achieves $\frac{1}{\alpha^1(G)}$ of the maximum throughput. Here $\alpha^1(G)$ is the 1-hop independence number of the interference graph. Lin [14] proposed a constant-overhead probabilistic scheduling algorithm (based on contention and random back-off) that achieves $\frac{1}{3} - \epsilon$ of the maximum throughput for primary-interference model and $\frac{1}{1+\Delta} - \epsilon$ of the maximum throughput for 2-hop interference model. Here Δ is the maximum number of node degree in the communication graph G. A modified and enhanced scheduling policy was proposed in [7] that guarantees $\frac{1}{2} - \epsilon$ of the maximum throughput for primary-interference model and efficiency ratio close to $\frac{1}{1+\Delta}$ for 2-hop interference model. The best distributed scheduling results for primary-interference model so far is [25] that proposed a distributed scheduling with control overhead $O(k)$ that achieves $k/(k+2)$ efficiency ratio.

Our results: The main contributions of this paper are as follows.

It is known that pick and compare scheme provides 100% throughput guarantee if we can pick an optimum scheduling with at least a constant probability, but it may have a very large time complexity. A question was posed by Sharma *et al.* in [4]: "Can one design a low complexity scheduling scheme with close to 100% throughput guarantee?". In this paper, we firmly present a positive answer to this question for reasonable interference models. Our results can also be extended to solve more general optimization problems, not necessarily the queue [15]. To the best of our knowledge, our algorithms are the first in the literature such that *any* arbitrary fraction of the capacity region can be achieved with constant overhead for more sophisticated wireless interference models, while the previous distributed scheduling methods either assume a simple primary-interference model to get the same efficiency ratio, or can only achieve efficiency ratio at most $1/(1 + \Delta)$ for 2-hop interference models. Our results also can be extended to the physical interference model where a reception is successful only if the SINR is at least a certain threshold β_0, under a reasonable signal attenuation model, however, the efficiency ratio is only a constant in $(0, 1)$.

To present our design approach, we first present efficient centralized scheduling methods that guarantees an efficiency ratio of $(1 - \frac{2}{k})^2$ in polynomial time. Our centralized methods work for a variety of interference models where nodes could have different interference radii. We then present an efficient *stable* distributed scheduling algorithm that achieves the same efficiency ratio, while the control overhead is only within $\Theta(k)$-hops. A control message is relayed by at most $O(k)$ nodes. The parameter k explicitly captures some tradeoffs between control overhead and the throughput performance of any scheduler. Two different classes of distributed methods are presented: one assumes the availability of geometry location of nodes (which guarantees a constant time complexity for distributed scheduling), and the other only assumes that the interference graph is growth-bounded (which is true for all interference models when interference ranges of nodes are within a constant factor of each other). Both classes of algorithms are parameterized by an integer k (called algorithm-k). We show that algorithm-k of our class that uses geometry location achieves $(1-2/k)^2$ of the capacity region, for every $k \geq 3$; algorithm-k of our class that did not use geometry location achieves $1/\rho$ of the capacity region, for every $k \geq 3$ and a constant ρ depending on k. The class of algorithms with known geometry location of nodes can find a new schedule in time $\Theta(k^2\Delta)$, where Δ is the minimum mini-time-slots such that each of the n nodes can communicate with its neighbors once, which is clearly the minimum time-slots required by *any* scheduling algorithm.

The rest of the paper is structured as follows. In Section 2 we present the wireless network model, and define the problems to be studied. We present our novel low complexity centralized and distributed scheduling algorithms in Section 3 and Section 4 respectively. Their properties and performances are analyzed. The simulation studies of our protocols are presented in Section 5. In Section 6, we briefly review the related works. We conclude our paper with future works in Section 7.

2. MODELS AND DEFINITIONS

2.1 Communication Network Model

A multihop wireless ad hoc network is modeled by a graph $G = (V, E)$, where the vertices $V = \{v_1, v_2, \cdots, v_n\}$ represent the set of $n = |V|$ wireless devices in the network, and a directed link $(u, v) \in E$ iff these two wireless devices u and v can communicate with each other directly without relaying. Node v is receiver and u is the sender of link (u, v). Node v is also called the (communication) neighbor of node u. We always use $e_{i,j}$ to denote link (v_i, v_j) hereafter. We assume that each node v_i has a fixed transmission radius T_i, and a fixed interference range R_i. For each node v_i, we assume that $R_i > (1 + \theta)T_i$ for a constant $\theta > 0$ ($\theta \geq 1$ in practice). We assume that the set V of communication terminals are deployed in a plane. Each wireless terminal is only equipped with *single* radio interface.

2.2 Interference Models

To schedule two links at the same time slot, we must ensure that the schedule will avoid interference. Previous studies on stable link scheduling mainly focused on *primary interference* model, in which no node can receive and send packets simultaneously. In addition to these interference, several different models have been used to model the interference. We briefly review the models we use in this paper.

Transmitter Interference Model (TIM) [32]: In this model, when the sources of two transmissions are at least a distance R away, the transmissions can be scheduled simultaneously.

Fixed Power Protocol Interference Model (fPrIM) [31]: We assume that, each node v_i, in addition to have a fixed transmission range T_i, has an *interference range* R_i such that any node v_j will be interfered by the signal from v_i if $\|v_i - v_j\| \leq R_i$ and node v_j is *not* the intended receiver of the transmission by v_i.

RTS/CTS Model: Each node v_i is associated with a range R_i. For each link (v_p, v_q), except v_p and v_q, no other nodes inside $D(v_p, R_p) \bigcup D(v_q, R_q)$ can transmit or receive simultaneously. Here $D(v, a)$ is a disk centered at v with radius a.

There are also other interference models, *e.g.*, K-hop interference model [26] (where two links interfere with each other if the hop-distance between them is at most $K - 1$), and physical interference model (where the SINR at the receiver must be at least a certain threshold). In this paper, we mainly focus on link scheduling under TIM, fPrIM and RTS/CTS models.

Assume that the communication links in wireless networks are predetermined. Given a communication graph $G = (V, E)$, we use *conflict graph* (*e.g.*, [6]) F_G to represent the interference in G. Each vertex (denoted by $e_{i,j}$) of F_G corresponds to a directed link (v_i, v_j) in the communication graph G. There is an *edge* between vertex $e_{i,j}$ and vertex $e_{p,q}$ in F_G iff $e_{i,j}$ conflicts with $e_{p,q}$ due to interference. Recall that whether two links conflict depends on the interference model used underneath, *e.g.*, fPrIM model or RTS/CTS model. Thus, for a given communication graph G, the interference graph F_G may be different.

2.3 Traffic Model and Scheduling

We now describe the standard model for link (or node) scheduling in the presence of wireless interference. As in the literature, we assume that time is slotted and synchronized among all wireless devices. Wireless nodes communicate with other nodes in the form of packets, whose size is normalized to one unit such that each packet can be communicated in one time-slot. For simplicity, we assume that all traffic is single-hop. Notice that most results can be extended to multihop traffic as done in [15, 28]. For each link $e \in E$, let $\mathbf{A}_t(e)$ be the number of new packets arrived for transmission over link e. Let $\mathbf{A}_t$ be the vector of all arrivals at time slot t. The arrival process $\mathbf{A}_t$ is assumed to be independent and identically distributed across time (it may be correlated across links), with an average arrival rate vector $a = E[\mathbf{A}_t]$ and bounded second moment $E[\mathbf{A}_t'\mathbf{A}_t] < \infty$. The packets $\mathbf{A}_t(e)$ can be transmitted at time slot t or later by link e. When the packets cannot be served immediately, they are put into the queue of link e. Let $Q_t(e)$ be the set of packets queued at link e at time t and $q_t(e)$ be the queue length. Let q_t be the vector of queue lengths at time t. Here we assume that there is no priori upper bound on the maximum queue size, thus there are delays for packets but no packet drops.

A link scheduling algorithm is to decide which links to be *active* and which links to be *inactive* for each time slot t. When a link e is active at time slot t, it can transmit exactly one packet out of its queue at time t. We will use the binary vector I_t of length $|E|$ to denote the set of active links at time t, with the convention that $I_t(e) = 1$ iff link e is active and has a positive queue at time t. The queue length vector thus evolves as $q_{t+1} = q_t + \mathbf{A}_{t+1} - I_t$. We say the system is *stable* if the total queue lengths of all links remain finite almost surely, *i.e.*, $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} Z(q_t, B) \to 0$ almost surely as $B \to \infty$. Here $Z(q_t, B) = 1$ if $\sum_{e\in E} q_t(e) \geq B$, and $Z(q_t, B) = 0$ otherwise.

A link scheduling algorithm for wireless networks is *valid* (or feasible) if the set of active links at any time slot does not cause any interference among all these active links. Let $\mathcal{I} = \{I^1, I^2, \cdots, I^{|\mathcal{I}|}\}$ be the set of all valid link schedulings (that could be exponential of the number of links $|E|$). The *capacity region* $\mathcal{C}$ (a point in the capacity region is a vector a of the mean arrival rate $E(\mathbf{A}_t)$) of the network G is the strict convex closure of all feasible schedulings $\mathcal{I}$: $a \in \mathcal{C}$ if and only if there exist non-negative numbers $\lambda_1, \lambda_1, \cdots, \lambda_{|\mathcal{I}|}$, such that

$$a = \sum_{m=1}^{|\mathcal{I}|} \lambda_m I^m, \qquad \text{and} \qquad \sum_{m=1}^{|\mathcal{I}|} \lambda_m < 1.$$

$\mathcal{C}$ has also been referred to in the literature as "stability region", and "100% throughput region". A scheduling policy is *throughput-optimal* if it can achieve the optimal capacity region $\mathcal{C}$. The *efficiency ratio* of a (possibly sub-optimal) scheduling policy is the largest number γ such that the scheduling policy can stabilize the system under any load vector $a \in \gamma\cdot\mathcal{C}$. By definition, a throughput-optimal scheduling policy has an efficiency ratio of 1.

Unlike recent scheduling algorithms in [10, 20, 25] that consider only *primary interference*, in this paper, we consider more practical interference models such as fPrIM and RTS/CTS model. Observe that under primary interference model, link scheduling only needs to ensure that any node communicates with at most one other node at any time slot. Thus, *any* matching in graph G is a valid link scheduling under primary interference model. Finding a matching that maximizes certain weight (*e.g.*, the total queue lengths of selected links to maximize throughput [25]) can be solved in polynomial time. The optimal scheduling policy is equivalent to computing a maximum weighted independent set (MWIS) of links in the conflict graph, where the weight of a link is its queue size, and a set $S \subseteq E$ of links are *independent* if they can be active simultaneously without interference. For interference models considered in this paper, it is well-known that finding MWIS is NP-hard [31]. Thus, we will rely on approximation algorithms. A scheduling policy S_γ is called *imperfect scheduling policy with ratio* $0 < \gamma \leq 1$ (see [15] for more discussions) if at *each* time-slot t, it will compute a schedule $I_t \in \mathcal{I}$ such that

$$I_t \cdot q_t \geq \gamma I_t^{\mathrm{OPT}} \cdot q_t, \text{ where } I_t^{\mathrm{OPT}} = \mathrm{argmax}_{I\in\mathcal{I}}(I \cdot q_t)$$

The following two propositions will be the foundation for studying the stability of our scheduling algorithms.

PROPOSITION 1. *[15] Fix $\gamma \in (0, 1]$. If the user rates, **a**, lie strictly inside $\gamma \cdot \mathcal{C}$ (i.e., **a** lies in the interior of $\gamma \cdot \mathcal{C}$), then any imperfect scheduling policy S_γ can stabilize the system.*

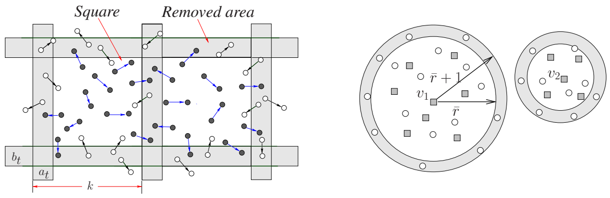
Observe that scheduling policy S_γ must find a γ-approximation scheduling at every time slot t. This requirement may be too strong to be satisfied by certain distributed algorithms. When we can only find the γ-approximation with a certain constant positive probability, the following proposition was proved.

PROPOSITION 2. *[25] Given any $\gamma \in (0, 1]$, suppose that an algorithm has a probability at least $\delta > 0$ of generating a independent set $\mathcal{A}_t$ of links with weight at least γ times the weight of the optimal. Then, capacity $\gamma \cdot \mathcal{C}$ can be achieved by switching links to the new independent set when its weight is larger than the previous one (otherwise, previous set of links will be kept for scheduling). The algorithm should generate the new scheduling I_t from the old scheduling I_{t-1} and current queue lengths q_t.*

Observe that here the main difficulty of applying Proposition 2 could be to compare the solution $\mathcal{A}_t$ and I_{t-1} to get schedule I_t, in addition to ensure that $\mathbf{Pr}\left(\mathcal{A}_t \cdot q_t \geq \gamma I_t^{\mathrm{OPT}} \cdot q_t\right) \geq \delta$ for constants $\delta > 0$ and $\gamma > 0$. In light of Proposition 1 and Proposition 2, we look for ways to generate link schedulings of approximately optimal weight, with at least a certain *constant* probability.

3. CENTRALIZED SCHEDULING

In this section, we will describe centralized efficient and stable scheduling algorithms with efficiency ratio arbitrarily close to 1. Two different kinds of algorithms will be presented: one kind of algorithms assume that we already know the geometry positions of all wireless nodes, whereas the other kind of algorithms assume that every wireless node u only knows the set of communication links incident on it (or more precisely, for each link (u, v), u knows all links that will interfere (u, v) and interfered by (u, v)). We will start by describing centralized scheduling algorithms that essentially illustrate our design approaches, and follow it in next section by presenting our distributed scheduling algorithms that are stable, throughput maximizing, and time efficient in next section.



(a) with location (b) without location

Figure 1: (a) Divide the space into grids for time-slot t. Here links with black nodes are candidates for I_t. Links whose transmitters fall in the removed area (links with white nodes in the figure) are removed. (b) Using bounded growth property when locations of nodes are unknown.

3.1 Utilizing Geometry Location

We first assume that each node u knows its geometry location, denoted as (x_v, y_v). There are two cases: 1) all nodes have uniform interference ranges and 2) nodes have different interference ranges.

3.1.1 Uniform Interference Ranges

We first assume that all nodes have uniform interference ranges. For simplicity of presentation, we first address the link scheduling problem under TIM: a scheduling of links is *feasible* if the distance between *any* two transmitting nodes are separated by an Euclidean distance at least $R = 1$. Recall that an optimum scheduling is to find a MWIS of links in the conflict graph. Thus, centralized algorithms (based on shifting strategy) of computing a MWIS have long been known when the input graph is a unit disk graph (UDG). Obviously, the conflict graph is rarely a UDG, even if the communication graph is a UDG. Our centralized scheduling approach is also based on shifting strategy and it works as follows.

We first partition the 2D space into grids using horizontal lines $x = i$ and vertical lines $y = j$ for all integers i and j. A vertical strip *with index* i is $\{(x, y) \mid i < x \leq i + 1\}$. Similarly, we can define a horizontal strip with index j and **cell**(i, j) as the intersection area of a vertical strip i and a horizontal strip j. See Figure. 1 for illustration, where the shaded area are strips. Thus, if two links are separated by a strip, then they can be scheduled for transmitting simultaneously under TIM. To divide the problem of finding a maximum weighted feasible scheduling into subproblems that are solvable in polynomial time, we divide links into groups based on grid partition. To ensure that the union of solutions of subproblems are still independent, as a standard approach, we will add a separation between adjacent subproblems as follows. At any time slot t, we will "remove" the links whose transmitters are located inside either vertical strips i with $i = a_t \mod k$ or horizontal strips j with $j = b_t \mod k$. Here we $a_t, b_t \in [0, k-1]$ are adjustable numbers. As illustrated in Figure 1, we remove all links with transmitters inside the gray strips. We define the preceding operations as Partition(k, a_t, b_t), *i.e.*, divide the space into grid-cells and remove some links. Given (a_t, b_t), define **square**(i, j) to be the set of cells $\{cell(x, y) \mid x \in [ik + a_t + 1, (i+1)k + a_t - 1], y \in [jk + b_t + 1, (j+1)k + b_t - 1]\}$. A subproblem is then, given a square(i, j), to find a MWIS of all links whose transmitter nodes are inside. Here each square has size $k - 1$ and two links whose transmitters are closer than $R = 1$ cannot transmit simultaneously. Thus, the size of *any* set of interference-free links for a square(i, j) is at most $\Lambda = (k-1)^2/\frac{\pi}{4} = O(k^2)$. This implies that a MWIS for each square(i, j) can be found by simple enumeration in time $n_{i,j}^{\Lambda}$, where $n_{i,j}$ is the number of nodes inside square(i, j).

Algorithm 1 Centralized Scheduling Using Geometry Information

Input: Location of nodes, queue size of every link, and k.
Output: Feasible active link set I_t for time slot t.

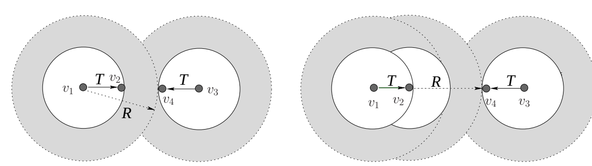
```
tmp = 0;
for a_t = 0 to k − 1 do
    for b_t = 0 to k − 1 do
        Partition(k, a_t, b_t);
        Compute a MWIS A_t^(i,j);
        if ⋃_(i,j) A_t^(i,j) · q_t > tmp · q_t then
            tmp = ⋃_(i,j) A_t^(i,j);
I_t = tmp;
```

We compute a scheduling as follows: At time slot t, we choose a partition (corresponding to some specific (a_t, b_t)) and compute a MWIS of links for each Square(i, j) that is not empty of nodes inside. Let $A_t^{(i,j)}$ be the optimum solution for square(i, j) using the weight q_t. Here the weight of a link $e_{i,j}$ is defined as the maximum queue size of node v_i. Obviously, there are k^2 different partitions since there are k^2 different choices for (a_t, b_t) and each of them corresponds to a distinct partition. Accordingly, we can choose the "best" partition among the k^2 partitions. Here the best partition refers to the partition such that the total weight of all MWISs for the squares is maximum among all k^2 different partitions. Let I_t be the union of the optimum solutions for all squares in the best partition. Pseudo-codes are listed in Algorithm 1.

Obviously, for any two links $e_{p,q}$ and $e_{x,y}$ from two different squares, the transmitters v_p and v_x are separated by distance at least the strip width $R = 1$. Thus, they are always interference-free. Thus, I_t generated by Algorithm 1 is an independent set. We then prove that it has a good approximation ratio.

THEOREM 3. *Given any vector q_t, there exists a partition such that the total weight of I_t computed by our algorithm, $I_t \cdot q_t \geq (1 - \frac{1}{k})^2 (I_t^{OPT} \cdot q_t)$.*

PROOF. In our algorithm, the links whose transmitters fall in some vertical strip i with $i \equiv a_t \mod k$ or horizonal strip j with $j \equiv b_t \mod k$ (the gray area in Figure 1) will be "removed". Note that there are totally k^2 different partitions since there are k^2 different (a_t, b_t) pairs and every cell(i, j) appears in the "removed" strips for exactly $2k - 1$ times. At the same time, all the removed cells form $2k - 1$ copies of the whole area of the network. Suppose the optimal solution is $(I_t^{OPT} \cdot q_t)$ for time slot t, then there always exists a partition such that the removed part of the optimal solution, *i.e.*, accumulated weight of the nodes in the gray area, is at most $\frac{2k-1}{k^2}(I_t^{OPT} \cdot q_t)$, by pigeonhole principle. Since the result generated by our algorithm is optimal in the squares, $I_t \cdot q_t$ is at least $(1 - \frac{2}{k} + \frac{1}{k^2})(I_t^{OPT} \cdot q_t)$, *i.e.*, $(1 - \frac{1}{k})^2 (I_t^{OPT} \cdot q_t)$. $\square$



(a) fPrIM (b) RTS/CTS Model

Figure 2: From TIM model other interference models.

Note that we only studied TIM. In fact our algorithm can be easily extended to deal with fPrIM and RTS/CTS model. We first extend our results to the scenarios where every node has uniform interference ranges under fPrIM and RTS/CTS model. Then we extend our results to scenarios where transmission and interference ranges can be different.

fPrIM: Suppose that interference range and transmission range of every node are R and T respectively. Consider any two directed links $e_{1,2}$ and $e_{3,4}$. Observe that if the distance $\|v_1v_3\|$ is at least $R+T$, then links $e_{1,2}$ and $e_{3,4}$ are interference-free. In Figure 2 (a), v_1 and v_3 can transmit simultaneously. We then partition the space into grids using horizontal lines $x = i(R+T)$ and vertical lines $y = j(R+T)$ for all integers i and j. In other words, the cell-size is now $R+T$. A vertical strip *with index* i is $\{(x,y) \mid i(R+T) \leq x \leq (i+1)(R+T)\}$. Similarly, we can define a horizontal strip with index j and cell(i,j) as the intersection area of a vertical strip i and a horizontal strip j. The other is the same with the simplified TIM. Clearly, the solution returned by Algorithm 1 is an independent set of links under fPrIM model.

Similar to Theorem 7, we can prove that the size of $A^t_{i,j}$ is at most a constant when $R > (1+\theta)T$ for a constant $\theta > 0$. Unfortunately, when $R = T$, we can construct example to show that $A^t_{i,j}$ could be as large as $\Theta(n_{i,j})$, where $n_{i,j}$ is the number of transmitters inside square(i,j). Then simple enumerating all independent sets could have exponential time $2^{n_{i,j}}$. When $R > (1+\theta)T$ (this is always true in practice) we can find the optimum solutions for each subsquare in polynomial time. Similar to Theorem 3, we have

THEOREM 4. *Under fPrIM, $\forall q_t$, there exists a partition such that I_t computed by our algorithm has $I_t \cdot q_t \geq (1-\frac{1}{k})^2(I^{OPT}_t \cdot q_t)$.*

RTS/CTS Model: We observe that under RTS/CTS model, for two nodes to transmit simultaneously, it suffices that the distance between them is at least $R+2T$. For example, in Figre 2(b), nodes v_1 and v_3 can transmit at same time without interfering each other. We then partition the space into grids as under fPrIM except that we let the width of each stripe be $R+2T$. Similar to Theorem 7, for any square, we can prove that the size of any set of interference-free links is at most a constant. Thus, Step 4 of Algorithm 1 can be done in polynomial time. Similarly we have,

THEOREM 5. *Under RTS/CTS model, $\forall q_t$, there exists a partition such that I_t computed by our algorithm satisfies $I_t \cdot q_t \geq (1-\frac{1}{k})^2(I^{OPT}_t \cdot q_t)$.*

3.1.2 Heterogeneous Interference Ranges

In this subsection, we will present algorithms for different interference models when endpoints of each link may have different transmission and interference ranges. Basically, we will find Maximum Weighted Interference Free Set(MWIFS) based on the result in [13]. Assume that we are given a set E=$\{e_{p_1,q_1}, e_{p_2,q_2}, ..., e_{p_m,q_m}\}$ of m links in a two-dimensional plane, where link $e_{p,q} = (v_p, v_q)$ has a sender v_p and receiver v_q, associated with an interference area $\mathcal{I}(e_{p,q})$, an interference length $L(e_{p,q})$ and weight $\omega(e_{p,q})$. The weight $\omega(e_{p,q})$ is the queue length of link $e_{p,q}$. For a subset of links $U \subseteq E$, let $\omega(U) = \sum_{e_{i,j}\in U} \omega(e_{i,j})$, i.e, the total weight of links in U.

The interference area $\mathcal{I}(e_{p,q})$ is the area in which links may be interfered by $e_{p,q}$. Here we say a link is in $\mathcal{I}(e_{p,q})$ when either sender or receiver of that link is in $\mathcal{I}(e_{p,q})$. The interference area $\mathcal{I}(e_{p,q})$ depends on interference model and interference ranges. For TIM, $\mathcal{I}(e_{p,q})$ is a disk $D(v_p, R_p)$ centered at v_p with radius R_p. For RTS/CTS model, $\mathcal{I}(e_{p,q})$ is $D(v_p,R_p)\bigcup D(v_q,R_q)$. For fPrIM, $\mathcal{I}(e_{p,q})$ is disk $D(v_p,R_p)$.

Our method will again partition the problem to smaller subproblems using grid-partition described previously. Recall that in sub-

---

Algorithm 2 Approximate MWIFS

Input:All m links, $L(e)$, $\mathcal{I}(e)$, $w(e)$ of each link e, and geometry location of each node and a constant Λ.
Output: Feasible active link set I_t for time slot t.

```
for all j = l + 1 downto 1 do
  for all every S with level j do
    Let R be all links in L(r, s) of level ≤ j interfere S.
    for all J ⊆ R with at most Λ links do
      if J is an interference-free set then
        Let X be interference links in J with level j.
        for every child square S' of S do
          Let I' be links in J interfere S'.
          Set X = X' ∪ IFS(S', I').
        Let I be links in J with level ≤ j.
        if ω(X) > ω(IFS(S, I)) then
          IFS(S, I) = X.
I_t ← ⋃_S IFS(S, ∅), where S is all squares with level 0;
```

---

subsection 3.1.1, all our proofs heavily rely on the fact that *any* feasible scheduling of links in *any* subproblem (links in a square) has at most a constant number of links. This is true only when "sizes" of the interference regions claimed by all links are same (true for result in subsubsection 3.1.1) or within a small constant factor of each other. This is clearly not true when nodes have different interference ranges. Our approach is 1) group links into different groups, where in each group, the "size" of the interference region claimed by every link is similar; 2) then partition the links of each group using space partition as did in subsubsection 3.1.1.

To characterize the size of interference region, we define the *interference length* $L(e_{p,q})$ of a link $e_{p,q}$ as following. For TIM, $L(e_{p,q}) = R_p$. For RTS/CTS model, $L(e_{p,q}) = \max(R_p, R_q)$. For fPrIM, $L(e_{p,q}) = R_p$. For simplicity, we normalize the largest $L(e_{p,q})$ to 1, and other lengths are scaled accordingly.

Let $\ell = \log \frac{\max_i L(e_i)}{\min_i L(e_i)}$. We partition links into $\ell+1$ levels such that level j, $0 \leq j \leq \ell$, consists of all links $e_{p,q}$ with interference length satisfying that $\frac{1}{(k+1)^{j+1}} < L(e_{p,q}) \leq \frac{1}{(k+1)^j}$. Let $\ell(e_{p,q})$ denote the level of link $e_{p,q}$, *i.e.*, $\ell(e_{p,q}) = \lfloor \log_{k+1} \frac{1}{L(e_{p,q})} \rfloor$. For each level j, we subdivide the space into grid by using a set of vertical lines $L_{j,v} : x = v\frac{1}{(k+1)^j}, v \in Z$ (with index v) and a set of horizontal lines $H_{j,h} : y = h\frac{1}{(k+1)^j}, h \in Z$ (with index z).

A (r,s)-shifting of the subdivision is the grid defined by all vertical and horizontal lines such that $v \mod k \equiv r$ and $h \mod k \equiv s$. For each (r,s)-shifting and level j, we divide the plane into squares by lines $L_{j,v}$ and $H_{j,h}$ such that $v \mod k \equiv r$ and $h \mod k \equiv s$. We call those squares level **j-square**. Furthermore, for each (r,s)-shifting, remove all links hit by the border lines (or more exactly, the links whose transmitters are inside the gray strips as in Figure 1) in corresponding j-squares, we call the remaining links $L(s,r)$. Here a link with level j will be removed *only* when it is hit by the border line of some j-square. It is *not* removed if only hit by squares from other levels.

Then we compute MWIFS using dynamic programming, as shown in Algorithm 2. The key difference between Algorithm 2 and algorithm in [13] is: instead of finding an independent set of nodes, we need to find a feasible interference-free set(IFS) of links. Notice that an interference-free link set is not necessary to be an independent node set. A link $e_{p,q}$ is said to interfere a square S if there is a link $e_{x,y}$ inside S such that $e_{p,q}$ interferes $e_{x,y}$. In this algorithm, we enumerate all possible feasible subset J in set R for each j-square. Since the size of J is no more than Λ, we can finish each enumeration within $O(m^\Lambda)$ time. Then the running time of this algorithm is $O(k^2 m^\Lambda)$. We will show that Λ is a constant in Theorem 7(see appendix for the proofs).

LEMMA 6. *Any given link $e_{p,q}$ is interfered by at most a constant number of interference-free links $e_{x,y}$ with larger length $L(e_{x,y})$.*

THEOREM 7. *Let S be any j-square and let I be a set of interference-free links with level at most j, each of which may interfere some links contained in S. Then there is a constant Λ depending on the interference model and k such that $|I| \le \Lambda$.*

For any set S, we use $OPT(S)$ to denote the total weight of MWIFS with links in S. The following lemma shows that our shifting strategy produces a good solution (see appendix for its proof).

LEMMA 8. *There is at least one (r, s)- shifting, $0 \le r,\ s < k$ such that $OPT(L(r, s)) \ge (1 - \frac{1}{k})^2 OPT(E)$ for TIM, fPrIM and RTS/CTS models respectively.*

3.2 Utilizing Bounded Growth Property

In this section, we present centralized approach for link scheduling in wireless networks where geometry information of nodes is unknown. We assume that a conflict graph $F_G = (V', E')$ is available through network measurement, where V' is the set of links E in G. Hereafter, our algorithm is based on F_G.

Our proposed method borrows idea from the algorithm for MWIS problem proposed in [21]. The basic idea is that for any time slot t we first select a vertex v with maximum weight in the current network; then we compute maximum weighted independent set Γ^r in the r-hop neighborhood N^r of v which includes v. See Figure 1 for illustration. Here N^r of vertex v is defined as:

$$N^r(v) := \{u \in V' | u \text{ has hop-distance at most } r \text{ from } v\}.$$

We repeat the process until the weight of Γ^r satisfies

$$W(\Gamma^{r+1}) = \Gamma^{r+1} \cdot q_t \ge \rho W(\Gamma^r) = \rho \Gamma^r \cdot q_t, \qquad (1)$$

where $\rho = 1 + \epsilon$ and $\epsilon > 0$. The process stops when inequality (1) is violated for the first time. We will prove that under fPrIM and RTS/CTS model, r is at most a constant $\bar{r}$ and $\bar{r}$ does exist for every vertex v we pick. We can prove that for all interference models, graph F_G is growth bounded. In other words, for any r, Γ^r has size at most $f(r)$ for a polynomial f when each vertex has weight 1: $f(r) = O(r^2)$ for all interference models studied.

We then "remove" $N^{\bar{r}+1}$ of vertex v including v. We repeat the above process until all the vertices in the network are "removed". Assuming that the vertices we have picked are $v_1, v_2, \cdots, v_b$, the candidate solution for I_t is the union $\bigcup_{i=1}^{b} \Gamma^{\bar{r_i}}(v_i)$. Note that we remove $(\bar{r_i} + 1)$-neighborhood of v_i instead of $N^{\bar{r_i}}$ in order to ensure that the union of $\Gamma^{\bar{r_i}}$ is independent.

Algorithm 3 Centralized Scheduling Using Bounded Growth

Input: $F_G = (V', E')$, queue size of every link, ρ and I_{t-1}.
Output: Feasible active link set I_t for time slot t.

1: **repeat**
2: Pick a vertex $v \in V'$ with maximum weight;
3: Compute $\Gamma^{\bar{r}}(v)$;
4: $I_t = I_t \cup \Gamma^{\bar{r}}(v)$;
5: $V' = V' - N^{\bar{r}+1}$;
6: **until** $V' = \varnothing$;

Now we prove that $\bar{r}$ does exist and is bounded by a constant (depending on ρ) in different interference models.

THEOREM 9. *There exists a constant $c = c(\rho)$ such that $\bar{r} \le c$ under fPrIM and RTS/CTS model.*

PROOF. We assume that the uniform transmission range of every node is T and the uniform interference range is R. Recall that, if two links $e_{p,q}$, $e_{x,y}$ conflict, the distance between two transmitting nodes is at most $R + T$ for fPrIM, at most R for TIM, and at most $R + 2T$ for RTS/CTS model. In all cases, the transmitters in $N^r(e_{p,q})$ is contained inside disk $D(v_p, r(R + 2T)) \subseteq D(v_p, r\frac{3+2\theta}{1+\theta}R)$. On the other hand, if two links $e_{p,q}$, $e_{x,y}$ are interference-free, the distance between the transmitting nodes v_p and v_x must be larger than $R - T \ge \frac{\theta}{1+\theta}R$ for fPrIM model and R for TIM and RTS/CTS models. This implies that, for fPrIM model, the cardinality of $\Gamma^r(e)$ for any link e is at most

$$|\Gamma^r| \le \pi(r\frac{2+\theta}{1+\theta}R)^2 / \pi(\frac{\theta}{1+\theta}R/2)^2 = \frac{4(2+\theta)^2}{\theta^2} r^2 = c_1 r^2, \qquad (2)$$

Similarly, we can prove that $|\Gamma^r| \le c_1 r^2$ for both TIM and RTS/CTS models for some constant $c_1 > 0$. Notice that

$$W(\Gamma^r) = \Gamma^r \cdot q_t = \sum_{i \in \Gamma^r} w_i \le \sum_{e_i \in \Gamma^r} w_{max} = |\Gamma^r| w_{max}, \quad (3)$$

where, w_i is the weight of a vertex e_i, *i.e.*, the queue size of link e_i, w_{max} is the weight of the initiating node. If $\bar{r}$ is not bounded, then according to (1), $\forall r$,

$$W(\Gamma^r) \ge \rho W(\Gamma^{r-1}) \ge \rho^2 W(\Gamma^{r-2}) \ge \cdots \ge \rho^r w_{max}. \quad (4)$$

Then $|\Gamma^r| w_{max} \ge \rho^r w_{max}$. Clearly, this will be violated when $r > r_0$ where r_0 satisfies $c_1 r_0^2 = \rho^{r_0}$, *i.e.*, $\rho = (c_1 r_0^2)^{1/r_0}$. In other words, $\bar{r} \le r_0$. ☐

Note that, due to (2), we may compute Γ^r by simple enumeration in time $O(n^{c^2})$, where $c = O(\bar{r}) = O(1/\epsilon^2 \log 1/\epsilon)$ and $\rho = 1 + \epsilon$. Similar to [21], we have

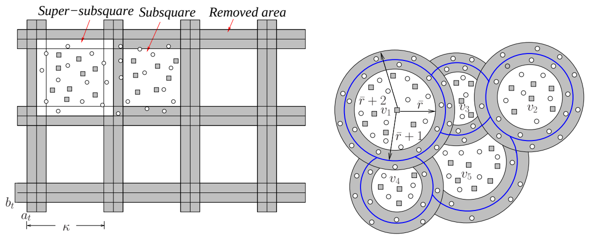
THEOREM 10. *I_t generated by Algorithm 3 is an independent set of weight at least $\frac{1}{\rho} = \frac{1}{1+\epsilon}$ of the weight of a MWIS. In other words, $I_t \cdot q_t$ is at least $\frac{1}{\rho}(I_t^{OPT} \cdot q_t)$.*

4. DISTRIBUTED SCHEDULING

In this section, we introduce our efficient distributed algorithms from two points of views as we did in section 3, *i.e.*, using geometry location and using bounded growth property. We first describe our distributed algorithms using TIM model where $R_i = 1$ for each v_i. Later, we will show that our algorithms can be easily extended to other interference models, like *fPrIM* and RTS/CTS model.

Notice that, all the communications of our distributed algorithms are based on a CDS, *i.e.*, the global CDS will be constructed at the beginning of both distributed algorithms. Our main idea is to let CDS relay packets between wireless nodes. Several methods [1] were proposed to get a CDS from G such that CDS C has a bounded degree d. In addition, because some of our algorithms works on conflict graph instead of communication graph, we assume one hop in the conflict graph of G corresponds to at most β hops on G, *i.e.*, we assume if there is a link between two vertices (corresponding to links in G) e_1 and e_2 in the conflict graph, one of end nodes of e_1 can reach one of end nodes of e_2 in at most β hops, where β is a constant, typically 2.

The challenges for designing low-complexity distributed scheduling with efficiency ratio $1 - \epsilon$ are that 1) when we use Proposition 1, it is difficult to find a MWIS distributively with approximation ratio $1 - \epsilon$ for *every* time-slot; 2) when we use Proposition 2, it is



(a) with location (b) without location

Figure 3: Here the round white nodes are the solution we computed from time slot $t-1$, grey rectangle nodes are the solution we computed from time slot t.

expensive to compare two global solutions. We will propose various methods that either address the first challenge or the second challenge or both simultaneously.

4.1 Using Geometry Location

First we consider the TIM when nodes have uniform $R = 1$. When every node knows its geometry location, we partition the whole space into cells with size 1. Then every node knows exactly which cell it belongs to. See Figure 3(a) for illustration. For centralized algorithm, our method guarantees finding a $(1-1/k)^2$-approximation of MWIFS at each slot t by finding the best partition. Clearly, this is impossible when we need low-complexity distributed scheduling. As in the literature, we will adopt the *pick and compare* approach. Randomly picking a partition (using random (a_t, b_t) at time slot t) guarantees that, with probability at least $1/k^2$ we will end up with the best partition, and a MWIFS whose weight is at least $(1-1/k)^2$ of the optimum. The challenge now is to compare such candidate solution $\mathcal{A}_t$ with previous solution I_{t-1} and then find the better one efficiently. To address this, we will find a *special* solution $\mathcal{A}_t$ that is guaranteed to be better than I_{t-1}. Then, the *compare* operation is not necessary.

Recall that, using space partition, I_{t-1} (similar to Algorithm 1) is composed of optimum solutions from each square(i,j). If we keep the *same* space partition (same (a_t, b_t) for all t) for all time-slots, clearly, we can produce the solution $A_t^{(i,j)}$ for each square(i,j) and $A_t^{(i,j)} \cdot q_t \geq I_{t-1}^{(i,j)} \cdot q_t$ from the optimality of $A_t^{(i,j)}$ for square(i,j) with weight q_t. Consequently, $A_t \cdot q_t \geq I_{t-1} \cdot q_t$. However, using same partition for all time-slots clearly violates the property that A_t has constant approximation ratio with constant probability when $t \to \infty$. The key observation is that, after *fixing* a partition, the removed links (whose senders fall inside the gray strips) will accumulate packets since they will *never* be served now. Thus, to ensure the constant probability of getting good solution, we need *randomly* choose a partition for every time-slot. The challenge now is to ensure that A_t is always better than I_{t-1}. To address this, for a square(i,j) partitioned in time t, when we compute a solution $A_t^{(i,j)}$, we will compare the local optimum solution using q_t, with some special partial solution of I_{t-1} that are locally known to square(i,j), and the better one is chosen as final $I_t^{(i,j)}$.

To describe our method in detail, we define some terms first. A **sub-square**(i,j) is the set of grid cells: $\{cell(x,y) \mid x \in [i*k+a_t+2, (i+1)*k+a_t-1] y \in [j*k+b_t+2, (j+1)*k+b_t-1]\}$. A **super-subsquare**$(i,j)$ is the set of grid cells $\{cell(x,y) \mid x \in [i*k+a_t+1, (i+1)*k+a_t] y \in [j*k+b_t+1, (j+1)*k+b_t]\}$. Clearly, the collection of super-subsquares will be a space partition. A sub-square(i,j) is contained inside the super-subsquare(i,j). See Figure 3(a) for illustration, where the larger square region is a super-subsquare(i,j) and the smaller square region is a sub-square$(i+1,j)$.

At any time slot t, we will "remove" the links whose senders are located inside either vertical strips i and $i+1$ with $i = a_t \mod k$ or horizontal strips j and $j+1$ with $j = b_t \mod k$, *i.e.*, links whose transmitters are inside the gray region of Figure 3(a) will be moved. Observe that, for every k strips, we remove *two* consecutive strips instead of one-strip used by centralized algorithm. This is used to ensure that the union of $A_t^{(i,j)}$ for all sub-square(i,j) is independent. Our algorithm works as follows.

Step 1: At time slot 0, we do a partition using $(a_0, b_0) = (0,0)$ and compute an optimum MWIS $A_0^{(i,j)}$ of nodes for each sub-square (i,j) that is not empty of nodes inside. Here the weight of a node v is defined as the maximum queue size of all *out-going* links of node v. Let the solution I_0 of time slot 0 be the union of the optimum solutions $A_0^{(i,j)}$ for all sub-squares.

Step $t+1$: For any time-slot t, we partition the space using (a_t, b_t). Here we choose (a_t, b_t) as (t,t) when $k \geq 5$. Observed that when $k = 3$ (or 4), some cells will be "removed" in *every* time-slot if $(a_t, b_t) = (t,t)$. Therefore, when $k = 3$ (or 4), we let (a_t, b_t) map to one distinct partition of the total 9 (or 16) different partitions (we can use a random permutation σ of $\{(a,b) \mid a, b \in [0, k-1]\}$ to get $(a_t, b_t) \leftarrow \sigma(t)$, the partition will repeat after k^2 time-slots. We then compute the optimum MWIS, denoted as $A_t^{(i,j)}$, for all sub-squares (i,j) using the weight q_t.

Let $I_{t-1}^{(i,j)}$ be the set of nodes from I_{t-1} (the global solution at time slot $t-1$) that fall in the super-subsquare (i,j) instead of sub-square(i,j). Clearly, we can compute such set locally. If $I_{t-1}^{(i,j)} \cdot q_t > A_t^{(i,j)} \cdot q_t$, let $I_t^{(i,j)} = I_{t-1}^{(i,j)}$, else $I_t^{(i,j)} = A_t^{(i,j)}$. Then the global solution is the union of $I_t^{(i,j)}$ for all super-subsquares.

The pseudo-codes are given in Algorithm 4. Note that for each super-subsquare, one node (assume u) will become the (only) coordinator computing the MWIFS of links inside this super-subsquare (actually the subsquare contained by this super-subsquare). Here we can simply choose the node which is closest to the center of each super-subsquare as the coordinator node for this super-subsquare. And, we assume the message RESULT($I_t^{(i,j)}$) used in Algorithm 4 contains all the needed information of all independent links selected by the coordinator inside super-subsquare(i,j) in time slot t. Here every node marks all its out-going links White at the beginning of a time-slot t; it will mark a link Red if it is chosen to be active, and Black otherwise.

THEOREM 11. *I_t generated by Algorithm 4 is an independent set.*

PROOF. We prove it by induction. At time slot 0, I_0 is the union of $A_0^{(i,j)}$, the MWIS in sub-square(i,j). The union of $A_0^{(i,j)}$ is an independent set since two links from two different sub-squares are independent. In other words, I_0 is an independent set. If I_{t-1} is an independent set when $t \geq 1$ then we prove I_t is an independent set. We observe that in each super-subsquare(i,j), either $A_t^{(i,j)}$ or $I_{t-1}^{(i,j)}$ is chosen to be a part of I_t. Consider any two different super-subsquare(i,j) and super-subsquare(i',j'). If $I_{t-1}^{(i,j)}$ and $I_{t-1}^{(i',j')}$ are chosen respectively as $I_t^{(i,j)}$ and $I_t^{(i',j')}$, the union of them is an independent set by induction. If either $I_t^{(i,j)}$ is $A_t^{(i,j)}$ or $I_t^{(i',j')}$ is $A_t^{(i',j')}$ or both, the union of them is still an independent set since they are separated by corresponding vertical and horizontal strips. Therefore I_t is an independent set. As an illustration, in Figure 3(a), the round white nodes in one super-subsquare do not

Algorithm 4 Distributed Scheduling by node v With Location

Input: k, a_t, b_t.
Output: Active or not for each of its outgoing links at time slot t.

state = White; active = NO; Coordinator = NO;
Calculates which cell Z node v resides in regarding to the current partition(k, a_t, b_t,);
if I am the closest one to the center of super-subsquare **then**
  Coordinator=YES;
if Coordinator = YES **then**
  Collect the queue size q_t for all links within the same super-subsquare(i, j), and the scheduled links $I_{t-1}^{(i,j)}$ in current super-subsquare(i, j) at previous time-slot $t - 1$ also.
  Computes MWIS $A_t^{(i,j)}$ in sub-square(i, j);
  if $I_{t-1}^{(i,j)} \cdot q_t > A_t^{(i,j)} \cdot q_t$ **then**
    $I_t^{(i,j)} = I_{t-1}^{(i,j)}$;
  else
    $I_t^{(i,j)} = A_t^{(i,j)}$;
  Broadcasts RESULT($I_t^{(i,j)}$) in super-subsquare(i, j);
if state= White **then**
  if receives message RESULT($I_t^{(i,j)}$) **then**
    if $v \in I_t^{(i,j)}$ **then**
      state = Red; active=YES;
    else
      state = Black; active=NO;

collide with other round white nodes in another super-subsquare since they are disjoint subsets of I_{t-1}; and the round white nodes in one super-subsquare do not collide with grey rectangle nodes in another sub-square since they are separated. This finishes the proof. □

As we have mentioned before, node u will use a good CDS to collect and send packets to all other nodes. Our main idea is that let the part of global CDS that fall in one super-subsquare to take charge of communications between wireless nodes, *i.e.*, relay messages. Because all super-subsquares are independent and we compute the local MWIS individually, we discuss the time complexity and message complexity inside one super-subsquare. For simplicity, without causing ambiguity, we use $G = (V, E)$ to denote the communication graph inside a super-subsquare and C is the set of nodes from global CDS falling in this super-subsquares. Assume that the graph formed by all nodes in C is $G_C = (V_C, E_C)$ and assume that G_C is connected. Let T_C denote the BFS tree of G_C. Then, the maximum degree in T_C is bounded by constant d as well. For any node $v \in C$, we add a link $(v, P(v))$ to T_C, here $P(v)$ is the dominator node of v. The resulting graph is called *Data Communication Tree (DCT)* H of G. Clearly, for each super-subsquare, each link in the H has at least one end node in the CDS. We first show that every link in G_C can be active at least once in constant time by Lemma 12.

LEMMA 12. *For any link e of $E(H)$, let $\psi(e)$ be the set of links that interfere with e, then $|\psi(e)| \leq c * d * \Delta(G)$, where c only depends on R/T.*

PROOF. For any edge $e = (u, v)$ of H, we know that either u or v (or both) will belong to C. Without loss of generality, we assume $u \in C$. For all edges interfere with e, both end nodes should be within distance $2T + R$ from u (this is true for all interference models we discussed). There are at most a constant number of nodes from CDS within this range since $R = \Theta(T)$ and the CDS has a constant-bounded degree. By an area argument, we can show that $\psi(e) \leq (\frac{R+2T}{T/2})^2 \cdot d \cdot \Delta(G)$. This finishes the proof. □

LEMMA 13. *Any two nodes inside a super-subsquare can communicate with each other in $O(k^2\Delta(G))$ mini-time-slots.*

PROOF. As we know, the area of one super-subsquare is equal to $(\phi k)^2$, where $\phi = \Theta(R)$ is the width of one strip. If we divide one super-subsquare into mini-cells with side length $\frac{T}{\sqrt{5}}$, every two nodes from two adjacent mini-cells can communicate with each other directly. We know that the hop number of shortest path (in original communication graph G) between any two nodes is bounded by $O(\frac{(2\phi k)^2}{(\frac{T}{\sqrt{5}})^2}) = O(k^2)$. When the communications are based on the CDS, the hop numbers will also be at most $O(k^2)$. By Lemma 12, any two nodes can communicate with each other in $O(k^2 \cdot c \cdot d \cdot \Delta(G))$ mini-time-slots. □

LEMMA 14. *The time complexity of one round in Algorithm 4 is $\Theta(N)$. Here N denotes the number of nodes in a super-subsquare.*

PROOF. As described in Algorithm 4, from the start of a round to time when all nodes (inside a super-subsquare) know their roles in this round, there are 3 phases. In the first phase, the coordinator node collects information (like ID, weight) from all other nodes in the same super-subsquare. In the second phase, based on the information collected, coordinator node will compute the MWIS and compare the MWIS with the total weight of all nodes in this super-subsquare from the global MWIS of last round. Coordinator node will pick the better one of two sets as the local MWIS for this round. In the last phase, coordinator node will broadcast the information of local MWIS to all nodes within the same super-subsquare. So the total time consumed by all 3 phases are one data collection (phase 1), computation (phase 2) and broadcast locally (phase 3). Phase 2 will not incur delay and phase 3 can clearly be done in time $O(N)$ using CDS.

From Lemma 12, in $O(\Delta(G))$ time-slots, the messages from each dominatee node are collected to the corresponding dominator node in the CDS. We can show that after $N + h(T_C)$ rounds, all messages can be scheduled to arrive in the root (the coordinator node) of tree H, where $h(T_C)$ is the height of the BFS tree T_C rooted at the coordinator node. Notice that $h(T_C) = O(k^2)$. Thus, the total mini-time-slots needed is $O(\Delta(G)) + O(N + O(k^2)) = O(N)$ since $\Delta(G) \leq N$. □

LEMMA 15. *Using CDS, any bit is relayed by at most a constant number of nodes.*

PROOF. We have shown in Lemma 13 that for any two nodes u and v within the same super-square, the hop number $H_G(u, v)$ is bounded by a constant, and the hop number between u and v on data communication tree H satisfies $H_T(u, v) \leq 2+3H_G(u, v)$, a constant as well. So, any bit will be only relayed constant times. □

LEMMA 16. *When $k \geq 5$, Algorithm 4 has a probability of at least $\frac{1}{k}$ to generate an independent set of links with weight at least $(1-\frac{4}{k})$ of optimal solution, i.e., **Pr** $\left(I_t \cdot q_t \geq (1 - \frac{4}{k})(I_t^{OPT} \cdot q_t)\right) \geq \frac{1}{k}$. When $k = 3$ or 4, Algorithm 4 has a probability of at least $\frac{1}{k^2}$ to generate an independent set of links with weight at least $(1-\frac{2}{k})^2$ of optimal solution, i.e., **Pr** $\left(I_t \cdot q_t \geq (1 - \frac{2}{k})^2(I_t^{OPT} \cdot q_t)\right) \geq \frac{1}{k^2}$.*

PROOF. Since we let $(a_t, b_t) = (t, t)$ when $k \geq 5$, there are total k different partitions. Each cell(i, j) appears in the "removed" strips for at most 4 times. Suppose the optimal solution is $(I_t^{OPT} \cdot q_t)$ for time slot t, then there exists at least one good partition such that the removed part of the optimal solution, i.e., accumulated weight of the nodes in the gray area, is at most $\frac{4}{k}(I_t^{OPT} \cdot q_t)$. Since the result generated by Algorithm 4 for this good partition is optimal in the remaining area, $I_t \cdot q_t$ is at least $(1-\frac{4}{k})(I_t^{OPT} \cdot q_t)$. Therefore the best partition generates an independent set of links with weight at least $(1-\frac{4}{k})$ of optimal solution. With probability $\geq \frac{1}{k}$, (t,t) is the best partition.

When $k = 3$ (or $k = 4$), there are total k^2 different partitions. Each cell appears in the "removed" strips for exactly $4k-4$ times. For similar reason above, there exists at least one partition such that the removed part of the optimal solution is at most $\frac{4k-4}{k^2}(I_t^{OPT} \cdot q_t)$ for any time slot t. Therefore for this good partition Algorithm 4 generates an independent set of links with weight at least $(-1\frac{4k-4}{k^2})$ of the optimal solution, *i.e.*, $(1-\frac{2}{k})^2$ of the optimal solution. The probability any partition is a good partition is at least $\frac{1}{k^2}$. □

THEOREM 17. *Algorithm 4 achieves $(1-\frac{4}{k})\mathcal{C}$ capacity for any $k \geq 5$ and $(1-\frac{2}{k})^2\mathcal{C}$ capacity for any $k = 3$ or 4.*

The claim holds by Proposition 2 and Lemma 16. For $k \geq 5$, the efficiency ratio can be improved to $(1-\frac{2}{k})^2$ using random (a_t, b_t) partition. For other interference models, instead of using cell size R, we will partition the space using cells of size $R+T$ for fPrIM and cells of size $R+2T$ for RTS/CTS model. Algorithm 4 again achieves $(1-\frac{2}{k})^2\mathcal{C}$ capacity for any $k \geq 3$ for these two interference models.

4.2 Using Bounded Growth Property

In this section, we introduce our distributed scheduling algorithm when we do not have geometry locations of nodes. Notice that in centralized scheduling, to guarantee the correctness, we start from a link e with the largest queue size and then grow the region until a certain criterion (inequality (1)) is violated. Given ρ, we know that we will explore at most $\bar{r}$-hops neighborhood $N^{\bar{r}}$ of e in the conflict graph. Thus, other links that do not have the global largest queue size can also start to explore its neighborhood $N^{\bar{r}}$ and find a MIS simultaneously. Let $k = \bar{r}$ be the control parameter depending on ρ. To ensure that two simultaneous explorings will be consistent, we need that any two intiating links must be separated by at least $2k+4$ hops in the conflict graph F_G. For having a low-complexity stable distributed scheduling with efficiency ratio arbitrarily close to 1, we will use the *pick and compare* idea similar to previous subsection. Assume that for each pair of conflicting links (v_p, v_q) and (v_x, v_y), the hop distance between them in communication graph G is at most a constant β. Our main idea is as follows:

Step 1: At the beginning of each time slot t, every node collects link information in its $(2k+4)$-hop neighborhood of conflict graph, that is, its $((2k+4)\beta)$-hop neighborhood $N_G^{(2k+4)\beta}(v)$ of communication graph G. Here constant k is a system parameter. The wireless node with maximum weight in its $((2k+4)\beta)$-hop neighborhood will become a coordinator for local MWIS computation. We choose the value of k in a way such that every node initiating a local MWIS computation terminates the computation within k hops. Note that we let each node collect information in its $((2k+4)\beta)$-hop neighborhood for ensuring that MWISs computed by simultaneous coordinators will always be independent.

Step 2: Based on the collected information, if a node v has maximum weight among all its neighbors within $\beta(2k+4)$ hops, v starts to compute local MWISs $\Gamma^0(v)$, $\Gamma^2(v)$, ..., $\Gamma^{\bar{r}}(v)$ by enumeration. Different from the centralized algorithm, we find a $\bar{r}$ such that $\rho W(\Gamma^r(v)) \leq W(\Gamma^{r+2}(v))$ when $r < \bar{r}$ and $\rho W(\Gamma^{\bar{r}}(v)) > W(\Gamma^{\bar{r}+2}(v))$. Note that we prove in Theorem 18 that an $\bar{r} \leq k$ does exist according the bounded growth property of wireless network under the interference models we considered. Here k depends on ρ. Let $A_t^v = \Gamma^{\bar{r}}(v)$, denoting the local result for time slot t using the weight q_t. Let $I_{t-1}^{\bar{r}+1}$ be the set of nodes from I_{t-1} (the

Algorithm 5 Distributed Scheduling Using Bounded Growth

Input: k, ρ.
Output: Active or not in time slot t.

state = $White$; active = NO; head = NO;
Collects information (*e.g.*, $q_t(e)$) from N^{2k+4} in F_G.
if $w_v \geq w_u$, for any $u \in N^{2k+4}(v)$ **then**
  head = YES;
if head = YES **then**
  Computes $\Gamma^0, \Gamma^2, ..., \Gamma^{\bar{r}}, \Gamma^{\bar{r}+2}$ such that $\Gamma^{i+2} \cdot q_t \geq \rho * \Gamma^i \cdot q_t$, for $0 \leq i \leq \bar{r}-2$, and $\Gamma^{\bar{r}+2} \cdot q_t < \rho\Gamma^{\bar{r}} \cdot q_t$.
  $A_t^v = \Gamma^{\bar{r}}$;
  $I_{t-1}^{\bar{r}+1} = I_{t-1} \cap N^{\bar{r}+1}$;
  if $I_{t-1}^{\bar{r}+1} \cdot q_t > A_t^v \cdot q_t$ **then**
    $I_t^v = I_{t-1}^{\bar{r}+1}$;
  else
    $I_t^v = A_t^v$;
  Broadcasts message RESULT(I_t^v) in $N^{\bar{r}+2+2k+4}$;
if state = $White$ AND head = NO **then**
  if receives message RESULT(I_t^u) **then**
    if $v \in I_t^u$ **then**
      state = Red; active = YES;
    if $v \in N^{\bar{r}+2}$ AND $v \notin I_t^u$ **then**
      state = $Black$; active = NO;
    if $v \in N^{\bar{r}+2+2k+4} \setminus N^{\bar{r}+2}$ **then**
      If v has no $White$ neighbor within $2k+4$ hops that has larger weight, goto 5;

global solution computed for time slot $t-1$) that $\in N^{\bar{r}+1}(v)$, i.e., $I_{t-1}^{\bar{r}+1} = I_{t-1} \cap N^{\bar{r}+1}(v)$. Note that at time slot 0, let $I_{t-1}^{\bar{r}+1}$ be zero vector without loss of generality. Obviously, we can compute $I_{t-1}^{\bar{r}+1}$ locally. In Figure 3(b), the region enclosed by the blue circle (middle circle) indicates the $(\bar{r}+1)$-neighborhood of a node; The white nodes are nodes in I_{t-1}; the grey rectangle ones are computed in time slot t. If $I_{t-1}^{\bar{r}+1} \cdot q_t > A_t^v \cdot q_t$, we let $I_t^v = I_{t-1}^{\bar{r}+1}$, otherwise $I_t^v = A_t^v$.

Step 3: v then announces I_v^t in its $(\bar{r}+2+2k+4)$-neighborhood of conflict graph ($\beta(\bar{r}+2+2k+4)$-hops in the communication graph) and "removes" $N^{\bar{r}+2}$ from the conflict graph (at most $N^{\beta(\bar{r}+2)}$ in communication graph). As a result, some node $u \in N^{\bar{r}+2+2k+4} \setminus N^{\bar{r}+2}$ might find that it has the maximum weight in its $(2k+4)$-neighborhood and it can start to compute its local MWISs.

Notice in the pseudocode of Algorithm 5, message RESULT(I_t^v) contains the information of all nodes selected by node v as the local solution in time slot t. In addition, we use several colors to distinguish the different status of vertices. For example, if a node v marks itself with color Red, it means v is selected in the solution for this round. $Black$ means the node is not selected as the solution in this round.

THEOREM 18. *There exists a constant $c = c(\rho)$ such that $\bar{r} \leq c$ in fPrIM model and RTS/CTS model.*

The proof is similar to that of Theorem 9, based on observation.

$$W(\Gamma^r) \geq \rho W(\Gamma^{r-2}) \geq \rho^2 W(\Gamma^{r-4}) \geq \cdots \geq \rho^{\lfloor \frac{r}{2} \rfloor} w_{max}.$$

THEOREM 19. *I_t generated by Algorithm 5 is an independent set and, $I_t \cdot q_t \geq \frac{1}{\rho}(I_t^{OPT} \cdot q_t)$.*

PROOF. Let $\bar{V'} = V' \setminus N^{\bar{r}+2}$, and inductively assume that $\Gamma' \subset \bar{V'}$ is a $\rho-$approximation independent weighted set in $F_G[\bar{V'}]$. Obviously, $I_t = \Gamma^{\bar{r}} \cup \Gamma'$ is an independent set in F_G. Since $\Gamma^{\bar{r}+2}$ is a MWIS in $N^{\bar{r}+2}$, we have $W(I_t^{OPT} \cap N^{\bar{r}+2}) \le W(\Gamma^{\bar{r}+2}) \le \rho W(\Gamma^{\bar{r}})$. Thus,

$$\begin{aligned} & W(I_t^{OPT}) = I_t^{OPT} q_t = W((I_t^{OPT} \cap N^{\bar{r}+2}) \cup (I_t^{OPT} \cap \bar{V}')) \\ = \; & W(I_t^{OPT} \cap N^{\bar{r}+2}) + W(I_t^{OPT} \cap \bar{V}') \\ \le \; & \rho W(\Gamma^{\bar{r}}) + \rho W(\Gamma') = \rho W(\Gamma^{\bar{r}} \cup \Gamma') \le \rho W(I_t) = \rho I_t \cdot q_t. \end{aligned}$$

Thus, $I_t \cdot q_t \ge \frac{1}{\rho} I_t^{OPT} \cdot q_t$. Note that here $W(I_t) \ge W(\Gamma^{\bar{r}} \cup \Gamma')$ since every initiating node chooses $max(\Gamma^{\bar{r}}, I_{t-1}^{\bar{r}+1})$ in terms of weight. □

Next, we show that our Algorithm 5 for growth bounded graph also has bounded time and message complexity. Remember that all the communications are based on the CDS, *i.e.*, when any node collects or sends information locally, we will construct CDS first and use it as backbone locally to relay information.

LEMMA 20. *For any node in the graph G, collecting (or sending) information from (or to) any node within $\beta(2k+4)$ hops can be done in time $O(k\Delta(G))$ and any bit will be relayed in $3 \times \beta(2k+4)$ mini-time-slots in each round. Here, β and k are constants.*

PROOF. In Algorithm 5, every node will collect information from all neighbor nodes within $\beta(2k+4)$ hops in each round, here β is a constant. Based on the CDS, we know every link can be active at least once in every $c \cdot d \cdot \Delta(G)$ mini time-slots due to Lemma 12. In addition, for any two nodes u and v which are at most $\beta(2k+4)$ hops away in graph G, the hop distance between them are at most $2 + 3 \times \beta(2k+4)$ by using CDS. Thus, the time for any node u to send a message to any other node v is at most $(2+3\times\beta(2k+4))c \cdot d \cdot \Delta(G)$ which is $O(k \cdot \Delta(G))$. Clearly, for any bit need to be sent from u to v, there are at most $3 \times \beta(2k+4))$ nodes on the path between u and v when CDS is used.

When a coordinator node u wants to send a message to some node within $\beta(3k+6)$ hops, the procedure can be viewed as a reverse of data collection from coordinator node. □

Notice that, in the growth bound graph a bunch of wireless nodes with largest weight in their local neighborhood will become the coordinator nodes locally and start the algorithm simultaneously. We show the total time complexity and message complexity of Algorithm 5 in each round are bounded by the following Lemma 21.

LEMMA 21. *With Algorithm 5, each node v_i will know if $v_i \in I_t$ in $O(\Delta(G) \cdot [\beta(2k+4)]^2)$ mini-time-slots, and the number of total messages needed to collect and broadcast information is $O(\Delta(G) \cdot [\beta(2k+4)]^2)$. Here, both β and k are constant.*

PROOF. At the beginning every node not on the CDS will send its information to its dominator, which can be finished in $c \cdot d \cdot \Delta(G)$. Any node u has at most $\Delta(G) \cdot [\beta(2k+4)]^2$ neighbors within $\beta(2k+4)$ hops by an area argument. By using the similar proof in Lemma 14, we get the total time used is $O(\Delta(G) \cdot [\beta(2k+4)]^2)$. In addition, the time of computation and comparison consumed by the coordinator can be considered as $O(1)$. Obviously, the time for the coordinator to broadcast the locally solution to all nodes within $\beta(3k+6)$ hops is at most of the first step. Thus, the total time used is bounded by $O(\Delta(G) \cdot [\beta(2k+4)]^2)$.

On the other hand, we know that the coordinator node has at most $\Delta(G) \cdot [\beta(2k+4)]^2$ nodes within $\beta(2k+4)$ hops. And for each node (within $\beta(2k+4)$ hops), the message from it to the coordinator node is at most relayed by $3\beta(2k+4))$ hops. Thus, the total message complexity is bounded by $O(\Delta(G) \cdot [\beta(2k+4)]^2) 3\beta(2k+4)) = O(\Delta(G) \cdot k^3)$. □

By Proposition 2 and Theorem 19, we have, $\forall k \ge 3$,

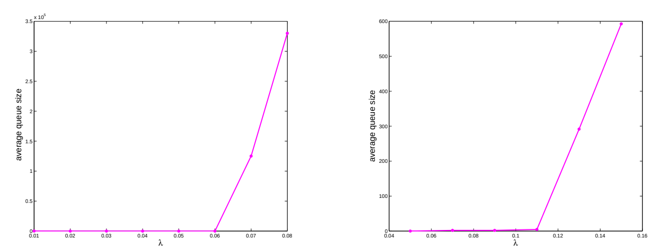
THEOREM 22. *Algorithm 5 is stable, achieves $\frac{1}{\rho} \cdot \mathcal{C}$ capacity. Here constants k and ρ satisfy that $c_1 k^2 = \rho^k$.*

Note that our distributed scheduling here guarantees to find a scheduling with efficiency ratio $1/\rho$, however, it could run in linear mini-time-slots in the worst case. Thus, using Proposition 1, we do not need compare solutions in time $t-1$ and t (thus the removed strips with width 1 still works for distributed method). However, our *pick and compare* approach here provides a foundation for designing time-efficient distributed scheduling using only topological information, in which we *only* need to ensure **Pr** $\left(I_t q_t \ge \gamma I_t^{\mathrm{OPT}} q_t\right) > \delta$ for a constant $\delta > 0$. A possible approach is to let a node v serve as coordinator with a probability depending on its queue size and control parameter k.

5. SIMULATION RESULTS

In this section, we present the simulation results that evaluate our distributed algorithms. Here, we mainly show the results concerning the stability of the network when our distributed algorithms execute due to the limited space. We simulated a fixed 10×10 homogeneous grid network similar to the one used in [25]. There are 100 wireless nodes with transmission range 1 deployed in each of the intersections of two lines. Our simulation is based on TIM and all nodes have interference range 2. We assume the packets arriving at a node by a poisson process with rate λ. Obviously the average queue size of all nodes always increases with time when $\lambda \notin \mathcal{C}$.

The performance of Algorithm 4 is illustrated in Fig. 4 (a). Here, we set $k = 8$. Initially the queue size of each node is 0. After running for 1000 time slots, the results of average queue size in the network are showed in Figure 4(a). We can see that when λ is no larger than 0.06, the queue size is almost 0, which indicates that all arrived packets for each node are scheduled. The queue size increases linearly over time when $\lambda > 0.06$.



(a) with geometric information (b) using bounded growth

Figure 4: Average queue size for distributed algorithm.

For Algorithm 5, similarly, Figure 4(b) shows the same property. We set $k = 3$ in this case. In addition, the number of mini-time-slots needed to schedule a growth bounded graph by algorithm 5 is bounded and does not increase with the number of nodes. In our simulations, the average loop time to complete the scheduling is 6.9 in a 36 nodes' random network; the average mini-time-slots is 10.6 in a 70 nodes' random network; 22.9 in a 282 nodes' random network; 28.6 in a 829 nodes' random network. Notice the performance of the network could be improved by increasing k, which allows several similar areas in the network scheduled simultaneously. We omit the simulation results here due to limited space.

6. RELATED WORK

Interference-free link scheduling in multihop wireless network has been extensively studied. In their seminal work [28], Tassiulas and Ephremides consider a synchronized slotted system where each frame consists of a single slot. They propose a scheduling policy that at each slot, selecting a link transmission set with maximum total queue sizes. It is proved that this policy achieves the maximum throughput region. Tassiulas proposes randomized centralized algorithms in [27] which can achieve the capacity region with $O(n)$ time complexity, where n is the network size.

The interference model adopted in the scheduling scheme is important. It is well known [4] that for an arbitrary interference model, the maximum throughput scheduling problem is NP-complete and not approximable within $m^{\frac{1}{3}-\epsilon}$ for any arbitrarily small $\epsilon > 0$ for a network of m links, unless NP = ZPP. On the other hand, since there is no central entity in the multihop wireless network, distributed link scheduling is preferred. Using primary interference model, Modiano *et al.* [20] present the first distributed link scheduling scheme for multihop wireless networks that achieves nearly the capacity region, based on the *pick and compare* approach [27] and a distributed matching. Scheduling overheads were not considered. In [4], Sharma *et al.* first compare maximal scheduling, pick and compare, and some constant-time scheduling approaches. Then they propose randomized maximal scheduling algorithms based on maximal matching under primary-interference model, and 2-hop interference model (without using pick and comparing approach), that runs in time $\log^3 |V|$ and achieves $\frac{1}{\alpha^1(G)}$ of the maximum throughput. Here $\alpha^1(G) = \max_{e \in E} \alpha^1(F_G)$ and $\alpha^1(F_G)$ is the 1-hop independence number of F_G, *i.e.*, the largest number of links from $\psi(e)$ that will not cause interference among themselves. Here $\psi(e)$ is the set of all links interfering a link e in the communication graph $G = (V, E)$. Penttinen *et al.* [22] propose a distributed and fair link scheduling algorithm for multihop wireless networks. In [10], the authors propose centralized and distributed algorithms computing an MWIS on conflict graph under primary interference model for tree-structured wireless networks. The scheduling overhead however grows with network size.

A few methods with constant overhead were also proposed. Lin and Rasool [14] propose two distributed and probabilistic scheduling algorithms which incur constant overhead. In their proposed algorithms, each link computes a transmission probability based on the queue-length information of its own and its interfering links. They prove that the proposed algorithms respectively achieve $\frac{1}{3} - \epsilon$ of the capacity region under primary interference model and $\frac{1}{1+\Delta} - \epsilon$ of the capacity region under 2-hop interference model. This is improved by [4,7]. Joo *et al.* [7] propose another distributed constant-overhead probabilistic scheduling algorithm based on [14]. Under primary interference model, the algorithm guarantees $\frac{1}{2} - \epsilon$ of the maximum throughput. Under 2-hop interference model, the algorithm achieves an efficiency ratio close to $\frac{1}{1+\Delta}$. Recently, using primary interference model, Sanghavi *et al.* [25] propose a distributed link scheduling algorithm based on matching augmentation. A new interference-free schedule can be generated in less than $4k + 2$ slots, provided that there are enough scheduling initiators in the network, where k is a system parameter. They prove that the proposed algorithm achieves $\frac{k}{k+2}$ of the capacity region, for every $k \geq 1$. In [8], Joo developed a simple distributed scheduling policy that achieves $O(\log |V|)$ complexity by relaxing the global ordering requirement of Greedy Maximal Scheduling (GMS) [9]. It deterministically schedules only links that have the largest queue length among their local neighbors and guarantees a fraction of the optimal performance no smaller than GMS.

Observe that maximum capacity scheduling requires the computing of a MWIS in the conflict graph. Marathe *et al.* [19] propose a simple centralized algorithm with approximation ratio 3 for computing MIS (without weight) in UDG. Harry *et al.* [5] present the first PTAS to approximate the MIS in UDGs. Nieberg *et al.* [21] propose a PTAS for MWIS problem in UDG. Li and Wang [13] further present PTASs for MWIS for a variety of wireless networks. A distributed PTAS approximation for MIS in UDG is proposed in [11].

7. CONCLUSION

In this paper, we address the interference-free link scheduling problem in multihop wireless networks. For networks with or without geometry location, we respectively propose two classes of centralized and distributed scheduling algorithms. We prove that the produced schedulings are stable and achieve any arbitrary fraction of capacity region. In specific, our distributed link scheduling algorithm using geometry location achieves $(1 - \frac{2}{k})^2$ of capacity region for every $k \geq 3$; when geometry location is unavailable, our distributed scheduling algorithm achieves $\frac{1}{1+\epsilon}$ of capacity region. More importantly, all our algorithms generate a valid new schedule by requiring communications within $\Theta(k)$ hops for every node. Additionally, our algorithms incur constant overhead. Moreover, the proposed algorithms using geometry location generate a new valid schedule in constant time-slots. Our methods can be extended to deal with physical interference model such that it has a constant efficiency ratio. The details are omitted due to space limit. It remains a challenge to design an efficient stable scheduling algorithm, without explicitly using geometry locations of nodes, that will run in almost a constant time.

8. REFERENCES

[1] Alzoubi, K., Li, X.-Y., Wang, Y., Wan, P.-J., and Frieder, O. Geometric spanners for wireless ad hoc networks. *IEEE TPDS 14*, 4 (2003), 408–421.

[2] Chaporkar, P., and Proutiere, A. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *MobiCom* (2007), pp. 135–146.

[3] Dhall, S., and Liu, C. On a Real-Time Scheduling Problem. *Operations Research 26*, 1 (1978), 127–140.

[4] G., S., C., J., and Shroff, N. Distributed scheduling schemes for throughput guarantees in wireless networks. *Allerton* (2006).

[5] Harry, I., Hunt, B., Marathe, M., Radhakrishnan, V., Ravi, S., Rosenkratz, D., and Stearns, R. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Networks 25* (1995), 59–68.

[6] Jain, K., Padhye, J., Padmanabhan, V. N., and Qiu, L. Impact of interference on multi-hop wireless network performance. In *MobiCom* (2003), pp. 66–80.

[7] Joo, C., and Shroff, N. Performance of random access scheduling schemes in multi-hop wireless networks. In *IEEE INFOCOM* (2007).

[8] Joo, C. A local greedy scheduling scheme with provable performance guarantee. In *ACM MobiHoc* (2008).

[9] C. Joo, X. Lin, and N. B. Shroff Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-hop Wireless Networks. In *IEEE INFOCOM*, 2008.

[10] Kabbani, A., Salonidis, T., and Knightly, E. Distributed Low-Complexity Maximum-Throughput Scheduling for Wireless Backhaul Networks. In *IEEE INFOCOM* (2007), pp. 2063–2071.

[11] Kuhn, F., Nieberg, T., Moscibroda, T., and Wattenhofer, R. Local approximation schemes for ad hoc and sensor networks. In *DIALM-POMC* (2005), pp. 97–103.

[12] Kumar, P., and Meyn, S. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control, 40*, 2 (1995), 251–260.

[13] Li, X.-Y., and Wang, Y. Simple approximation algorithms and PTASs for various problems in wireless ad hoc networks. *Journal of*

Parallel and Distributed Computing (2005).

[14] LIN, X., AND RASOOL, S. Constant-time distributed scheduling policies for ad hoc wireless networks. In *IEEE CDC* (2006).

[15] LIN, X., AND SHROFF, N. B. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. Netw. 14*, 2 (2006), 302–315.

[16] LIU, Y., AND KNIGHTLY, E. Opportunistic fair scheduling over multiple wireless channels. In *IEEE INFOCOM* , vol. 2.

[17] LU, S., BHARGHAVAN, V., AND SRIKANT, R. Fair scheduling in wireless packet networks. *IEEE/ACM Trans. on Networking (TON) 7*, 4 (1999), 473–489.

[18] LU, S., AND KUMAR, P. Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. on Automatic Control, 36*, 12 (1991), 1406–1416.

[19] MARATHE, M., BREU, H., HUNT III, H., RAVI, S., AND ROSENKRATZ, D. Simple heuristics for unit disk graphs. *Networks 25* (1995), 59–68.

[20] MODIANO, E., SHAH, D., AND ZUSSMAN, G. Maximizing throughput in wireless networks via gossiping. In *ACM SIGMETRICS* (2006), pp. 27–38.

[21] NIEBERG, T., HURINK, J., AND KERN, W. A robust ptas for maximum weight independent sets in unit disk graphs. In *Workshop on Graph-Theoretic Concepts in Compt. Sci.* (2004), pp. 214–221.

[22] PENTTINEN, A., KOUTSOPOULOS, I., AND TASSIULAS, L. Low-complexity distributed fair scheduling for wireless multi-hop networks. In *1st Workshop on Reso. Alloc. in Wireless Netw.* (2005).

[23] RAMABHADRAN, S., AND PASQUALE, J. Stratified round Robin: a low complexity packet scheduler with bandwidth fairness and bounded delay. In *ACM Sigcomm* (2003), pp. 239–250.

[24] RAMANATHAN, S., AND LLOYD, E. Scheduling algorithms for multi-hop radio networks. *IEEE/ACM Trans. on Networking 1* (April 1993), 166–172.

[25] SANGHAVI, S., BUI, L., AND SRIKANT, R. Distributed link scheduling with constant overhead. In *ACM SIGMETRICS* (2007), pp. 313–324.

[26] SHARMA, G., SHROFF, N., AND R., M. R. On the complexity of scheduling in wireless networks. In *ACM MobiCom '06* (2006).

[27] TASSIULAS, L. Linear complexity algorithms for maximum throughtpu in radionetworks and input queued switches. In *INFOCOM* (1998), pp. 533–539.

[28] TASSIULAS, L., AND EPHREMIDES, A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control, 37*, 12 (1992), 1936–1949.

[29] TASSIULAS, L., AND SARKAR, S. Maxmin fair scheduling in wireless networks. In *IEEE INFOCOM* (2002), vol. 2.

[30] VAIDYA, N., DUGAR, A., GUPTA, S., AND BAHL, P. Distributed Fair Scheduling in a Wireless LAN. *IEEE Trans. on Mobile Computing, 4*, 6 (2005), 616–629.

[31] WANG, W., WANG, Y., LI, X.-Y., SONG, W.-Z., AND FRIEDER, O. Efficient interference aware tdma link scheduling for static wireless mesh networks. In *ACM MobiCom* (2006).

[32] YI, S., PEI, Y., AND KALYANARAMAN, S. On the capacity improvement of ad hoc wireless networks using directional antennas. In *ACM MobiHoc* (2003), pp. 108–116.

APPENDIX

Proof of Lemma 8

PROOF. For **TIM** and RTS/CTS model, a link e_i will be removed in a (r, s)-shifting only if its interference area $I(e_i)$ is hit by the lines of $l(e_{p,q})$-squares. Since the diameter of $I(e_{p,q})$ is at most $\frac{1}{(k+1)^{l(e_{p,q})}}$ and the lines of $l(e_{p,q})$-square will move at least $\frac{1}{(k+1)^{l(e_{p,q})}}$ during shifting, each link can be removed at most $2k-1$ times in all k^2 shifting. Similar to the proof of Theorem 3, we can prove that $OPT(L(r,s)) \geq (1-\frac{1}{k})^2 OPT(E)$.

For **fPrIM**, the proof is similar. Note the diameter of $\mathcal{I}(e_{p,q})$ is at most $\frac{2}{(k+1)^{l(e_{p,q})}}$. Thus each link can be removed at most $4k-4$ times in all k^2 shifting. Thus, $OPT(L(r,s)) \geq (1-\frac{2}{k})^2 OPT(E)$. □

Proof of Lemma 6

PROOF. Here we show a constant c_0 for each model.

For **TIM** and RTS/CTS model, c_0 is at most 5. If sender v_p(or receiver if under RTS/CTS model) is covered by 6 or more interference areas, then we can find two senders v_i and v_j(or receivers if under RTS/CTS model) such that $\angle v_i v_p v_j \leq \frac{\pi}{3}$ according to pigeonhole principle. Let $d_{i,j} = \|v_i - v_j\|$. For v_i and v_j, $d_{i,j}^2 \leq d_{i,p}^2 + d_{j,p}^2 - 2\cos\frac{\pi}{3} \cdot d_{i,p} \cdot d_{j,p} = \max\{d_{i,p}, d_{j,p}\}^2 + \min\{d_{i,p}, d_{j,p}\}^2 - \max\{d_{i,p}, d_{j,p}\} \cdot \min\{d_{i,p}, d_{j,p}\} \leq \max\{d_{i,p}, d_{j,p}\}^2 \leq \max(R_{v_i}, R_{v_j})^2$. Thus, these two links with v_i and v_j are not interference free which contradicts our assumption.

For **fPrIM**, c_0 is at most $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$, where $\gamma = \max_i \frac{R_i}{T_i}$. We divide the whole space into $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$ equal cones using $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$ rays from v_p. If v_p is in more than $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$ interference areas, then we can find two receivers v_i and v_j such that $\angle v_i v_p v_j \leq \arcsin\frac{\gamma-1}{2\gamma}$ by pigeonhole principle. Wang *et al.* [31] proved that the two links with v_i and v_j are not interference free in this condition, which contradicts our assumption. □

Proof of Theorem 7

PROOF. We prove it for each interference model separately.

For **TIM**, we first consider all interference free links inside S. Notice that all links in I have level at most j, which implies that the interference area of each link in I has a diameter $\frac{1}{(k+1)^{j+1}}$. Then the distance between the senders of any two interference free links is at least $\frac{1}{(k+1)^{j+1}}$. The j-square S has side length $\frac{k}{(k+1)^j}$. Therefore, there are at most $(\frac{k}{(k+1)^j})^2 / \pi(\frac{1}{2(k+1)^{j+1}})^2 = \frac{4k^2(k+1)^2}{\pi}$ interference free links contained in S by area argument.

Then we concentrate on estimating how many interference free links such that (1) their senders are outside S, (2) with level at most j, (3) each of them may interfere with a link inside S. We show that there are only a constant number of such links by an area argument. Consider the four strips, denoted by $B(S)$, surrounding S with width $\frac{1}{2(k+1)^{j+1}}$. For each link $e_{p,q}$ that interferes with some links inside S, it is not difficult to show that $B(S) \bigcap \mathcal{I}(e_{p,q})$ achieves the smallest area when v_p is on the boundary of $B(S)$ and $R_{v_p} = \frac{1}{2(k+1)^{j+1}}$. The smallest area of $B(S) \bigcap \mathcal{I}(e_{p,q})$ is $\pi\frac{1}{8(k+1)^{2(j+1)}}$. According to Lemma 6, every point in $B(S)$ is covered by at most 5 interference free disks. The area of $B(S)$ is $\frac{2k(k+1)+1}{(k+1)^{2(j+1)}}$. Thus, the number of interference free disks is at most $5 \times \frac{2k(k+1)+1}{(k+1)^{2(j+1)}} / \pi\frac{1}{8(k+1)^{2(j+1)}} = \frac{80k(k+1)+40}{\pi}$.

So the total number of interference free links I with level j and intersecting S is at most $\Lambda = \frac{4k^2(k+1)^2}{\pi} + \frac{80k(k+1)+40}{\pi}$.

For RTS/CTS model, the proof is similar to the proof above. We first consider links inside S. We use the position of the endpoint with larger interference radius to represent the position of a link. The distance between any two interference free links is at least $\min_{e_{p,q}\in I}(L(e_{p,q})) \geq \frac{1}{(k+1)^{j+1}}$ under RTS/CTS model. Therefore, there are at most $(\frac{k}{(k+1)^j})^2 / \pi(\frac{1}{2(k+1)^{j+1}})^2 = \frac{4k^2(k+1)^2}{\pi}$ interference free links contained in S.

Then we consider links outside S. Consider the same strips $B(S)$ as above, surrounding S with width $\frac{1}{2(k+1)^{j+1}}$. According to Lemma 6, every point in $B(S)$ is covered by at most 5 interference free links. Using the same area argument as above, the number of interference free links is at most $5\times\frac{2k(k+1)+1}{(k+1)^{2(j+1)}} \Big/ \pi\frac{1}{8(k+1)^{2(j+1)}} = \frac{80k(k+1)+40}{\pi}$.

So the total number of interference free links I with level j and intersecting S is at most $\Lambda = \frac{4k^2(k+1)^2}{\pi} + \frac{80k(k+1)+40}{\pi}$.

For **fPrIM**, the proof is also similar. Consider the same strips $B(S)$ as above, surrounding S with width $\frac{1}{2(k+1)^{j+1}}$. According to Lemma 6, every point in $B(S)$ is covered by at most $\lceil\frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}}\rceil$ interference free links. The area of $B(S)\bigcup S$ is $\frac{1}{(k+1)^{2(j-1)}}$. Each link covers at least an area of $\pi\frac{1}{8(k+1)^{2(j+1)}}$. The smallest area is achieved when the sender is on the boundary and the interference radius is $\frac{1}{2(k+1)^{j+1}}$. So the total number of interference free links I with level j and intersecting S is at most $\Lambda = \lceil\frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}}\rceil \times \frac{1}{(k+1)^{2(j-1)}} \Big/ \pi\frac{1}{8(k+1)^{2(j+1)}} = \lceil\frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}}\rceil \times \frac{8(k+1)^4}{\pi}$. $\square$