Throughput Optimizing Localized Link Scheduling for Multihop Wireless Networks Under Physical Interference Model

Yaqin Zhou, Xiang-Yang Li, *Senior Member, IEEE,* Min Liu, *Member, IEEE*, Xufei Mao, *Member, IEEE*, Shaojie Tang, *Member, IEEE*, Zhongcheng Li, *Member, IEEE*

***Abstract*—We study throughput-optimum localized link scheduling in wireless networks. The majority of results on link scheduling assume binary interference models that simplify interference constraints in actual wireless communication. While the physical interference model reflects the physical reality more precisely, the problem becomes notoriously harder under the physical interference model. There have been just a few existing results on link scheduling under the physical interference model, and even fewer on more practical distributed or localized scheduling. In this paper, we tackle the challenges of localized link scheduling posed by the complex physical interference constraints. By integrating the partition and shifting strategies into the pick-and-compare scheme, we present a class of localized scheduling algorithms with provable throughput guarantee subject to physical interference constraints. The algorithm in the oblivious power setting is the first localized algorithm that achieves at least a constant fraction of the optimal capacity region subject to physical interference constraints. The algorithm in the uniform power setting is the first localized algorithm with a logarithmic approximation ratio to the optimal solution. Our extensive simulation results demonstrate performance efficiency of our algorithms.**

***Index Terms*—localized link scheduling, physical interference model, maximum weighted independent set of links (MWISL), capacity region.**

I. Introduction

As a fundamental problem in wireless networks, link scheduling is crucial to improve network performances through maximizing throughput and fairness. It has recently regained much interest from networking research community because of wide deployment of multihop wireless networks, *e.g.*, wireless sensor networks for monitoring physical or environment [2] [3] through collection of sensing data [4]. Generally, link scheduling involves determination of which links should transmit at what times, what modulation and coding schemes to use, and at what transmission power levels should communication take place [5]. In addition to its great significance in wireless networks, developing an efficient scheduling algorithm is extremely difficult due to the intrinsically complex interference among simultaneously transmitting links in the network.

The link scheduling problem has been studied with different optimization objectives, *e.g.*, throughput-optimum scheduling, minimum length scheduling. Our study mainly focuses on maximizing throughput in multihop wireless networks. Taking queue length of every link as its weight, it is well known that a throughput-optimum scheduling policy that tries to find a maximum weighted independent set of links is generally NP-hard in wireless networks [5].

Despite of numerous results gained for the problem [6], [5], [7], [8], [9], [10], most assume simple binary interference models, *e.g.*, hop-based, range-based, and protocol interference models [11]. Under this category of interference model, a set of links are conflict-free if they are pairwise conflict-free. Conflict of transmissions on two distinct links is predetermined independently of the concurrent transmissions of other links. Thus, interference relationships based on these models can be represented by conflict graphs, and we can leverage classic graph-theoretical tools for solutions. However, in actual wireless communication, interference constraints among concurrent transmissions are not local and pairwise, but global and additive. Conflict of distinct transmissions is determined by the cumulative interference from all concurrent transmissions, which is often depicted by the physical interference model, *e.g.*, the Signal-to-Interference-plus-Noise Ratio (SINR) interference model. The global and additive nature of the physical interference model drives previous traditional techniques based on conflict graphs inapplicable or trivial. Consequently, designing and analyzing scheduling algorithms under the physical interference model becomes especially challenging.

Some recent research results [12], [11], [13], [14], [15], [16], [17], [18], [19] have addressed some related challenges. To the best of our knowledge, however, all of these, with throughput maximization or other optimization objectives such as a minimum length schedule [11], just focus on centralized implementation. Distributed or even localized scheduling un-

The research of authors is partially supported by NSF CNS-0832120, NSF CNS-1035894, NSF ECCS-1247944, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202,No. 61132001, No. 61120106008, No. 61070187, No.61272474, No. 61202410, No.61272475 and No.61003225, China 973 Program under Grant No.2011CB302705. A previous conference vision, “Distributed Link Scheduling for Throughput Maximization under Physical Interference Model” [1], has been published in Infocom 2012. It mainly focuses on theoretical analysis in oblivious power setting.

Yaqin Zhou, Min Liu, and Zhongcheng Li are with Research Center of Network Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

Xiang-Yang Li, contact author, is with TNLIST, School of Software, Tsinghua University, Beijing, China, and Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA.

Shaojie Tang is with Department of Computer and Information Science (Research) at Temple University, Philadelphia, PA, USA.

Xufei Mao is with TNLIST, School of Software, Tsinghua University, Beijing, China.

der the physical interference model is more demanding out of practical relevance.

Though [20], [21], [22] consider distributed implementation of centralized algorithms, they require global propagation of messages and [20] [21] fail to provide an effective localized scheduling algorithm with satisfactory theoretical guarantee and. A scheduling algorithm without theoretical guarantee may cause arbitrarily bad throughput performance, and global propagation of messages throughout network is inefficient in terms of time complexity [23], especially for large-scale networks. This motivates us to develop localized link scheduling algorithms with provable throughput performance. Here by a localized scheduling algorithm we mean that each node only needs information (*i.e.*, queue length and link status) within constant distance to make scheduling decisions, while a distributed algorithm may inexplicitly need information far away. Since just local information is available for each node to collaborate on schedulings with globally coupled interference constraint, it poses significant challenges to designing efficient localized scheduling algorithms with theoretical throughput guarantee.

In this paper we tackle these challenges of practical localized scheduling for throughput maximization under physical interference with the commonly-used oblivious and uniform power assignment. We prove that our algorithms can respectively achieve a constant and $O(\log|V|)$ fraction of capacity region for the oblivious and uniform power setting, where $|V|$ is the number of nodes. Our extensive simulation shows that our proposed algorithms outperform simple heuristics.

The remainder of the paper is organized as follows. We define the network model and problems in Section II, propose our localized algorithms for the oblivious and uniform power assignment respectively in Section III and IV, and evaluate their performance in Section V. We review related works in Section VI, and conclude our work in Section VII.

II. Models and Assumptions

A. Network Communication Model

We model a wireless network by a two-tuple (V, E), where V denotes the set of nodes and E denotes the set of links. Each directed link $l = (u, v) \in E$ represents a communication request from a sender u to a receiver v. Let $\|l\|$ or $\|uv\|$ denote the length of link l. We assume each node knows its own location and the topology of the network.

B. Interference model

Under the physical interference model, a *feasible schedule* is defined as an independent set of links (ISL), each satisfying

$$SINR_{uv} \triangleq \frac{P_u \cdot \eta \cdot \|uv\|^{-\kappa}}{\sum_{w \in T_u} P_w \cdot \eta \cdot \|wv\|^{-\kappa} + \xi} \geq \sigma,$$

where ξ denotes the ambient noise, σ denotes certain threshold, and T_u denotes the set of simultaneous transmitters with u. It assumes path gain $\eta \cdot \|uv\|^{-\kappa} \leq 1$, where the constant $\kappa > 2$ is path-loss exponent, and η is the reference loss factor.

We consider the following two transmission power settings.

1) *Oblivious power setting:* a sender u transmits to a receiver v always at the power $P_u = c \cdot \|uv\|^{\beta}$ where c and β are both constant satisfying $c > 0, 0 < \beta \leq \kappa$.
2) *Uniform power setting:* all links always transmit at the same power $P_u = P$. Under the uniform power case, we further assume that all links have a length adequately less than the maximum transmission radius $\sqrt[\kappa]{\frac{\eta P}{\sigma \xi}}$ as links with length almost the same as the maximum transmission radius are vulnerable to fail.

We use P_l and P_u alternatively to denote the transmitting power of link $l = (u, v)$. The distance between u and v satisfies $r \leq \|uv\| \leq R$, where r and R respectively denotes the shortest link length and the longest link length. We suppose that r and R are known by each node.

C. Traffic models and scheduling

The maximum throughput scheduling is often studied in the following models. It assumes time-slotted wireless systems, and single-hop flows with stationary stochastic packet arrival process at an average arrival rate λ_l. The vector $A(t) = \{A_l(t)\}$ denotes the number of packets arriving at each link in time slot t. Every packet arrival process $A_l(t)$ is assumed to be i.i.d over time. We also assume that all packet arrival processes $A_l(t)$ have bounded second moments and they are bound by $A_{\max}$, *i.e.*, $A_l(t) \leq A_{\max}, \forall l \in E$. Let a vector $\{0, 1\}^{|E|}$ denote a schedule $S(t)$ at each time slot t, where $S_l(t) = 1$ if link l is active in time slot t and $S_l(t) = 0$ otherwise. Packets departure transmitters of activated links at the end of time slots. Then, the *queue length* vector $Q(t) = \{Q_l(t)\}$ evolves as $Q(t+1) = \max\{0, Q(t) - S(t)\} + A(t+1)$ where $Q_l(t)$ is queue length (*weight* or *backlog*) of link l.

The throughput performance of link scheduling algorithms is measured by a set of supportable arrival rate vectors, named *capacity region* or *throughput capacity*. That is, a scheduling policy is *stable*, if for any arrival rate vector in its capacity region [10], it satifies $\lim_{t \to \infty} \mathbb{E}[Q(t)] < \infty$.

Though the policy of finding a MWISL to schedule regarding to the underlying interference models is *throughput-optimal* [24], finding a MWISL itself is NP-hard generally [25]. Thus we have to rely on approximation or heuristic methods to develop suboptimal scheduling algorithms running in polynomial time. A suboptimal scheduling policy can achieve a fraction of the optimal capacity region depicted by efficiency ratio γ [5].

A suboptimal scheduling policy with efficiency ratio γ must find a γ-approximation scheduling at every time slot t to achieve γ times of the optimal capacity region [26]. It remains difficult to achieve in a decentralized manner. The pick-and-compare approach proposed in [6] enables that we just need to find a γ-approximation scheduling with a constant positive probability. The basic pick-and-compare [6] works as follows. At every time slot, it generates a feasible schedule that has a constant probability of achieving the optimal capacity region. If the weight of this new solution is greater than the current solution, it replaces the current one. Using this approach achieves the optimal capacity region. The proposition below further extends this approach to suboptimal cases.

Proposition 1: ([8]) Given any $\gamma \in (0,1]$, suppose that an algorithm has a probability at least $\delta > 0$ of generating an independent set $\mathcal{X}(t)$ of links with weight at least γ times the weight of the optimal. Then, capacity $\gamma \cdot \Lambda$ 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 $S(t)$ from the old scheduling $S(t-1)$ and current queue length $Q(t)$.

III. The Algorithm in the oblivious power Setting

In this section, we focus on the design of localized scheduling algorithm for the oblivious power setting.

A. Basic idea

The basis of our idea is to create a set of disjoint local link sets where scheduling can be done independently without violating the global interference constraint. The decoupling of the global interference constraint is based on the fact that distance dominates the interference between two distinct links. That is, if a transmitting link is placed a certain distance away from all the other transmitting links, the total interference it receives may get bounded. Based on this, then we employ the partition strategies to divide the network graph into disjoint local areas such that each local area is separated away by a certain distance to enable independent local computation of schedulings inside every local area. Links lying outside local areas will keep silent to ensure separation of local areas. As links lying outside local areas cannot remain unscheduled all the time or it will induce network instability, we use the shifting strategy to change partitions at every time slot to make sure that every backlogged link will be scheduled. These locally computed scheduling link sets compose a new global schedule $\mathcal{X}(t)$ at every time slot t.

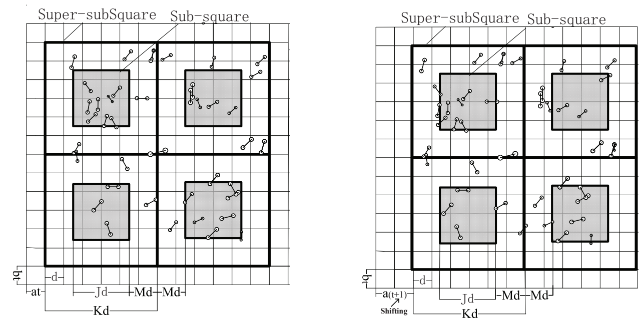
As the pick-and-compare scheme, we choose a more weighted schedule, denoted as $S(t)$, between a newly generated schedule $\mathcal{X}(t)$ and the last-time schedule $S(t-1)$ using $Q(t)$. Meanwhile, by Proposition 1, if we guarantee that

$$\mathbb{P}(S(t) \cdot Q(t) \geq \gamma S^*(t) \cdot Q(t)) \geq \delta \tag{1}$$

for some constant $\gamma > 0,\ \delta > 0$, the queue length vector $Q(t)$ will eventually converge to a stable state.

B. Detailed description

We first describe the partition and shifting strategies [9], as illustrated in Fig. 1. The plane is partitioned into cells with side length $d = R$, by horizontal lines $x = i$ and vertical lines $y = j$ for all integers i and j. A vertical strip with index i is $\{(x, y)|i < x \leq i+1\}$. Similarly, we define the horizontal strip j. cell(i, j) is the intersection area of a vertical strip i and a horizontal strip j. A super-subSquare(i, j) is the set of cells: $\{cell(x, y)|x \in [i * K + a_t, (i+1) * K + a_t), y \in [j * K + b_t, (j+1) * K + b_t)\}$, and a sub-square$(i, j)$ inside it is the set of cells:$\{cell(x, y)|x \in [i * K + a_t + M, (i+1) * K + a_t - M), y \in [j * K + b_t + M, (j+1) * K + b_t - M)\}$. The corresponding link set Y_{ij} (or L_{ij}) consists of links with both ends inside super-subSquare(i, j) (or sub-square(i, j)). Let constant $K = 2M + J$, where M is a constant that would



(a) Partition(K, a_t, b_t). Here the gray area is the local area the link set of which participate in computing the new schedule; the links in the white area keep silent.

(b) Partition$(K, a_{(t+1)}, b_t)$. Change the partition through shifting to the right by one cell, ensuring that links in the white area in previous partitions have opportunity to be scheduled.

Fig. 1: The partition and shifting process

be defined in Lemma 2. Let integers $a_t,\ b_t, 0 \leq a_t, b_t < K$ be the *horizontal* and *vertical shifting* respectively, then we call the resulting plane *Partition*(K, a_t, b_t). By separately adjusting $a_t,\ b_t$, we can get K^2 different partitions for a plane totally.

At each time slot each nodes runs Algorithm 1 to collaboratively compute a globally feasible scheduling. As every node knows the locality from which it will collect information, it then participates the corresponding local computation, and at last it sends (if it is a coordinator) or receives (if not) the results.

Time slot $t = 0$: Every node first decides in which cell it resides when $(a_0, b_0) = (0, 0)$; then it participates in the process of computing a local scheduling $S_{ij}(0)$ for the sub-square(i, j) it belongs to. Let the solution $S(0)$ be the union of the local solutions $S_{ij}(0)$ for all sub-squares.

Time slot $t \geq 1$: Every node decides in which cell it resides by a partition starting from (a_t, b_t). The shifting strategy for a_t and b_t works as follows. We let $a_t = t \mod K$; and $b_t = (b_t + 1) \mod K$ if $a_t = 0$, or it keeps unchanged. Each node then participates in computing the new local scheduling, denoted as $\mathcal{X}_{ij}(t)$, for its sub-square(i, j) using the weight $Q(t)$. Let $S_{ij}(t-1)$ be the set of links from $S(t-1)$(the global solution at time slot $t-1$) falling in the super-subSquare(i, j) instead of sub-square(i, j). If $S_{ij}(t-1) \cdot Q(t) > \mathcal{X}_{ij}(t) \cdot Q(t)$, let $S_{ij}(t) = S_{ij}(t-1)$, else $S_{ij}(t) = \mathcal{X}_{ij}(t)$, the global solution is the union of $S_{ij}(t)$ from all super-subSquares.

In our algorithm, a_t increases from 0 to $K-1$ in K sequent time slots when b_t is fixed, and b_t increases by 1 every K time slots. The vertical and horizontal distance between any two sequent sub-squares is $2M$. Initially, both a_t and b_t are 0. Thus in the worst case it takes $2M \cdot K$ time slots for a uncovered link to lie between two sequent sub-squares in horizontal line, i.e., increase b_t by $2M$; and another $2M$ time slots to be finally covered by a sub-square, i.e., increase a_t by $2M$. Thus we say it will take $(K+1)2M$ time slots to get every link covered by a sub-square. The vertical and horizontal distance between any two sequent super-subSquares is 0. A link crosses two vertical and horizontal cells at most. In worst case, it requires

TABLE I: Summary of notations

J	side length of sub-square	L_{ij}	link set of sub-square(i,j)	Y_{ij}	link set of super-subSquare(i,j)
K	side length of super-subSquare	$\mathcal{X}_{ij}(t)$	new scheduling for L_{ij} at t	$S_{ij}(t)$	scheduling for Y_{ij} at t
Z_i	local link set	$OPT_{ij}(t)$	local optimal MWISL for L_{ij}	$S^*(t)$	global optimal MWISL at t
R	longest link length	$S^*_{ij}(t)$	intersection of L_{ij} and $S^*(t)$	$S(t)$	global scheduling at time slot t
d	side length of cell	$\|uv\|$	link length	$r_S(l)$	relative interference l get from link set S
$a_S(l)$	affectness l get from link set S	$Q(t)$	queue length vector	$W(S)$	weight of link set S
I^l_S	interference link l suffered from link set S	I^l_{max}	the maximum interference l can bear	I_{max}	the minimum of I^l_{max}

K time slots to increase b_t by 1, and another 1 time slot to increase a_t by 1, where a uncovered link finally gets covered by a super-subSquare.

Algorithm 1 Distributed Scheduling by node v under the oblivious power setting

state = White; active = No; Coordinator = No;
Calculate which cell node v resides in regarding to the current partition(K, a_t, b_t);
if v is the closest node to the center of super-subSquare **then**
  Coordinator = Yes;
end if
if Coordinator = Yes **then**
  Collect $Q(t)$ and $S_{ij}(t-1)$.
  Compute $\mathcal{X}_{ij}(t)$ in sub-square(i,j) by enumeration;
  if $S_{ij}(t-1)\cdot Q(t) > \mathcal{X}_{ij}(t)\cdot Q(t)$ **then**
    $S_{ij}(t) = S_{ij}(t-1)$;
  else
    $S_{ij}(t) = \mathcal{X}_{ij}(t)$;
  end if
  Broadcast RESULT$(S_{ij}(t))$ in super-subSquare(i,j);
end if
if state = White **then**
  if receive message RESULT$(S_{ij}(t))$ **then**
    if $v \in S_{ij}(t)$ **then**
      state = Red; active = Yes;
    else
      state = Black; active = No;
    end if
  end if
end if

C. Theoretical analysis and proof

Given a network (V,E), supposing $\cup Z_i$ is a set of disjoint local link sets inside for scheduling, where $Z_i \in E$ and $Z_i \cap Z_j = \phi$ if $i \neq j$, for any link $l \in Z_i$, if l is activated, then

$$I^l = I^l_{in} + I^l_{out}$$

where I^l denotes cumulative interference from all other activated links in the network, I^l_{in} denotes the total interference from simultaneously transmitting links inside Z_i and I^l_{out} denotes the total interference from transmissions outside.

Therefore, we can do independent scheduling inside Z_i without consideration of I^l_{out} from concurrent transmissions outside Z_i, if I^l_{out} gets bounded by a constant, *i.e.*,

$$I^l_{in} \le (1-\varepsilon)\cdot I^l_{\max},\ I^l_{out} \le \varepsilon \cdot I_{\max},\ 0<\varepsilon<1,\ l \in Z_i.$$

Let $I_{\max}$ denote the maximum interference that the longest links in E can tolerant during a successful transmission, and $I^l_{\max}$ represent the maximum interference that an activated link l can tolerant during a successful transmission. Then we have the two Lemmas bellow.

Lemma 1: In the oblivious power setting, the number of independent links for a local link set Z_i inside a square with a size length JR is bounded by a constant. Let OPT_i be a local MWISL of Z_i, $|OPT_i| \le \frac{(\sqrt{2}JR)^\kappa}{(1-\varepsilon)}\left[\frac{1}{\sigma} - \frac{\xi\cdot r^{\beta-\kappa}}{c\eta}\right]+1$.

Proof: This proof is available in Appendix A. ■

Lemma 2: Under the oblivious power setting, if the Euclidean distance between any two disjoint local link sets is at least $M\times R$, then activated links in each local link set suffer a bounded cumulative interference from all other activated link sets, *i.e.*, for each activated link l in local link set Z_i,

$$I^l_{out} \le \varepsilon\cdot I_{\max},\ 0<\varepsilon<1,$$

where M is a constant, satisfying $M \ge \left[\frac{2\pi c\eta R^{\beta-\kappa}\cdot|OPT_i|_{ub}}{(\kappa-2)\varepsilon I_{\max}}\right]^{\frac{1}{\kappa}}$. Herein $|OPT_i|_{ub}$ denotes an upper bound of the size of the local optimal MWISL for Z_i.

Proof: This proof is available in Appendix B. ■

To analyze the theoretical performance of our method, we first review the following definitions.

Definition 2: (affectness [16]) The relative interference of link l^* on l is the increase caused by l^* in the inverse of the SINR at l, namely $r_{l^*}(l) = I^l_{l^*}/(P_l\eta\|uv\|^{-\kappa})$. For convenience, define $r_l(l)=0$. Let $c_l = \frac{\sigma}{1-\sigma\xi/(P_l\eta\|uv\|^{-\kappa})}$ be a constant that indicates the extent to which the ambient noise approaches the required signal at receiver t_v. The affectness of link l caused by a set S of links, is the sum of relative interference of the links in S on l, scaled by c_l, or $a_S(l) = c_l\cdot\sum_{l^*\in S} r_{l^*}(l)$.

*Definition 3: (p-*signal *set [16]) We define a p-*signal *set to be one where the affectness of any link is at most $1/p$. Clearly, any ISL is a 1-*signal *set.*

Lemma 3: ([16]) There is a polynomial-time protocol that takes a p-signal set and refines into a p'-signal set, for $p'>p$, increasing the number of slots by a factor of at most $4(\frac{p'}{p})^2$. It indicates that a p-signal set can be refined into at most $4(\frac{p'}{p})^2$ p'-signal set through a polynomial-time algorithm, e.g., a first-fit algorithm. Using the result, we have the Lemma below.

Lemma 4: The weight of $\mathcal{X}_{ij}(t)$ has a constant approximation ratio to the weight of the intersection set by the local link set L_{ij} and the global optimal MWISL $S^*(t)$.

Proof: Normally any ISL is a 1-signal set. That is, for the affectness of a normal ISL it holds that:

$$a_S(l) = c_l\cdot\sum_{l^*\in S} r_{l^*}(l) \le \frac{\sigma}{1-\sigma\xi/P_l}\cdot\frac{I^l_{\max}}{P_l} \le 1, \quad (2)$$

whereas the affectness of a locally computed ISL for sub-

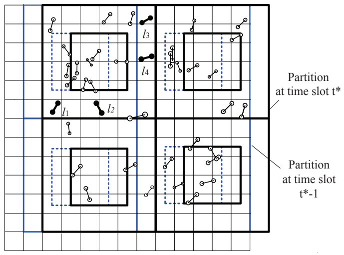


Fig. 2: Partitions at time slot t^*-1 and time slot t^*

square must satisfy that,

$$a_{S_{ij(t)}}(l) = c_l \cdot \sum_{l^* \in S_{ij(t)}} r_{l^*}(l) \leq \frac{\sigma \cdot (1-\varepsilon)}{1-\sigma\xi/P_l} \cdot \frac{I^l_{\max}}{P_l} \leq 1-\varepsilon. \quad (3)$$

Therefore, by Lemma 3, a normal MWISL can be refined into $\frac{4}{(1-\varepsilon)^2}$ $\frac{1}{(1-\varepsilon)}$-Signal link sets at most. Since the $\frac{1}{(1-\varepsilon)}$-Signal link set returned by enumeration is most weighted, so the locally computed link scheduling sets $\mathcal{X}_{ij}(t)$ has a weight

$$W(\mathcal{X}_{ij}(t)) \geq \frac{(1-\varepsilon)^2}{4} W(OPT_{ij}(t)). \quad (4)$$

Let $S^*_{ij}(t) = S^*(t) \cap L_{ij}$ denote the intersection by L_{ij} and $S^*(t)$, where $S^*(t)$ is the global optimal MWISL at time slot t. It is obvious that
$W(S^*_{ij}(t)) \leq W(OPT_{ij}(t)) \leq \frac{4}{(1-\varepsilon)^2} W(\mathcal{X}_{ij}(t))$. ■

Theorem 1: $S(t) = \cup S_{ij}(t)$ computed by our algorithm is an independent link set under the physical interference model in the oblivious power setting. The weight of $S(t)$, *i.e.*, $W(S(t))$, is a constant approximation of the weight of the global optimal MWISL with probability of at least $1/K^2$.

Proof: The proof consists of two phases. We first prove that $S(t) = \cup S_{ij}(t)$ is an independent set. We next derive the approximation bound that $S(t)$ achieves.

Phase I: We rely on induction to infer that at every time slot $S(t)$ is a union of disjoint activated local link sets that are separated by at least M cells from each other. Then we have that $S(t)$ is an independent link set under the physical interference model by Lemma 2. The following are the details.

For any link $l \in S(t)$, assuming $l \in S_{ij}(t)$, the total interference l suffers from all the other simultaneously transmitting links in $S(t)$ is denoted by $I^l_{S(t)}$.

At time slot 0, every local activated link set $S_{ij}(0) = \mathcal{X}_{ij}(0)$ is kept $2M$ cells away from each other, so $S(0)$ is an independent set by Lemma 2.

At time slot 1, either $S_{ij}(0)$ or $\mathcal{X}_{ij}(1)$ is chosen to be a part of $S(1)$. For those super-subSquares whose $S_{ij}(1) = S_{i'j'}(0) \cap Y_{ij}(0)$, their distance is kept at least $2M$ cells away. For those super-subSquares whose $S_{ij}(1) = \mathcal{X}_{ij}(1)$, their distance is also kept at least $2M$ cells away. And the distance between the two kinds of link set is at least $2M-1$ cells away. So $S(1)$ is an independent set.

At some time slot t^*, $t^* > 1$, for some super-subSquares, $S_{ij}(t^*)$ consists of disjoint subsets from several different $S_{i'j'}(t^*-1)$ which fall into $Y_{ij}(t^*)$, *i.e.*,

$$S_{ij}(t^*) = S(t^*-1) \cap Y_{ij}(t^*) = \bigcup_{i'j'} \{S_{i'j'}(t^*-1) \cap Y_{ij}(t^*)\}. \quad (5)$$

For instance, as illustrated in Fig. 2, link sets $\{l_1, l_2\}$ and $\{l_3, l_4\}$ respectively get scheduled in two different super-subSquares (*i.e.*, super-subSquare$(1,2)$ and super-subSquare$(2,2)$) at time slot t^*-1, the links $\{l_1, l_2, l_3, l_4\}$ are then get scheduled in the same super-subSquare$(1,2)$ at time slot t^*. Let $\Phi^{ij}_{i'j'}(t^*) = S_{i'j'}(t^*-1) \cap Y_{ij}(t^*)$ for brevity. Each $S_{i'j'}(t^*-1)$ is kept at least M cells away from each other, so is each $\Phi^{ij}_{i'j'}$.

Clearly, $S(t^*)$ can be divided into two separated subsets, one formed by some subsets of $S(t^*-1)$, the other formed by newly computed $\mathcal{X}_{ij}(t^*)$, *i.e.*,

$$S(t^*) = \bigcup_{ij} S_{ij}(t^*) = \left\{ \bigcup_{pq} \bigcup_{i'j'} \Phi^{pq}_{i'j'}(t^*) \right\} \bigcup \left\{ \bigcup_{\substack{mn, \\ mn \neq pq}} \mathcal{X}_{mn}(t^*) \right\}. \quad (6)$$

Since $\bigcup_{pq} \bigcup_{i'j'} \Phi^{pq}_{i'j'}(t^*)$ is a subset of $S(t^*-1)$, it is composed by disjoint subsets with a mutual distance of M cells at least. The distance between any distinct $\mathcal{X}_{mn}(t^*)$ is no less than $2M$ cells. Then we consider the distance between a disjoint subset of $\bigcup_{pq} \bigcup_{i'j'} \Phi^{pq}_{i'j'}(t^*)$ and a $\mathcal{X}_{mn}(t^*)$. Since $\mathcal{X}_{mn}(t^*)$ locates in sub-square(m,n), which is M cells away from the border of super-subSquare(m,n), the distance between a disjoint subset of $\bigcup_{pq} \bigcup_{i'j'} \Phi^{pq}_{i'j'}(t^*)$ and a $\mathcal{X}_{mn}(t^*)$ is still no less than M cells. Comprehensively, $S(t^*)$ consists of disjoint subsets which are separated by at least M cells.

Note that a disjoint subset of $S(t^*)$ does not equalize to a $S_{ij}(t^*)$ since a $S_{i'j'}(t^*-1)$ may be reserved completely in different super-subSquares at time slot t^*. Here we denote the disjoint subset by $\psi_i(t^*)$, and $S(t^*) = \bigcup \psi_i(t^*)$.

By Lemma 2, for each link $l \in \psi_i(t^*)$, where $\psi_i(t^*)$ comes from the former part of equation (6), we have $I^l_{S(t^*)} \leq I^l_{\max}$. Meanwhile, for each $l \in \psi_i(t^*)$, where $l \in \psi_i(t^*)$ comes from the later part of (6), it holds that $I^l_{S(t^*)} \leq I^l_{\max}$. Then we have

$$I^l_{S(t^*)} \leq I^l_{\max}, \forall l \in S(t^*), \quad (7)$$

indicating that $S(t^*)$ is an independent set.

Next we consider situations at time slot t^*+1. Similarly, $S(t^*+1)$ composes of disjoint subsets separated by no less than M cells. Using the same technique as at time slot t^*, we can get that $S(t^*+1)$ is still an independent set.

By induction we can conclude that $S(t)$ is a union of disjoint activated subsets separated by M cells at least, thus it is an independent set. Herein we finish the first phrase of the proof.

Phase II: We derive the approximation ratio by the pigeonhole principle, Lemma 4, and Proposition 1.

Let $D(t)$ denote the link set of the removed strips at Partition(K, a_t, b_t). $D^*(t)$ represents a subset of $S^*(t)$, links of which fall inside $D(t)$, *i.e.*, $D^*(t) = D(t) \cap S^*(t)$. Recalling that there are K^2 different partitions for a plane totally. If we tried all these partitions in a single slot, each cell(i,j) would appear in the "removed" strips at most $2KM$ times. Then we let $D_i(t)$ denote the link set of the removed strips when the i_{th} one of the K^2 partitions happens, and let $D^*_i(t) = D_i(t) \cap S^*(t)$. Thus the weight of all $D^*_i(t)$ in the K^2 partitions should be $2KMW(S^*(t))$, *i.e.*,

$$\sum_{\mathrm{i}=0}^{K^2-1} W(D^*_i(t)) \leq 2KMW(S^*(t)). \quad (8)$$

Since actually we only experience one partition at time slot t, by the pigeonhole principle the corresponding Partition(K, a_t, b_t) has a probability at least $1/K^2$ to be an *optimal partition*, the weight of whose removed links in $D^*(t)$ is no greater than that of other partitions. Instantly we have the following with probability of $1/K^2$ at least,

$$W(D_i^*(t)) \leq \frac{1}{K^2} \cdot \sum_{i=0}^{K^2-1} W(D_i^*(t)) \leq \frac{2M}{K} W(S^*(t)), \quad (9)$$

$$W(\cup S_{ij}^*(t)) = W(S^*(t) \backslash D^*(t)) \geq (1 - \frac{2M}{K}) W(S^*(t)). \quad (10)$$

Following Lemma 4, with a probability of $1/K^2$ it holds,

$$(1 - \frac{2M}{K}) W(S^*(t)) \leq \frac{4}{(1-\varepsilon)^2} W(\cup \mathcal{X}_{ij}(t)). \quad (11)$$

So by Proposition 1 we get that the approximation ratio for the optimal is $\frac{4}{(1-2\cdot\frac{M}{K})(1-\varepsilon)^2}$. ■

D. Time and communication complexity

We define the *time complexity* as the time units required by the running of Algorithm 1 inside each local area in the worst case [23]. It includes time units for local information collection and local computation time at the center node. At every time slot, the coordinators shall collect queue information and last scheduling status of all links inside the super-subSquares. After computation, the coordinators broadcast the scheduling results throughout the super-subSquares.

It may require multihop propagation to collect and broadcast the needed information inside super-subSquares. To avoid collision, the coordinator can first compute a tree that determines the sequence of transmissions at each node in the super-subSquare based on topology information already known. In the worst case each node has to transmit one by one at different mini slot, then it causes time complexity $O(n_{ij}^2)$ where n_{ij} is the number of nodes inside super-subSquare(i, j) in the worst case. Herein we just give a basic scheme, the time complexity may be further reduced with better broadcast scheduling.

After all required information gathered, the local computation complexity at the center node is $O(2^{|L_{ij}|})$. Since the time units for computation process is much smaller than the time unit for message propagation, the local computation complexity can be ignored comparing to the time for information collection. Thus the total time complexity is $O(n_{ij})$. The *communication complexity* is the total number of basic messages transmitted during each scheduling in the worst case [23]. The communication complexity is $O(|V|)$, and the number of messages transmitted in each local area is $O(n_{ij})$.

IV. The Algorithm in the Uniform Power Setting

We now extend the framework to the uniform power setting. Instead of enumeration, we compute a MWISL of the candidate links for each sub-square by adopting the method proposed in [19], as the cardinality of the optimal MWISL in each sub-square is no longer bounded by a constant in the uniform power setting. Except for the difference, the general structure of the algorithm is the same with that of the oblivious power setting.

We first describe the main idea of Algorithm 1 in [19] for computation of MWISL inside each sub-square. Given a set of links and weights associated with the links, the algorithm works as follows:

Phase I: Remove every link whose associated weight is at most $\frac{w_{\max}}{n}$ where $w_{\max}$ denotes the maximum weight among all links and n is the number of all given links. Let $w_{\min}$ denote the minimum weight from the remaining links.

Phase II: Partition the remaining links into $\log \frac{w_{\max}}{w_{\min}}$ groups such that the links of the $i-$th group G_i have weights within $[2^i w_{\min}, 2^{i+1} w_{\min}]$. For each group G_i of links, it finds an independent set of links among it by adopting the method in [27]. Totally it will get $\log \frac{w_{\max}}{w_{\min}}$ ISLs, one for each group. Then it chooses the one with the maximum weight among the $\log \frac{w_{\max}}{w_{\min}}$ ISLs as the final solution.

As the resulted link scheduling set is an ISL by the method in [27], we can prove that its size has a constant upper bound through the following lemma.

Lemma 5: ([27]) Consider a link $l = (u, v)$ and a set N of nodes other than u whose distance from u is at most $\rho\|uv\|$. If link l succeeds in the presence of the interference from N, then $|N| \leq \frac{(\rho+1)^\kappa}{\sigma} \left[1 - (\frac{\|uv\|}{R})^\kappa\right]$.

The corollary below asserts an upper bound of the number of successfully transmitting links in a local link set with size $JR \times JR$ when utilizing Algorithm 1 in [19].

Corollary 1: The cardinality of the resulted link scheduling set by Algorithm 1 in [19] is upper bounded, *i.e.*, $|\mathcal{X}_{ij}(t)| \leq \frac{(\sqrt{2}J\frac{R}{r}+1)^\kappa}{\sigma} \left[1 - (\frac{r}{R})^\kappa\right]$ within an area of $JR \times JR$.

Proof: In the method presented in [27], it adds firstly the shortest link among all the candidates to the scheduling set. And the distance between any pair of nodes will be no greater than $\sqrt{2}JR$. Therefore, let r be the shortest link length and then $\rho = \frac{\sqrt{2}JR}{r}$, we derive the upper bound. ■

We let $|\mathcal{X}_{ij}(t)|_{ub}$ denote an upperbound of the size of the local computed ISL for Z_i by Algorithm 1 in [19]. Then we present the lemma below .

Lemma 6: In the uniform power setting, if the Euclidean distance between any two disjoint local link sets is at least $M \times R$, then activated links in each local set suffer negligible cumulative interference from all other activated link sets, *i.e.*, for each activated link l in local link set Z_i,

$$I_{out}^l \leq \varepsilon \cdot I_{\max},\ 0 < \varepsilon < 1,$$

where M is a constant, satisfying $M \geq \left[\frac{2\pi\eta P \cdot |\mathcal{X}_{ij}(t)|_{ub}}{(\kappa-2)\varepsilon I_{\max} R^\kappa}\right]^{\frac{1}{\kappa}}$.

Proof: The proof is available in Appendix C. ■

In light of Lemma 6, the partition strategy to enable distributed implementation will remain effective in the uniform power setting. Similarly, we have

Lemma 7: The weight of the newly computed link scheduling set inside each sub-square(i, j), *i.e.*, $W(\mathcal{X}_{ij}(t))$, has an approximation ratio of $\frac{4\mu}{(1-\varepsilon)^2}$ to the weight of the intersection set by the corresponding local link set L_{ij} and the global optimal MWISL $S^*(t)$.

Proof: The newly computed local scheduling set for each sub-square should be a $(1-\varepsilon)$-signal set, so its weight is at least $\frac{(1-\varepsilon)^2}{4}$ times the corresponding 1-signal set. Since the solution for MWISL with uniform power assignment proposed in [19] finds an ISL with affectness no greater than 1, it is obvious that

$$W(S_{ij}^*(t)) \leq W(OPT_{ij}(t)) \leq \frac{4\mu}{(1-\varepsilon)^2} W(\mathcal{X}_{ij}(t)), \tag{12}$$

where $\mu = \log |V|$ is the approximation bound of the algorithm in [19]. ■

Then, we will state an exact bound on the throughput performance of our algorithm in Theorem 2.

Theorem 2: The union of the local computed scheduling link sets, *i.e.*, $S(t) = \bigcup S_{ij}(t)$, is feasible under the uniform power setting. The weight of $S(t)$ achieves a fraction of $O(\log |V|)$ times the optimal solution.

Proof: We first show that the resulted scheduling link set by our algorithm is an independent link set with uniform power at every time slot. Using the same technique in the proof of Theorem 1, we can inductively conclude that each global scheduling set $S(t)$ is composed by disjoint link sets which are kept at least M cells away from each other. Therefore, by Lemma 6, we can get that for each link $l \in S(t)$, the total interference it receives satisfies that,

$$I_{S(t)}^l \leq (1-\varepsilon) \cdot I_{\max}^l + \varepsilon \cdot I_{\max} \leq I_{\max}^l, \tag{13}$$

indicating $S(t)$ is an independent link set. Using Lemma 6 and the same techniques in proof of Theorem 1, we get

$$\mathbb{P}\big(W(S(t)) \geq \frac{(1-\varepsilon)^2(K-2M)}{4K\mu} S^*(t)\big) \geq \frac{1}{K^2}, \tag{14}$$

where $\mu = \log |V|$. ■

The time complexity and communication complexity is the same as that of the oblivious power assignment.

V. PERFORMANCE EVALUATION

We do simulation experiments to evaluate the throughput performance of our proposed algorithms. The general setting of our experiments is as follows. We consider a network with 500 nodes, half of which as senders randomly located on a plane with size 200×200 units, the other half as receivers positioned uniformly at random inside disks of radius R around each of the senders. Packets arrive at each link independently in a Poisson process [1] with the same average arrival rate λ. Initially, we assign each link k packets where k is randomly chosen from $[100, 300]$. The path loss exponent is set to be 3.

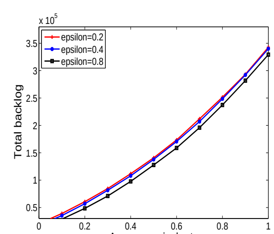
We conduct two series of experiments to focus on the evaluation of average throughput performance in terms of total backlog (the total number of unscheduled packets). In the first series, we study how some related variables affect the performance of the algorithms. In the second series, we further study the performance efficiency of the proposed algorithms by comparisons with two distributed algorithms:

1) Distributed Greedy Maximal Schedule (DGMS) [21], it needs to pre-computation network wide to determine a neighborhood of each link. If a link has maximum length in its neighborhood, it will get scheduled.
2) Distributed Randomized Algorithms (DRA), where each link determines to be active with a probability. To ensure that the global scheduling set is feasible or almost feasible, the probability has to be set quite small.

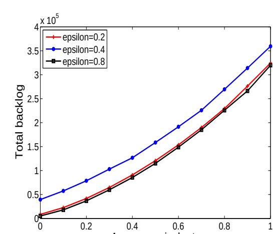
[1]The results hold under any arrival process satisfying the strong law of large numbers, using the fluid model approach [28] [10].

A. Under oblivious power Setting

We set the first series of experiments to study the performance of Algorithm 1 alone under different values of variables that may impact the average throughput performance actually. The variables $\frac{K}{M}$ and ε dominate the theoretical bound of Algorithm 1. When ε is fixed, a larger $\frac{K}{M}$ implies a bigger fraction of the optimal capacity region, but with a smaller probability of $\frac{1}{K^2}$ to achieve it. ε denotes a weighting factor between the interference a activated link suffers inside and outside the sub-square. A bigger value of ε indicates a smaller value of M theoretically. Therefore, though under the fixed value of $\frac{K}{M}$ a smaller ε leads to a bigger fractional capacity region, the probability to achieve this region gets smaller because of the resulted bigger M. Since the running time of the simulation is much shorter than the time for the algorithm to achieve the theoretical value, the probability will impact the actual average throughput in our experiments. Typically, a small probability of $\frac{1}{K^2}$ may cause poorer throughput performance. An experiment study of the two variables are illustrated in Fig. 3 and Fig. 4. In Fig. 3 we compare the total backlog by increasing the feasible values of $\frac{K}{M}$ where ε serves as a constant. Fig. 3(a), 3(b), 3(c) respectively shows the comparisons of total backlog at time slot 1000 with increasing arrival rate when $\varepsilon = 0.2$, $\varepsilon = 0.4$, $\varepsilon = 0.8$. Note that the value of $\frac{K}{M}$ must be greater than 2, or the size of the sub-squares will be 0. It shall also be noticed that the feasible values of $\frac{K}{M}$ are different when ε varies, since ε affects the value of M. It then explains why we set different values for $\frac{K}{M}$ for varying ε. All the three figures show that the throughput performance generally improves as $\frac{K}{M}$ increases, which coincides with the theoretical results we derive. Though the relevant probability shall become smaller as $\frac{K}{M}$ increases, there is no obvious sign shown in Fig. 3. (a), 3. (b). We can see an obvious impact in Fig. 3(c) where $\frac{K}{M}$ can be set bigger values. It shows the average throughput becomes a little worse at $\frac{K}{M} = 7$ than $\frac{K}{M} = 6$.



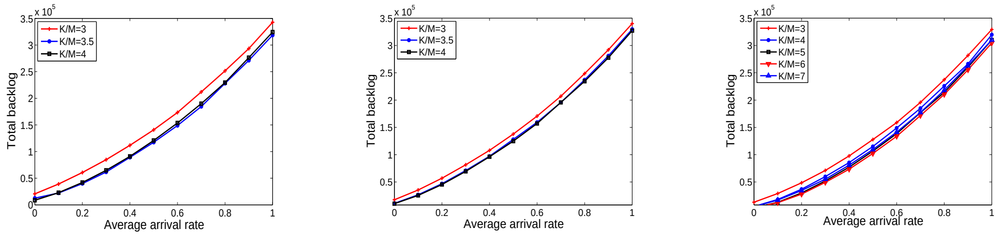
(a) $\frac{K}{M} = 3$



(b) $\frac{K}{M} = 4$

Fig. 4: Total backlog vs. average arrival rate vs. different values of ε at time slot 1000 in the oblivious power setting

We briefly illustrate the impact of ε at fixed $\frac{K}{M}$ in Fig. 4. Though the theoretically achievable capacity region shall have

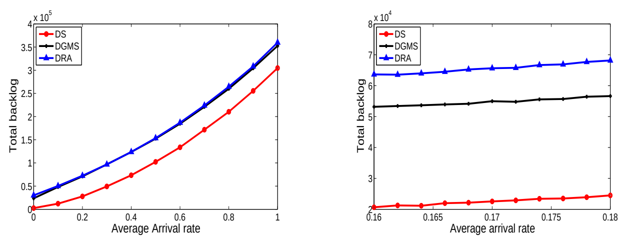


(a) $\varepsilon = 0.2$ (b) $\varepsilon = 0.4$ (c) $\varepsilon = 0.8$

Fig. 3: Total backlog vs. average arrival rate vs. different values of $\frac{K}{M}$ at time slot 1000 in the oblivious power setting

been greater with a smaller ε, it seems to show contradicted results in Fig. 4(a). The apparent contradiction lies behind the probability of $\frac{1}{K^2}$ which becomes quite small because of a much bigger M caused by a smaller ε. Fig. 4(b) shows a similar consequence caused by the crucial impact of ε both on the achievable capacity region and the relevant probability.

We then focus on comparisons with DGMS and DRA in Fig. 5, 6. We set $\varepsilon = 0.8$, $\frac{K}{M} = 6$ to conduct the following simulations. These simulation results indicate that our distributed scheduling algorithm (denoted by DS in figures) achieves much better performance than DGMS and DRA.



(a)Total backlog vs. average arrival rate (b) Zoom in of (a)

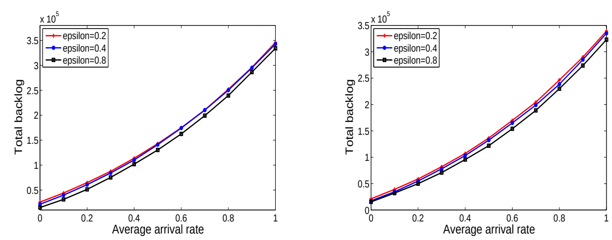
Fig. 5: Total backlog vs. average arrival rate vs. different algorithms at time slot 1000 in the oblivious power setting

In Fig. 5 we compare the total backlog changes of the three algorithms as average arrival rate increases. It shows that our algorithm may approximately support a maximum average rate no greater than 0.2, and the other two support no greater than 0.1. We zoom in a subgraph of the Fig. 5(a) in the Fig. 5(b), where the average arrival rate is in $[0.16, 0.18]$. It shows that our algorithm has a total queue length much smaller than the two. Fig. 6 then illustrates detailed comparisons of achievable capacity region for the three algorithms. It shows that our algorithm can support a larger traffic arrival rate vector. From the three subfigures we can see that our proposed algorithm can keep the total backlog stable at an arrival rate no greater than 0.18, while the counterpart of the distributed greedy algorithm and random algorithm is 0.07 and 0.05.

B. Under Uniform Power Setting

In the first series of experiment, we show the effect of $\frac{K}{M}$ at different values of ε where $\varepsilon = 0.2$, 0.4, 0.8 respectively in Fig. 7. Fig. 7(a), 7(b) and 7(c) plot the total backlog changes as increasing average arrival rate under different values of $\frac{K}{M}$. The trends are a little different from those in the oblivious power setting. From the three figures we can see that the average throughput performance gets better as increasing $\frac{K}{M}$ under a fixed ε. The decreasing probability of $\frac{1}{K^2}$ shows no obvious influence on the results. It may be partly caused by the algorithm for computing new schedulings inside sub-squares since it selects candidate links based on their distance with previous selected links. Thus a larger area implies more links get scheduled. This improvement remits the influence of the decreasing probability of $\frac{1}{K^2}$.

The similar trend occurs when ε increases at different values of $\frac{K}{M}$. We give a brief illustration in Fig. 8. Both the figures show that a larger ε generates better performance at fixed $\frac{K}{M}$. Since the affectness of local ISLs computed by the algorithm of [19] inside each sub-square may be much smaller than $1-\varepsilon$, it explains why ε has little influence on the theoretical bound actually. But ε has much more influence on the value of M and the corresponding probability. Therefore, in the uniform power setting, the algorithm achieves better performance with a larger ε.



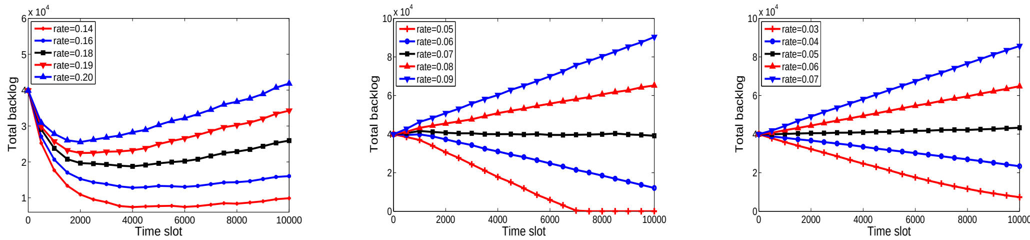
(a) $\frac{K}{M} = 3$ (b) $\frac{K}{M} = 4$

Fig. 8: Total backlog vs. average arrival rate vs. different values of ε at time slot 1000 in the uniform power setting

We conduct the second set of experiments to compare with DGMS and DRA on average throughput where we set $\varepsilon = 0.9$, $\frac{K}{M} = 9$. The results in Fig. 9, 10 convey the same kind of information as those in Fig. 5, 6. These graphs jointly show that Algorithm 1 still outperforms the DGMS and DRA in terms of total backlog. For example, the maximum supportable average arrival rate of our algorithm is around 0.12, comparing with 0.07 by DGMS and 0.04 by DRA.

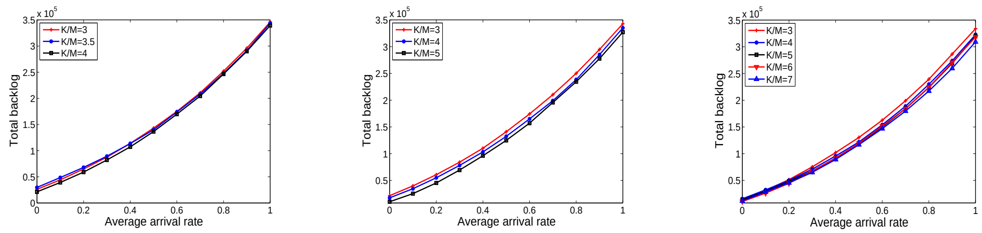
VI. Related Work

Numerous literatures consider different optimization measures and assume different interference models. Here we focus on related works on physical interference model. A complete review is available in our technique report [29].



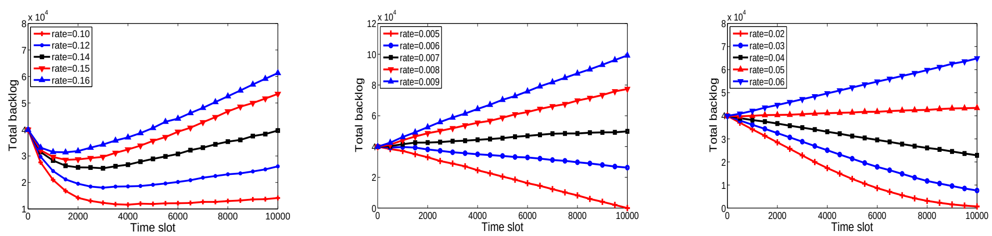
(a) DS (b) DGMS (c) DRA

Fig. 6: Achievable capacity region of different algorithms in the oblivious power setting



(a) $\varepsilon = 0.2$ (b) $\varepsilon = 0.4$ (c) $\varepsilon = 0.8$

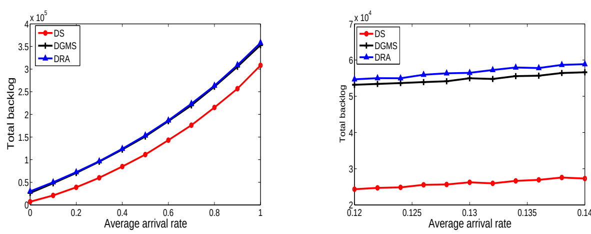
Fig. 7: Total backlog vs. average arrival rate vs. different values of $\frac{K}{M}$ at time slot 1000 in the uniform power setting



(a) DS (b) DGMS (c) DRA

Fig. 10: Achievable capacity region of different algorithms in the uniform power setting

1



(a) Total backlog vs. average arrival rate (b) Zoom in of (a)

Fig. 9: Total backlog vs. average arrival rate vs. different algorithms at time slot 1000 in the uniform power setting

Goussevskaia *et al.* [12] firstly present the NP-completeness proofs for the scheduling problem under the physical interference model. It firstly proposes algorithms with logarithmic approximation ratio $O(g(|E|))$ under the uniform power setting without noise, where $O(g(|E|))$ presents the link diversity among all links. Goussevskaia *et al.* further [13] attempt to develop a constant approximation-ratio algorithm for the problem of maximum independent set of links (MISL), a special case of MWISL with uniform weight. However, it is valid only without noise as pointed in [30]. Wan *et al.* finally succeed in developing a constant approximation-ratio algorithm for MISL with the existence of noise in [27].

Under the oblivious power setting, by utilizing partition and shifting strategies, Xu *et al.* get a constant approximation algorithm for MWISL subject to physical interferences [15]. Very recently Xu *et al.* [19] propose another constant approximation algorithm for the same problem based on the solution for the maximum weighted independent set of disks problem in [31]. They also develop a logarithmic approximation algorithm for the uniform power setting. Chafekar *et al.* [17] provide algorithms for maximizing throughput with logarithmic approximation ratio in the two power settings as well. However, the attained bound is not relative to the original optimal throughput capacity, but to the optimal value by using slightly smaller power levels.

Despite the main concern of this paper is on link scheduling for throughput maximization, we also make a review on the closely related problem of minimum length scheduling, which seeks a link schedule of minimum length that satisfies all link demands. A very recent work in [11] gives an overall analysis on link scheduling problems under the physical interference model from an algorithm view. It reveals that the algorithmic reduction from the minimum length scheduling to the throughput maximization scheduling is approximation-preserving.

The NP-completeness proofs of the minimum length scheduling problem, and a logarithmic approximation algorithm without noise in the uniform power setting, is available in [12]. With noise taken into consideration, Moscibroda *et al.* present a scheduling algorithm for the problem with power control, but without provable guarantee in [32]. They subsequently get a linear approximation bound in [33]. An attempt on a constant approximation bound with uniform power setting fails in [16], pointed out by Wan. *et al.* [18]. They [18] then propose a logarithmic approximation algorithm for the minimum length scheduling problem with power control.

Blough *et al.* [20] claim that the so-called black links, with length exactly at the maximum transmission range of the sender, hinder a tighter approximation bounds for the minimum length scheduling problem. They try to get a constant approximation ratio by limiting scheduling of such kind of links. Thus they revise the algorithm GOW proposed in [12] by partitioning links according to the SNR diversity, instead of the length diversity. However, their revised algorithm GOW* can only guarantee a constant approximation ratio when the number of black links is bounded by a constant. They have also recognized the necessity of distributed implementation for the scheduling algorithm. Some discussions on the suitability of the algorithm for distributed execution are then presented.

VII. Conclusion

We tackle the problem of throughput-optimum localized link scheduling subject to physical interference constraints. Our work provides theoretical guarantee for this practical link scheduling problem for multihop wireless networks, keeping them away from arbitrarily bad throughput performance. We believe that our work can find applications in some time-slotted wireless networks, *e.g.*, time-slotted wireless sensor networks or wireless mesh networks. Understanding these factors that determine the throughput of a network also helps to better deploy a multihop wireless network and enhance the overall throughput performance.

References

[1] Y. Q. Zhou, X.-Y. Li, M. Liu, Z. C. Li, S. J. Tang, X. F. Mao, and Q. Y. Huang, "Distributed link scheduling for throughput maximization under physical interference model," in *Proc. IEEE Infocom*, 2012, pp. 2691–2695.

[2] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, p. 10, 2009.

[3] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1132–1144, 2010.

[4] X.-H. Xu, X.-Y. Li, P.-J. Wan, and S.-J. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," *IEEE/ACM Transactions on networking*, Aug. 2011.

[5] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE Infocom*, 2008, pp. 1103–1111.

[6] L. Tassiulas and A. Ephremides, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proc. IEEE Infocom*, 1998, pp. 533–539.

[7] X. Lin and S. B. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proc. IEEE CDC*, 2006, pp. 1258–1263.

[8] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proc. ACM SIGMETRICS*, 2007, pp. 313–324.

[9] S.-J. Tang, X.-Y. Li, X. Wu, Y. Wu, X. Mao, P. Xu, and G. Chen, "Low complexity stable link scheduling for maximizing throughput in wireless networks," in *Proc. IEEE SECON*, 2009, pp. 1–9.

[10] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. ACM SIGMETRICS*, 2006, pp. 27–38.

[11] P.-J. Wan, O. Frieder, X.-H. Jia, F. Yao, X.-H. Xu, and S.-J. Tang, "Wireless link scheduling under physical interference model," in *Proc. IEEE Infocom*, 2011, pp. 838–845.

[12] O. Goussevskaia, Y. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *Proc. ACM Mobihoc*, 2007, pp. 100–109.

[13] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, and E. Welzl, "Capacity of arbitrary wireless networks," in *Proc. IEEE Inforcom*, 2009, pp. 1872–1880.

[14] X.-Y. Li, "Multicast capacity of wireless ad hoc networks," *IEEE/ACM Transactions on networking*, vol. 17, pp. 950–961, 2009.

[15] X.-H. Xu, S.-J. Tang, and P.-J. Wan, "Maximum weighted independent set of links under physical interference model," in *LNCS*, vol. 6221, 2010, pp. 68–74.

[16] M. Halldorsson and R. Wattenhofer, "Wireless communication is in APX," in *Proc. 36th International Colloquium on Automata, Languages and Programming*, 2009, pp. 525–536.

[17] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, "Arrpoximation algorithms for computing capacity of wireless networks with SINR constraints," in *Proc. IEEE Infocom*, 2008, pp. 1166–1174.

[18] P.-J. Wan, X.-H. Xu, and O. Frieder, "Shortest link scheduling with power control under physical interference model," in *Proc. IEEE MSN*, 2010, pp. 74–78.

[19] X. H. Xu, S. J. Tang, and X.-Y. Li, *Stable Wireless Link Scheduling Subject to Physical Interferences With Power Control*, 2011, manuscript.

[20] D. M. blough, G. Resta, and P. Santi, "Approximation algorithms for wireless link scheduling with SINR-Based inteference," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1701–1712, 2010.

[21] L.-B. Le, E. Modiano, C. Joo, and N. B. Shroff, "Longest-queue-first scheduling under SINR interference model," in *Proc. ACM Mobihoc*, 2010, pp. 41–50.

[22] G. Brar, D. Blough, and P. Santi, "The SCREAM approach for efficient distributed scheduling with physical interference in wireless mesh networks," in *Proc. IEEE ICDCS*, 2008, pp. 214 –224.

[23] D. Peleg, *Distributed computing: a locality-sensitive approach*, ser. SIAM Monographs on Discrete Mathematics and Applications. SIAM, 1987, vol. 5.

[24] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE/ACM Transactions on Automatic Control*, vol. 37, pp. 1936–1948, 1992.

[25] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MobiCom*, 2006, pp. 227–238.

[26] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proc. IEEE Infocom*, 2005, pp. 1804–1814.

[27] P.-J. Wan, X.-H. Jia, and F. Yao, "Maximum independent set of links under physical interference model," in *WASA*, 2009, pp. 169–178.

[28] S. Shakkottai and R. Srikant, "Network optimization and control," *Foundations and Trends in Networking*, vol. 2, no. 3, pp. 271–379, 2007.

[29] Y. Q. Zhou, X.-Y. Li, M. Liu, X. F. Mao, S. J. Tang, and Z. C. Li, "Throughput Optimizing Localized Link Scheduling for Multihop Wireless Networks Under Physical Interference Model," *ArXiv e-prints, http://arxiv.org/abs/1301.4738*, Jan. 2013.

[30] X.-H. Xu and S.-J. Tang, "A constant approximation algorithm for link scheduling in arbitrary networks under physical interference model," in *Proc. ACM FOWANC*, 2009, pp. 13–20.

[31] X.-Y. Li and Y. Wang, "Simple approximation algorithms and PTASs for various problems in wireless ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 66, pp. 515–530, 2006.

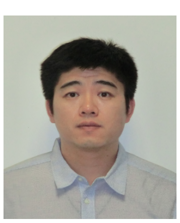
[32] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *Proc. IEEE Infocom*, 2006, pp. 1–13.

[33] ——, "Topology control meets SINR: the scheduling complexity of arbitrary topologies," in *Proc. IEEE Mobihoc*, 2006, pp. 310–321.

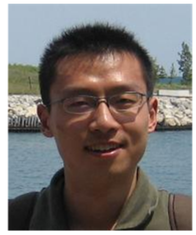
Yaqin Zhou is currently working toward the Ph.D. degree at Institute of Computing Technology, Chinese Academy of Sciences. She received her B.S. and M.S. degree in computer science from China Agricultural University, P.R. China, respectively in 2008 and 2010. Her research interests include link scheduling and channel accessing in wireless networks.

Xiang-yang Li has been a Professor (since 2012), Associate Professor (from 2006 to 2012) and Assistant Professor (from 2000 to 2006) of Computer Science at the Illinois Institute of Technology. He is recipient of China NSF Outstanding Overseas Young Researcher (B). Prof. Li received M.S. (2000) and Ph.D. (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received a Bachelor degree at Department of Computer Science and a Bachelor degree at Department of Business Management from Tsinghua University, P.R. China, both in 1995. He published a monograph ”Wireless Ad Hoc and Sensor Networks: Theory and Applications”. He also co-edited the book ”Encyclopedia of Algorithms”. The research of Prof. Li has been supported by USA NSF, HongKong RGC, and China NSF. His research interests span sensor networks, mobile computing, social networks, cryptography and network security, privacy, and computational geometry. Prof. Li is an editor of several journals, including IEEE Transaction on Parallel and Distributed Systems (2009 to present), and IEEE Transactions on Mobile Computing. He also served as TPC chair of several conferences, including, IEEE MASS 2013, ACM Mobihoc 2014. He is a senior member of the IEEE.

Min Liu received her B.S. and M.S. degrees in computer science from Xian Jiaotong University, China, in 1999 and 2002, respectively. She got her Ph.D in computer science from the Graduate University of the Chinese Academy of Sciences in 2008. She is currently an professor at the Networking Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include Mobility Management, Wireless Resource Management and Delay/Disruptive-Tolerant Networks.



Xufei Mao (M’10) received the Ph.D. degree in Computer Science from Illinois Institute of Technology, Chicago in 2010. He received the MS degree (2003) in Computer Science and the Bachelor degree (1999) in Computer Science at Northeastern University and Shenyang University of Technology respectively. He is with the School of Software and TNLIST, Tsinghua University, Beijing China. His research interests span wireless ad-hoc networks, wireless sensor networks, pervasive computing, mobile cloud computing and game theory.



Shaojie Tang is an Assistant Professor in the Department of Computer and Information Science (Research) at Temple University. He received his Ph.D degree from Department of Computer Science at Illinois Institute of Technology in 2012. He received B.S. in Radio Engineering from Southeast University, P.R. China in 2006. He is a member of IEEE. His main research interests focus on wireless networks (including sensor networks and cognitive radio networks), social networks, pervasive computing, mobile cloud computing and algorithm analysis and design. He has recently served as guest editor of Journal of Tsinghua Science and Technology. He also served as TPC member of a number of conferences such as IEEE ICPP, IEEE IPCCC, MSN.

Zhongcheng Li received his B.S. degree in computer science from Peking University in 1983, and M.S. and Ph.D. degrees from Institute of Computing Technology, Chinese Academy of Sciences in 1986 and 1991, respectively. From 1996 to 1997, he was a visiting professor at University of California at Berkeley. He is currently a professor of Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include next generation Internet, dependable systems and networks, and wireless communication.