Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks

Yu Wang, *Member*, *IEEE*, Weizhao Wang, Xiang-Yang Li, *Member*, *IEEE*, and Wen-Zhan Song, *Member*, *IEEE*

Abstract—We study efficient interference-aware joint routing and TDMA link scheduling for a multihop wireless network to maximize its throughput. Efficient link scheduling can greatly reduce the interference effect of close-by transmissions. Unlike the previous studies that often assume a unit disk graph (UDG) model, we assume that different terminals could have different transmission ranges and different interference ranges. In our model, it is also possible that a *communication link* may *not* exist due to barriers or is not used by a predetermined routing protocol, while the transmission of a node always result interference to *all* nonintended receivers within its interference range. Using a mathematical formulation, we develop interference-aware joint routing and synchronized TDMA link schedulings that optimize the networking throughput subject to various constraints. Our linear programming formulation will find a flow routing whose achieved throughput is at least a constant fraction of the optimum, and the achieved fairness is also a constant fraction of the requirement. Then, by assuming known link capacities and link traffic loads, we study link scheduling under the request-to-send and clear-to-send (RTS/CTS) interference model and the protocol interference model (PrIM) with fixed transmission power. For both models, we present both efficient centralized and distributed algorithms that use timeslots within a constant factor of the optimum. We also present efficient distributed algorithms whose performances are still comparable with optimum, but with much less communications. We prove that the timeslots needed by our faster distributed algorithms are only at most $O(\min(\log n, \log \psi))$ for RTS/ CTS interference model and PrIM. Our theoretical results are corroborated by extensive simulation studies.

Index Terms—Link scheduling, interference, graph coloring, distributed algorithm, wireless networks.

1 Introduction

Wireless multihop radio networks such as ad hoc, mesh, or sensor networks are formed of autonomous nodes communicating via radio. Wireless networks draw lots of attentions in recent years due to their potential applications in various areas. For example, wireless mesh networks are being used as the last mile for extending the Internet connectivity for mobile nodes. These networks behave almost like wired networks since they have infrequent topology changes, limited node failures, etc. For wireless mesh networks or sensor networks, the aggregate traffic load of each routing node changes infrequently also. A unique characteristic of wireless networks is that the radio sent out by a wireless terminal will be received by all the terminals within its transmission range, and also possibly causes signal interference to some terminals that are not intended receivers. In other words, the communication channels are shared by the wireless terminals. Thus, one of the major problems facing wireless networks is the reduction of capacity due to interference caused by simultaneous transmissions. Using multiple channels and multiple radios can alleviate but not eliminate the interference. To achieve robust and collision-free communication, there are two alternatives. One is to utilize a random access MAC layer scheme. The other is to carefully construct a transmission schedule. One variant, link scheduling in the context of time division multiplexing (TDM) is the subject of this paper.

In this paper, we assume that the time is slotted and synchronized. A link scheduling is to assign each link a set of timeslots $\subset [1, T]$ on which it will transmit, where T is the scheduling period. A link scheduling is *interference-aware* (or called *valid*) if a scheduled transmission on a link $x \rightarrow y$ will not result in a collision at either node x or node y (or any other node). In this context, two types of collisions must be avoided, namely, primary interference and secondary interference. Link scheduling has received a great attention from both networking and theory fields [1], [2], [3], [4], [5], [6], [7], [8], [9] in the past few years due to its application for assigning timeslots in TDMA MAC protocols that eliminate collision and guarantee fairness. Many scheduling problems in wireless networks have been shown to be NP-complete, including TDMA broadcast scheduling [10] and link scheduling [11], [12]. For some of these problems, even polynomial-time algorithms with constant approximation ratios appear unlikely for general graphs.

Previous studies on link scheduling either assume a very general graph model or assume a very specific graph model such as unit disk graph (UDG). It is widely accepted in the

- *Y. Wang is with the Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223. E-mail: yu.wang@uncc.edu.*
- *W. Wang is with Google Inc., 19540 Jamboree Rd., Irvine, CA 92612. E-mail: weizhao@google.com.*
- *X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, 10 W. 31st Street, Chicago, IL 60616, and also with Microsoft Research Asia, Beijing 100080, P.R. China. E-mail: xli@cs.iit.edu.*
- *W.-Z. Song is with the School of Engineering and Computer Science, Washington State University, Vancouver, WA 98686. E-mail: songwz@wsu.edu, song@vancouver.wsu.edu.*

Manuscript received 4 Apr. 2007; revised 5 Oct. 2007; accepted 3 Jan. 2008; published online 21 Mar. 2008.
Recommended for acceptance by X. Zhang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-04-0105.
Digital Object Identifier no. 10.1109/TPDS.2008.53.

1045-9219/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

wireless networking community that neither a general graph model nor a UDG model accurately captures unique properties of wireless networks. A general graph model could not capture a certain geometry property of wireless networks, e.g., two nodes must be within certain distance to be able to communicate directly (or one node's transmission could interfere the other node's reception). A UDG model is idealistic since in practice two nearby nodes may still be unable to communicate due to various reasons such as barrier and path fading. In this paper, we give efficient centralized and distributed algorithms to obtain a valid link scheduling with theoretically proven performances for a more realistic wireless network model.

For wireless networks, another challenging issue is to route the flow cooperatively among all flows to maximize the network throughput. For example of sensor networks, if routing scheme is not designed carefully, nodes near the sink node will get a large share of the network bandwidth than the nodes that are far away from the sink nodes. Thus, given demands of nodes, we need to jointly optimize the routing and TDMA link scheduling to maximize the throughput.

The main contributions of this paper are as follows:

Theoretical performance guarantee for efficient algorithms. We first consider the joint routing and scheduling problem to maximize either the min-fairness or maximize the network throughput under a given min-fairness requirement $\lambda_0 \geq 0$. We present a linear programming formulation based on both necessary and sufficient conditions for schedulable flows under various interference models. Based on this, we design a joint routing and TDMA link-scheduling algorithm that will achieve a network throughput within a constant factor of the optimum. Here, we consider two interference models: request-to-send and clear-to-send (RTS/CTS) model and fixed power protocol interference model (fPrIM). After flow routing is computed, we then present both centralized and distributed link-scheduling algorithms that use timeslots at most a constant factor of the optimum. All algorithms involve a novel study of interference properties in wireless networks. One of our distributed algorithms has not only small communication complexity but also good performance guarantee that is only logarithmic of the ratio between the maximum and minimum interference range. Specifically, we prove that the timeslots needed by our faster distributed algorithms are only at most $O(\min(\log n, \log \psi))$ for RTS/CTS model and fPrIM model, where ψ is the ratio between the largest and smallest interference ranges among all n nodes. Although some of our algorithms are similar to some algorithms proposed before, to the best of our knowledge, we are the first one to prove asymptotic optimal bounds for the performance.

More realistic model. We address the link scheduling in a more realistic networking model: 1) each node has its own transmission power and, thus, its own transmission range; 2) that the receiver must be within the transmission range of the sender is only a necessary (but not sufficient) condition for two nodes to communicate directly, i.e., two nearby nodes may still be unable to communicate directly; and 3) if a node v is within certain distance of a sender u, then the transmission by u will interfere the reception of node v. In summary, the communication graph could be an arbitrary geometry graph. Notice that similar realistic models using weighted and unweighted flows, modeling interference range to be different from transmission range, etc., have all been proposed and modeled in earlier work, e.g., in [4], [7], and [13], and heuristic algorithms have been given for each or all of these. Our contributions here are that we provide theoretical bounds for link-scheduling algorithms in these cases.

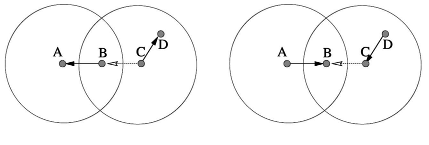
Both weighted and unweighted flow. In several wireless networks (e.g., mesh and sensor networks), we can estimate the traffic demand by each wireless node. Thus, based on a given routing algorithm, we can predict the average traffic load $f(e)$ on each link e of the network. We then design link-scheduling algorithms to meet this traffic demand if possible. We model this by assuming that each link e has an integral *weight* $w(e)$ specifying the number of slots it needed in a period to support its traffic load. Here, $w(e) = \lceil T \cdot \frac{f(e)}{\mathbf{c}(e)} \rceil$, where $\mathbf{c}(e)$ is the capacity of link e if there is no interference, and T is a given period for a schedule. In certain networks, it is difficult, if not impossible, to estimate the load of every link. We then assume that each node needs one timeslot for transmission, and our objective is to design a scheduling that minimizes T.

The rest of the paper is organized as follows: Section 2 discusses our network and interference models and formally defines the problem studied in this paper. A mixed integer programming formulation of proposed problems is presented in Section 3. Our centralized and distributed algorithms for link scheduling are given in Sections 4 and 5, respectively. We also analyze the theoretical guaranteed performances of our algorithms. In Section 6, we study how to assign timeslots to links when each link has a requirement of the least number of timeslots needed. Our simulation studies are reported in Section 7. In Section 8, we briefly review the related works in the literature. We conclude our paper in Section 9 with the discussion of some possible future works. A preliminary conference version of this article appeared in [14]. Due to space limit, some detailed proofs are omitted in this version with a simple reference to [14].

2 System Model and Assumptions

2.1 Network and Interference Models

NETWORK MODEL: We assume that there is a set V of communication terminals deployed in a plane. Each wireless terminal is only equipped with *single* radio interface. The complete communication graph is a *directed* graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is the set of terminals and E is the set of possible directed communication links. Every terminal v_i has a transmission range t_i such that the necessary condition for a terminal v_j to receive correctly the signal from v_i is $\|v_i - v_j\| \leq t_i$, where $\|v_i - v_j\|$ (sometimes we denote it as $d_{i,j}$ for simplicity) is the euclidean distance between v_i and v_j. Notice that $\|v_i - v_j\| \leq t_i$ is not the sufficient condition for $(v_i, v_j) \in E$. Some links do not belong to G because of either the physical barriers or the selection of routing protocols. This is the major distinction of our model with the majority previous studies on link scheduling. To the best of our knowledge, only [7] used the similar model



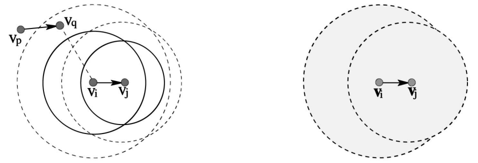
(a) (b)

Fig. 1. Communication restriction by RTS/CTS. (a) Due to RTS. (b) Due to CTS.

as ours. We always use $\mathbf{L}_{i,j}$ to denote (v_i, v_j) hereafter. For a link e, we use $\mathbf{c}(e)$ to denote its expected capacity when no interference links are transmitting simultaneously. Each terminal v_i also has an interference range r_i such that v_j is interfered by the signal from v_i if $\|v_i - v_j\| \leq r_i$ and v_j is not the intended receiver. The interference range r_i is not necessarily the same as the transmission range t_i. Typically, $r_i > t_i$. We call the ratio between them as the *Interference-Transmission Ratio* for node v_i, denoted as $\gamma_i = \frac{r_i}{t_i}$. In practice, $2 \leq \gamma_i \leq 4$. For all wireless nodes, let $\gamma = \max_{v_i \in V} \frac{r_i}{t_i}$. For a node u, we use $\Lambda^+(u)$ to denote the set of incoming links (all directed links pointed to u). Similarly, we use $\Lambda^-(u)$ to denote the set of outgoing links at node u.

INTERFERENCE MODELS: To schedule two links at the same timeslot, we must ensure that the schedule will avoid the interference. Two different types of interference have been studied in the literature, namely, *primary interference* and *secondary interference*. Primary interference occurs when a node transmits and receives packets at the same time. Secondary interference occurs when a node receives two or more separate transmissions. Here, all transmissions could be intended for this node, or only one transmission is intended for this node (thus, all other transmissions are interference to this node). In addition to these interferences, there could have some other constraints on the scheduling, e.g., the radio networks that deploy the IEEE 802.11 protocol with RTS/CTS mechanism will pose some additional constraints. Several different interference models have been used to model the interferences in wireless networks. We briefly review the models we use in this paper.

Protocol interference model (PrIM) [15]. In this model, a transmission by a node v_i is successfully received by a node v_j if and only if the intended destination v_j is sufficiently apart from the source of any other simultaneous transmission, i.e., $\|v_k - v_j\| \geq (1+\eta)\|v_i - v_j\|$ for any node $v_k \neq v_i$. Here, constant $\eta > 0$ models situations where a guard zone is specified by the protocol to prevent a neighboring node from transmitting on the same channel at the same time. This model *implicitly* assumed that each node v_k will adopt the power control mechanism when it transmits signals. Simulation analysis [16] as well as the analytical results [17] indicates that the PrIM does not necessarily provide a comprehensive view of reality due to the aggregate effect of interference in wireless networks. However, it does provide some good estimations of interference, and most importantly, it enables a theoretical performance analysis of a number of protocols designed in the literature. Link scheduling under PrIM and network model similar to ours has been studied in [7].



(a) (b)

Fig. 2. RTS/CTS interference model. (a) $\mathbf{L}_{i,j}$ interferes $\mathbf{L}_{p,q}$. (b) Interference region $I_{i,j}$.

Fixed power protocol interference model (fPrIM). We adopt the following interference model throughout this paper. We assume that a node will *not* dynamically change its power based on the intended receiver in a packet level. Note that this assumption does not preclude the power control that can further reduce the power consumption. We only assume that there is no power adaptation at the packet level and the power is not adjustable for a certain period of time, which is close to the real situation. However, we do assume that each node v_i has its own fixed transmission power and, thus, a fixed transmission range t_i. We also assume that each node v_k has an *interference range* r_k such that any node v_j will be interfered by the signal from v_k if $\|v_k - v_j\| \leq r_k$ and node v_k is sending signal to some node other than v_j. In other words, the transmission from v_i to v_j is viewed successful if $\|v_k - v_j\| > r_k$ for every node v_k transmitting in the same timeslot using the same channel.

RTS/CST interference model. This model was also studied previously, e.g., [1]. For every pair of transmitter and receiver, all nodes that are within the interference range of either the transmitter or the receiver cannot transmit. Fig. 1a shows the case that communication from B to A and C to D cannot take place simultaneously due to RTS. Fig. 1b shows the case that communication from A to B and D to C cannot take place simultaneously due to CTS. Although RTS/CTS is not the interference itself, for convenience of our notation, we will treat the communication restriction due to RTS/CTS as *RTS/CTS interference* model. Thus, for every pair of simultaneous communication links, say v_iv_j and v_pv_q, it should satisfy that 1) they are distinct four nodes, i.e., $v_i \neq v_j \neq v_p \neq v_q$ and 2) v_i and v_j are not in the interference ranges of v_p and v_q, and vice versa. Fig. 2a shows an example where link $\mathbf{L}_{i,j}$ interferes $\mathbf{L}_{p,q}$. Here, a solid circle with center v denotes the transmission region and dotted circle denotes the interference region of node v. The *interference region*, denoted by $I_{i,j}$, of a link $\mathbf{L}_{i,j}$ is the union of the interference region of nodes v_i and v_j (see Fig. 2b for illustration). When a directed link v_iv_j (or v_jv_i) is active, all simultaneous transmitting links v_pv_q cannot have an endpoint inside the area $I_{i,j}$.

There are also other interference models, e.g., Transmitter Interference Model [18] and Physical Interference Model. However, in this paper, we mainly focus on joint routing and link scheduling for fPrIM and RTS/CTS models. Note that these two models are different, e.g., in Fig. 1a, links BA and CD can be assigned the same channel in fPrIM model, but not in RTS/CTS model. Similar statement holds for links AB and DC in Fig. 1b.

Assume that the communication links in the wireless network are predetermined. Given a communication graph $G = (V, E)$, we use the *conflict graph* (e.g., [13]) F_G to represent the interference in G. Each vertex (denoted by

$\mathbf{L}_{i,j}$) of F_G corresponds to a directed link (v_i, v_j) in the communication graph G. There is an *edge* between vertex $\mathbf{L}_{i,j}$ and vertex $\mathbf{L}_{p,q}$ in F_G if and only if $\mathbf{L}_{i,j}$ conflicts with $\mathbf{L}_{p,q}$ 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. To avoid the confusion, we use F_G^P to denote the interference graph under the fPrIM model and F_G^{D2} to denote interference graph under RTS/CTS model.

2.2 Problem Formulation

Assume that each ordinary node u will aggregate the traffic from all its users and then route them to the Internet through some gateway nodes. We use $\ell_O(u)$ to denote the total aggregated outgoing traffic of node u users and $\ell_I(u)$ to denote the total aggregated incoming traffic of node u users. We will mainly concentrate on incoming traffic in this paper. For notation simplicity, we use $\ell(u)$ to denote such load for node u. Notice that the traffic $\ell(u)$ is not requested to be routed through a specific gateway node, neither requested to be using a single routing path. We also assume that among the set V of all wireless nodes, some of them have gateway functionality and provides the connectivity to the Internet. For simplicity, let $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_g\}$ be the set of g gateway nodes, where $\mathbf{s}_i$ is actually node v_{n+i-g}. All other wireless nodes v_i (for $1 \leq i \leq n-g$) are called *ordinary* wireless nodes. We assume that the gateway nodes will *not* act as relay node for a pair of ordinary wireless nodes. The **routing problem** is to decide a multipath routing structure for each source node and an assignment of its flow to all links in the network. The flow assignment should satisfy certain restrictions such as flow conservation. Most importantly, the assigned flow should be schedulable by the coupled link-scheduling method.

After the flow is assigned to each link, we then need to decide when a node should be actively sending data to a neighboring node, when TDMA link scheduling is adopted. Our objective of the **scheduling problem** is to give each link $\mathbf{L} \in G$ a transmission schedule $\mathcal{S}(\mathbf{L})$, which is the list of timeslots it could send packets such that the schedule is interference free and the overall throughout of the network is maximized. Let $X_{e,t} \in \{0,1\}$ be the indicator variable which is 1 if and only if e will transmit at time t. We will focus on periodic schedules in this paper. A schedule is periodic with period T if, for every link e and timeslot t, $X_{e,t} = X_{e,t+i\cdot T}$ for any integer i. For a link e, let $I(e)$ denote the set of links e' that will cause interference if e and e' are scheduled at the same timeslot. A schedule $\mathcal{S}$ is *interference free* if $X_{e,t} + X_{e',t} \leq 1$ for any $e' \in I(e)$. In the graph theory terminology, the interference-free link-scheduling problem is essentially the weighted *vertex coloring* of F_G.

When the traffic load of links are unknown, the objective of link scheduling is to find a scheduling with the minimum period. If we schedule all links within a period χ such that no two links in same timeslot interfere with each other, then at least one packet can be delivered over each communication link in every χ timeslots. Thus, $1/\chi$ is often used to estimate the *throughput* of the network based on this schedule. The second case is that the average traffic load $f(e)$ of each link is known in advance from the routing. We model this by assuming that each communication link e (vertex in the conflict graph) has a *weight* $w(e)$ specifying the minimum number of timeslots it required in each period. Here, $w(e) = \lceil T \cdot \frac{f(e)}{\mathbf{c}(e)} \rceil$, where $\mathbf{c}(e)$ is the capacity of link e if there is no interference, and T is a given period for a schedule. Our main focus in this paper is how to schedule the communication links in an interference-free manner such that the throughput of the network is maximized, i.e., with the smallest T.

Notice that, for simplicity, we assume that there is only a single channel in the network. All our results can be easily extended to the case when multiple channels are available as in [1]. If nodes have preassigned channels for each link, then the link scheduling with multiple channels is just the simple union of a set of schedulings, where each scheduling is for all links using the same channel. However, we agree that the static assignment of correct channels to appropriate links is a bigger factor in determining the performance. If links can dynamically switch channels, then our greedy algorithms will find the channel with the smallest available timeslot for each link to be scheduled and the same performances hold.

3 Joint Routing and Link Scheduling

In this section, we first give a mixed Integer Programming formulation of the problem to be studied.

First, assume that each source node has a demand for data rate $\ell(u)$. We want to find a routing that maximizes the minimum fairness, which is defined as the ratio of the achieved data rate over the required data rate. Given a link e, let $f(e)$ be the total flow assigned to link e. We formulate the max-min-fairness routing problem as follows:

$$\max \lambda$$
$$\begin{cases} \sum_{e \in \Lambda^+(u)} f(e) - \sum_{e \in \Lambda^-(u)} f(e) & = & f(u) & \forall u \notin \mathcal{S}, \\ f(u) & \geq & \lambda \ell(u) & \forall u \notin \mathcal{S}, \\ \alpha(e) \cdot \mathbf{c}(e) & = & f(e) & \forall e, \\ \alpha(e) & \geq & 0 & \forall e, \\ \alpha(e) & \leq & 1 & \forall e, \\ \textit{exists} \text{ interference-free schedule for} & & f(e). & \end{cases}$$

Here, $f(u)$ is the achieved data rate for node u with flow assignment f; $0 \leq \alpha(e) \leq 1$ is the fraction of the time link e will be actively transmitting to achieve such flow assignment. Notice that, for links that interfere with each other, clearly, the summation of their $\alpha(e)$ should be no more than 1. It is widely known that it is NP-hard to decide whether a feasible scheduling $X_{e,t}$ exists when given the flow $f(e)$ (or equivalently, $\alpha(e)$) for wireless networks with interference constraints. Similarly, we can formulate the problem of routing for maximizing the throughput where the objective function is $\max_{u \in V} f(u)$ and the λ in the section inequality is replaced by some minimum fairness requirement constant $\lambda_0 \geq 0$.

Schedulable flows. We then mathematically formulate the necessary and sufficient condition for schedulable flow $f(e) = \alpha(e) \cdot \mathbf{c}(e)$: flow f (equivalently, whether a given vector $\alpha(e)$ for all e is schedulable) is schedulable if and only if we can find integer solution $X_{e,t}$ satisfying the following conditions:

$$\begin{cases} X_{e,t} + X_{e',t} & \leq & 1 & \forall e' \in \mathbf{I}(e), \forall e, \forall t, \\ \frac{\sum_{1 \leq t \leq T} X_{e,t}}{T} & = & \alpha(e) & \forall e, \\ X_{e,t} & \in & \{0, 1\} & \forall e, \forall t. \end{cases}$$

Recall that here $X_{e,t}$ denotes whether link e is active at time $t \in [1, T]$. For some interference models, several papers gave relaxed necessary conditions and relaxed sufficient conditions for schedulable flows that can be decided in polynomial time. For example, for RTS/CTS model with uniform transmission range and uniform interference range, [1] gave a sufficient condition $\alpha(e) + \sum_{e' \in \mathbf{I}(e)} \alpha(e') \leq 1$, and a necessary condition $\alpha(e) + \sum_{e' \in \mathbf{I}(e)} \alpha(e') \leq C(q)$. Here, $C(q)$ is a constant depending on the ratio of interference range over the transmission range.

For *each* of the interference models discussed in this paper, we will later present a necessary and a sufficient condition for schedulable flows. Generally, we have the following theorem (whose proof is deferred to later section):

Theorem 1. *Assume that the network is single-channel network. A sufficient condition for a flow defined by $\alpha(e)$ to be schedulable is*

$$\alpha(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha(e') \leq 1,$$

and a necessary condition for such flow to be schedulable is

$$\alpha(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha(e') \leq C_{\mathcal{M}}.$$

Here, $\mathbf{I}_{\mathcal{M}}(e) \subseteq \mathbf{I}(e)$ is defined based on the specific interference model $\mathcal{M}$ for the purpose of link scheduling; $C_{\mathcal{M}}$ is a constant depending on the specific interference model and γ. $C_{RTS/CTS}$ is a constant defined in Lemma 6; while $C_{fPrIM} = \lceil \frac{2\pi}{\arcsin \frac{\gamma - 1}{2\gamma}} \rceil$ is proved in Lemma 9.

Consequently, we need to solve the following Linear Programming (**LP-Flow-fairness**) for $\alpha(e)$ such that

$$\max \lambda$$
$$\begin{cases} \sum_{e \in \Lambda^+(u)} f(e) - \sum_{e \in \Lambda^-(u)} f(e) & = & f(u) & \forall u \notin \mathcal{S}, \\ f(u) & \geq & \lambda \ell(u) & \forall u \notin \mathcal{S}, \\ \alpha(e) \cdot \mathbf{c}(e) & = & f(e) & \forall e, \\ \alpha(e) & \geq & 0 & \forall e, \\ \alpha(e) & \leq & 1 & \forall e, \\ \alpha(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha(e') & \leq & 1 & \forall e. \end{cases}$$

In majority of applications, we not only have to guarantee certain fairness of the achieved flows for all end wireless devices but we also have to achieve the largest possible throughput under certain fairness constraints. Assume that we have a minimum fairness constraints λ_0. To *approximately* find the maximum throughput routing, we will solve the following linear programming (**LP-Flow-throughput**) for $\alpha(e)$ such that

$$\max \sum_{I=1}^{g} f(\mathbf{s}_i)$$
$$\begin{cases} \sum_{e \in \Lambda^+(u)} f(e) - \sum_{e \in \Lambda^-(u)} f(e) & = & f(u) & \forall u \notin \mathcal{S}, \\ f(u) & \geq & \lambda_0 \ell(u) & \forall u \notin \mathcal{S}, \\ \sum_{e \in \Lambda^-(\mathbf{s}_i)} f(e) - \sum_{e \in \Lambda^+(\mathbf{s}_i)} f(e) & = & f(\mathbf{s}_i) & \forall \mathbf{s}_i \in \mathcal{S}, \\ \alpha(e) \cdot \mathbf{c}(e) & = & f(e) & \forall e, \\ \alpha(e) & \geq & 0 & \forall e, \\ \alpha(e) & \leq & 1 & \forall e, \\ \alpha(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha(e') & \leq & 1 & \forall e. \end{cases}$$

Based on the above linear programming formulations, we will solve $\alpha(e)$ for all links e. In following sections, we will present both centralized algorithms (Algorithms 1 and 2 for link scheduling in RTS/CTS and fPrIM models, respectively) and distributed algorithms for scheduling link activities to achieve the flows. These efficient algorithms, together with our linear programming formulations, imply the following theorems:

Theorem 2. *Algorithms 1 and 2 together with Algorithm 6 and the linear programming formulation* ***LP-Flow-fairness*** *produce a feasible interference-free link-channel scheduling whose achieved fairness is at least $\frac{1}{C_{\mathcal{M}}}$ of the optimum.*

Proof. Consider an optimum flow assignment defined by $\alpha^*(e)$, i.e., the flow supported by a link e is $\alpha^*(e) \cdot \mathbf{c}(e)$. From Theorem 1, we know that

$$\alpha^*(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha^*(e) \leq C_{\mathcal{M}}.$$

Define a new flow α' as $\alpha'(e) = \frac{\alpha^*(e)}{C_{\mathcal{M}}}$. Obviously,

$$\alpha'(e) + \sum_{e' \in \mathbf{I}_{\mathcal{M}}(e)} \alpha'(e) \leq 1.$$

It is easy to show that the new flow α' satisfies all conditions of our linear programming **LP-Flow-fairness**. In other words, α' is a feasible solution for this LP. Consequently, the solution of **LP-Flow-fairness** is at least that of α', which is $\frac{1}{C_{\mathcal{M}}}$ of the optimum. This finishes the proof. □

Similarly, we have:

Theorem 3. *Algorithms 1 and 2 together with Algorithm 6 and the linear programming formulation* ***LP-Flow-throughput*** *produce a feasible interference-free link-channel scheduling whose achieved throughput is at least $\frac{1}{C_{\mathcal{M}}}$ of the optimum, whose achieved fairness is at least $\frac{1}{C_{\mathcal{M}}} \lambda_0$.*

4 Centralized Link Scheduling

In this section, we propose centralized link-scheduling algorithms under different interference models when the objective is to schedule every link once and minimize the time period T used. Some fundamental studies of interference graph here will form the bases for scheduling links when each link has a requirement on the number of timeslots it needed in a scheduling period.

4.1 Scheduling under RTS/CTS Model

A number of centralized algorithms for link scheduling have been proposed in the literature, e.g., [1] and [7]. A common approach is to assign each link the best possible channels (the smallest timeslots here) by greedy. The difference between them is the processing order of links: [7] processes links with smaller lengths first while [1] processes links in an arbitrary order (since it uses UDG models for both communication and interference). Our centralized algorithm (Algorithm 1) processes links in a special order as in [19]. The basic idea is to first sort links as follows: every time we pick a link, say **L**, from the remaining graph that has the smallest number of interfered links in the remaining graph and then remove **L** from this graph; repeat this till the graph becomes empty. We then assign timeslots to links in the reverse order of picked links using the smallest timeslot available (not used by interfering links). In summary, a link e with larger $I(e)$ will be more likely processed earlier.

```
Algorithm 1: Centralized scheduling under RTS/CTS
             model
Input: A communication graph $G = (V, E)$ of $m$ links.
Output: An interference-free link scheduling.
  1: Construct the conflict graph $F_G^{D2}$ and let graph
     $G' = F_G^{D2}$.
  2: while $G'$ is not empty do
  3:     Find the vertex with the smallest total degree in $G'$
         and remove this vertex from $G'$ and all its incident
         edges. Let $\mathbf{L}_k$ denote the $(m-k+1)$th vertex
         removed, and the degree of $\mathbf{L}_k$ in graph $G'$ just
         before it is removed be its δ-degree.
  4: Process links from $\mathbf{L}_1$ to $\mathbf{L}_m$ and assign to each $\mathbf{L}_k$ the
     smallest timeslot not yet assigned to any of its
     neighbors in $F_G^{D2}$.
```

We first present some necessary definitions and properties needed to prove the performance of our algorithms. Given a communication link $\mathbf{L}_{i,j}$, we define the *interference radius of link* $\mathbf{L}_{i,j}$ as $r_{i,j} = \max\{r_i, r_j\}$. If $r_i > r_j$ or $r_i = r_j$ and ID of node v_i is larger than the ID of node v_j, then v_i is called the *head* (denoted as $h_{i,j}$) of link (v_i, v_j) and v_j is the *tail* (denoted as $t_{i,j}$) of this link. Notice that, here, the *head* of a link is not necessarily the sender of the directed communication link. Given a node v_k, we use $R(v_k, x)$ to denote the disk centered at v_k and with radius $x \cdot r_k$. A node v_k interferes a node v_i if node v_i is inside the interference region (i.e., disk $R(v_k, 1)$) of node v_k. We say a link $\mathbf{L}_{p,q}$ interferes a node v_k if either v_p or v_q interferes v_k. For a given node v_k, we use $N^{\geq}(v_k, \alpha)$ to denote the set of nodes satisfying that 1) each of their interference radius is at least r_k and 2) each of them interferes some nodes in $R(v_k, \alpha)$. Notice that a node from $N^{\geq}(v_k, \alpha)$ could be arbitrarily far away from node v_k. Similarly, for a link $\mathbf{L}_{i,j}$, let $R(\mathbf{L}_{i,j}, x)$ denote the union of two disks centered at v_i and v_j, respectively, with radius $x \cdot r_i$ and $x \cdot r_j$, respectively. Let $N^{\geq}(\mathbf{L}_{i,j}, \alpha)$ denote the union of node sets $N^{\geq}(v_i, \alpha)$ and $N^{\geq}(v_j, \alpha)$. The following theorem estimates the local chromatic number based on node degree:

Theorem 4. *For a given node v_k and any node set $V_k \subseteq N^{\geq}(v_k, \alpha)$ with constant α, there exists a subset V'_k of V_k with cardinality $|V_k|/C_\alpha$ such that each node interferes with each other, where $C_\alpha \leq (6\alpha + 1)^2 + 11$.*

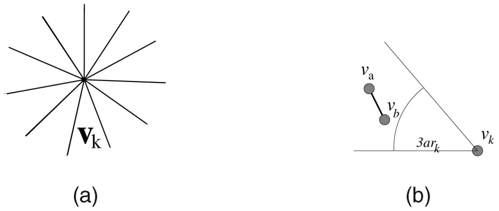


Fig. 3. Illustration of the partition of the region. (a) Divide into 11 cones. (b) Two nodes interfere in same cone.

Proof. We consider a partition of V_k: the nodes in and outside region $R(v_k, 3\alpha)$, denoted by V_k^1 and V_k^2, respectively.

First, we consider the node set V_k^1. Using a simple area argument, there are at most $\frac{\pi((3\alpha+\frac{1}{2})r_k)^2}{\pi(\frac{1}{2}r_k)^2} = (6\alpha + 1)^2$ disks with radius $\frac{r_k}{2}$ can be placed inside the disk $R(v_k, 3\alpha)$. Thus, there exists a node set in V_k^1 with a size of, at least, $|V_k^1|/(6\alpha + 1)^2$, such that each node in the set interferes with each other.

Second, we consider the node set V_k^2. We divide the whole space into 11 equal cones using 11 rays from v_k, as shown Fig. 3a. If v_a and v_b are in the same cone, then $\angle v_a v_k v_b < 33°$. Let $d_{a,b} = \|v_a - v_b\|$. Since $v_a \in N^{\geq}(v_k, \alpha)$, v_a interfere with some nodes in $R(v_k, \alpha)$, $d_{a,k} \leq r_a + \alpha \cdot r_k$. Similarly, $d_{b,k} \leq r_b + \alpha \cdot r_k$. Thus, $\max\{d_{a,k}, d_{b,k}\} \leq \max\{r_a, r_b\} + \alpha \cdot r_k$. On the other hand, since both v_a and v_b are outside $R(v_k, 3\alpha)$, $\min\{d_{a,k}, d_{b,k}\} \geq 3\alpha \cdot r_k$. As shown in Fig. 3b, for v_a and v_b:

$$\begin{aligned}
d_{a,b}^2 &< d_{a,k}^2 + d_{b,k}^2 - 2\cos(33°) \cdot d_{a,k} \cdot d_{b,k} \\
&= \max\{d_{a,k}, d_{b,k}\}^2 + \min\{d_{a,k}, d_{b,k}\}^2 \\
&\quad - \frac{5}{3}\max\{d_{a,k}, d_{b,k}\} \cdot \min\{d_{a,k}, d_{b,k}\} \\
&\leq \max\{d_{a,k}, d_{b,k}\}\left[\max\{d_{a,k}, d_{b,k}\} - \frac{2}{3}\min\{d_{a,k}, d_{b,k}\}\right] \\
&\leq (\max\{r_a, r_b\} + \alpha \cdot r_k) \cdot [\max\{r_a, r_b\} + \alpha \cdot r_k - 2\alpha \cdot r_k] \\
&\leq \max\{r_a, r_b\}^2 - \alpha^2 \cdot r_k^2 < \max\{r_a, r_b\}^2.
\end{aligned}$$

The transition between the second and third inequalities is because $\max\{d_{a,k}, d_{b,k}\} \leq \max\{r_a, r_b\} + \alpha \cdot r_k$ and $\min\{d_{a,k}, d_{b,k}\} \geq 3\alpha \cdot r_k$. Thus, v_a interferes with v_b. Therefore, each pair of nodes in the same cone interferes with each other. This proves that there exists a node set in V_k^2 with a size of, at least, $|V_k^2|/11$, such that the nodes in the set interfere with each other.

Consequently, there exists a node set with a size of at least

$$\max\{|V_k^1|/(6\alpha + 1)^2, |V_k^2|/11\} \geq \frac{|V_k^1| + |V_k^2|}{(6\alpha + 1)^2 + 11} = \frac{|V_k|}{C_\alpha},$$

such that all nodes in the set interfere with each other. Here, $C_\alpha \leq (6\alpha + 1)^2 + 11$, and we call it the α-hop interference number. Notice that $(6\alpha + 1)^2 + 11$ is an upper bound on C_α, and it can be improved by using a more tight analysis. □

Notice that Theorem 4 works for the interference on nodes only. For a link $e = \mathbf{L}_{i,j}$, let $I^{\geq}(e)$ be the links e' interfering with e under RTS/CTS model and whose radius is not smaller than e. The following theorem shows a counterpart that works for links also:

Theorem 5. *For a given link $e = \mathbf{L}_{i,j}$, at least $|I^{\geq}(e)|/(2C_1)$ timeslots are needed to schedule all links in $I^{\geq}(e)$.*

Proof. For each link $\mathbf{L}_{p,q} \in I^{\geq}(e)$, without loss of generality, we assume that $r_p \geq r_q$. Recall that $e' = \mathbf{L}_{p,q}$ and e interfere by definition, as discussed in the following cases:

Case 1. The interference region of v_p covers either v_i or v_j.

Case 2. The interference region of node v_p can neither cover v_i nor v_j, and v_q is *outside* the union $R(\mathbf{L}_{ij}, 1)$ of interference region of v_i and v_j. Clearly, in this case, v_p must also be outside of $R(\mathbf{L}_{ij}, 1)$. Since e and e' interfere, it must be that the interference region of v_q covers either v_i or v_j.

Case 3. The interference region of node v_p can neither cover v_i nor v_j, and v_q is *inside* the union $R(\mathbf{L}_{ij}, 1)$ of interference region of v_i and v_j. Then, v_p will "interfere" a dummy node v_q.

In summary, we conclude that at least one end node of $\mathbf{L}_{p,q}$ interferes with some nodes in region $R(\mathbf{L}_{i,j}, 1)$, i.e., the head of $\mathbf{L}_{p,q}$ is in $N^{\geq}(\mathbf{L}_{i,j}, 1)$. Recall that $N^{\geq}(\mathbf{L}_{i,j}, 1) = N^{\geq}(v_i, 1) \bigcup N^{\geq}(v_j, 1)$. The head of $\mathbf{L}_{p,q}$ is either in $N^{\geq}(v_i, 1)$ or $N^{\geq}(v_j, 1)$. Without loss of generality, we assume that at least $|I^{\geq}(e)|/2$ heads of the links in $I^{\geq}(e)$ are in $N^{\geq}(v_i, 1)$. From Theorem 4, there are at least $|I^{\geq}(e)|/(2C_1)$ heads that interfere with each other. Thus, there are at least $|I^{\geq}(e)|/(2C_1)$ links in $I^{\geq}(e)$ that interfere with each other. This finishes the proof. □

Consequently, we have the following necessary condition for any interference-free link scheduling under RTS/CTS model:

Lemma 6. *For any timeslot τ, any valid RTS/CTS interference-free link scheduling $\mathcal{S}$ must satisfy that*

$$X_{e,\tau} + \sum_{e' \in I^{\geq}(e)} X_{e',\tau} \leq C_{RTS/CTS},$$

where constant $C_{RTS/CTS} = 2C_1$, and $I^{\geq}(e)$ is the link interfering with e whose radius is not smaller than e.

Notice that above theorems hold for any multihop wireless networks in which both the transmission range and interference range could be heterogeneous and some links could be missing due to various reasons. If the interference range is homogeneous, then the constant C_α could be improved.

Let $\delta(F_G^{D2})$ be the *maximum* δ-degree of all links $\mathbf{L}_k$ in steps 2-3 of Algorithm 1. We now prove that Algorithm 1 has the following performance guarantee:

Theorem 7. *Under RTS/CTS model, Algorithm 1 needs at most $2C_1 \cdot \delta_{opt}$ timeslots for all links without interference, where δ_{opt} is the minimum schedule period T.*

Proof. Let H be the vertex induced subgraph of F_G^{D2} such that each vertex in H has a degree of, at least, $\delta(F_G^{D2})$. The existence of H is straightforward from the definition of $\delta(G)$. Without loss of generality, let $\mathbf{L}_{i,j}$ be the vertex in H with the smallest interference range. From Theorem 5, there exists a clique of size at least $\frac{\delta(F_G^{D2})+1}{2C_1}$ in F_G^{D2}. The optimal solution, thus, needs $\geq \frac{\delta(F_G^{D2})+1}{2C_1}$ colors. Algorithm 1 uses $\leq \delta(F_G^{D2}) + 1$ colors. This finishes our proof. □

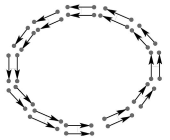


Fig. 4. Bad example for simple greedy.

4.2 Scheduling under fPrIM Model

Kumar et al. [7] studied the scheduling under a different PrIM (with parameter δ): where a transmission by a node v_i is successfully received by a node v_j if and only if $\|v_k - v_j\| \geq (1 + \delta)\|v_i - v_j\|$ for any node $v_k \neq v_i$. This needs every node to dynamically change its transmission power based on receiving node. Recall that, in this paper, we assume that any node will have a fixed transmission power. It is not difficult to design network examples where the methods (processing links in the order of decreasing length) developed in [7] will not work under our model.

Under the RTS/CTS model, we essentially showed that the optimal color assignment needs at least $\delta(F_G^{D2})$ colors. Note that when the graph is modeled by UDG, $\delta(F_G^{D2})$ is essentially $\Delta(F_G^{D2})$, where $\Delta(F_G^{D2})$ is the maximum degree of the conflict graph F_G^{D2}. Thus, almost any greedy-based coloring method (using at most $\Delta(F_G^{D2}) + 1$ colors) has a constant approximation ratio. Several previous literatures claimed the same result (that the optimal coloring needs $\Theta(\Delta(F_G^P))$ colors) under the fPrIM model and proposed some algorithms to color the communication graph G using $O(\Delta(F_G^P))$colors, where $\Delta(F_G^P)$ is the maximum degree of the conflict graph F_G^P under fPrIM model. We can also define $\delta(F_G^P)$ as the maximum δ-degree of the F_G^P which can be computed by applying steps 2-3 of Algorithm 1 on F_G^P. However, as we will show later, there are examples of communication graphs whose optimal coloring needs constant colors, while, on the other hand, both $\Delta(F_G^P)$ and $\delta(F_G^P)$ are $O(n^{1-\epsilon})$ for any $0 \leq \epsilon < 1$ if all nodes have the same transmission range and $t_i = r_i = r$. This shows that any greedy algorithm that uses$\Theta(\Delta(F_G^P))$or even $\Theta(\delta(F_G^P))$ colors could be very bad compared to the optimal solution.

We now describe such an example as in Fig. 4. Here, all nodes have the same transmission range and interference range r. The links formed several groups such that all links in each group are parallel and each link has length r. The groups are placed in a cyclic manner such that any sender of one group interferes with all receivers in the previous group and does not interfere with any other receivers in other groups. The number of links in each group is $n^{1-\epsilon}$, and there are n^{ϵ} groups. Obviously, in the conflict graph F_G^P, the degree of each vertex (corresponding to a physical link) is $n^{1-\epsilon}$. Thus, $\Delta(F_G^P) = \delta(F_G^P) = n^{1-\epsilon}$. On the other hand, we can use at most three colors to color all the links without conflict: we color groups in clockwise order, and all links in the same group are assigned the same color that is the smallest available.

The above example shows that it is unclear whether Algorithm 1 can find a scheduling that approximates the optimal solution when the interference range equals the transmission range (the proof of Theorem 7 does not extend to this scenario). Fortunately, the ratio of the interference range over the transmission range is usually around 2 in practice. Next, we utilize this property to design an efficient link scheduling with a constant approximation ratio.

Given any two nodes $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ in conflict graph F_G^P such that v_j and v_q are receivers, if $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ interfere with each other, then it is possible that 1) v_i interferes v_q, or 2) v_p interferes v_j, 3) or both. If v_p interferes v_j, then we treat the link between $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ as an *incoming link* for $\mathbf{L}_{i,j}$. Similarly, if v_i interferes v_q, we treat the link as an *outgoing link* for $\mathbf{L}_{i,j}$. Let $d_{i,j}^{in}(F_G^P)$ and $d_{i,j}^{out}(F_G^P)$ be the incoming and outgoing degree of $\mathbf{L}_{i,j}$ in the conflict graph F_G^P, respectively. The number of incoming links of a vertex in F_G^P is its incoming degree, and the number of outgoing links are its outgoing degree. Similarly, we define $\Delta^{in}(F_G^P)$ and $\Delta^{out}(F_G^P)$ as the maximum incoming and outgoing degree in graph F_G^P, respectively. When $\gamma_i > 1$ for each node v_i, we can show that the optimal coloring needs at least $\Theta(\Delta^{in}(F_G^P))$ colors, where the hidden constant depending on $\min_i \gamma_i$ (which is typically 2 in practice).

Lemma 8. *Consider any link $\mathbf{L}_{i,j}$, where v_j is the receiver. Consider two links $\mathbf{L}_{p,q}$ and $\mathbf{L}_{s,t}$ that are $\mathbf{L}_{i,j}$'s incoming links in conflict graph F_G^P, where v_q and v_t are the receivers. If $\angle v_q v_j v_t \leq \arcsin\frac{\gamma-1}{2\gamma}$, then link $\mathbf{L}_{p,q}$ interferes with link $\mathbf{L}_{s,t}$.*

Proof. Due to space limit, the detailed proof is omitted. Please refer to Lemma 5 in the conference version [14]. □

Similar to Lemma 6, we have the following necessary condition for interference-free link scheduling under fPrIM model:

Lemma 9. *For any timeslot τ, any valid interference-free link scheduling $\mathcal{S}$ under PrIM must satisfy that*

$$X_{e,\tau} + \sum_{e' \in I^{in}(e)} X_{e',\tau} \leq \left\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \right\rceil,$$

where $I^{in}(e)$ is the set of incoming links of e that interfere e.

This is because for all incoming neighboring links of link e, Lemma 8 implies that there are at most $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$ links that can be scheduled at any same timeslot. Notice that when $\gamma = 1$, $X_{e,\tau} + \sum_{e' \in I(e)} X_{e',\tau}$ could be arbitrarily large as shown by a network example illustrated in Fig. 4. In practice, $\gamma \geq 2$, which implies that $\lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil \leq 25$. We then present our main theorem about the optimum coloring for fPrIM model with $\gamma_i > 1$.

Theorem 10. *Optimal vertex coloring for conflict graph F_G^P needs $\Theta(\Delta^{in}(F_G^P))$ colors if $\min_i \gamma_i$ is some constant > 1.*

Proof. For any link $\mathbf{L}_{i,j}$ such that v_j is the receiver, we partition the space using b equal-sized cones apexed at node v_j, where $b = \lceil \frac{2\pi}{\arcsin\frac{\gamma-1}{2\gamma}} \rceil$. From the Pigeon hole principle, $\mathbf{L}_{i,j}$ has at least $d_{i,j}^{in}(F_G^P)/b$ links whose receivers are in the same cone. From Lemma 8, all links in the same cone interfere with each other. Thus, $\mathbf{L}_{i,j}$ has at least $d_{i,j}^{in}(F_G^P)/b$ incoming links such that they interfere with each other. It implies that any valid coloring will use at least $d_{i,j}^{in}(F_G^P)/b$ among the incoming neighbors of link $\mathbf{L}_{i,j}$. Thus, the optimal coloring needs at least $\Delta^{in}(F_G^P)/b + 1$ colors. □

Note that $\Delta(F_G^P)$ could be arbitrarily larger than $\Delta^{in}(F_G^P)$. Thus, simple greedy algorithm using $\Delta(F_G^P)$ colors does not work, e.g., the algorithm proposed in [1] for UDG networking model. It is known that the optimal coloring can be obtained by using a greedy approach on a certain ordering of vertices in F_G^P. Next, with a careful selection of link ordering, we present our centralized scheduling method (Algorithm 2) that needs at most $2 \cdot \Delta^{in}(F_G^P) + 1$ colors which is asymptotically optimal.

Algorithm 2: Centralized scheduling under fPrIM
Input: A communication graph $G = (V, E)$ of m links.
Output: An interference-free link scheduling.

```
Construct the conflict graph $F_G^P$ and let graph $G' = F_G^P$.
while $G'$ is not empty do
    Find the link $\mathbf{L}_{i,j}$ with the largest $d_{i,j}^{in}(G') - d_{i,j}^{out}(G')$
    in $G'$ and remove this vertex from $G'$ and all its
    incident edges. Let $\mathbf{L}_k$ denote the $k$th vertex
    removed.
Process the sequences of links $\mathbf{L}_{i,j}$ from $\mathbf{L}_m$ to $\mathbf{L}_1$.
Assign each link $\mathbf{L}_k$ the smallest timeslot not yet
assigned to any of its neighbors in $F_G^P$.
```

Theorem 11. *Algorithm 2 uses at most $2 \cdot \Delta^{in}(F_G^P) + 1$ colors.*

Proof. The key observation is that in any directed graph, the sum of all vertices' incoming degree equals the sum of outgoing degree. For the link $\mathbf{L}_{i,j}$ with the largest $d_{i,j}^{in}(G') - d_{i,j}^{out}(G')$ in G', we must have $d_{i,j}^{in}(G') \geq d_{i,j}^{out}(G')$. Thus, when we assign color (or timeslot) for the link $\mathbf{L}_{i,j}$, the subgraph induced by all the links that have already been processed is exactly the subgraph G' right before vertex $\mathbf{L}_{i,j}$ was removed in the **while** loop of Algorithm 2. Therefore, there are at most $2 \cdot d_{i,j}^{in}(G')$ adjacent neighbors of $\mathbf{L}_{i,j}$ in F_G^P that have already been processed. In other words, the smallest timeslot assigned to $\mathbf{L}_{i,j}$ is at most $2 \cdot d_{i,j}^{in}(G') + 1$, which is at most $2 \cdot d_{i,j}^{in}(F_G^P) + 1$. This proves that we need at most $2 \cdot \Delta^{in}(F_G^P) + 1$ timeslots for an interference-free schedule. □

5 Distributed Link Scheduling

In a wireless network, centralized algorithm may not be possible, and even if possible, due to the dynamic features of wireless networks, it is inefficient to update the coloring using a centralized algorithm. Thus, in this section, we design efficient distributed algorithms to get a valid coloring with good performance guarantee.

5.1 Scheduling under RTS/CTS Model

In literatures, several distributed algorithms have been proposed for the vertex coloring. The first solution is to simply apply a distributed vertex coloring on the conflict graph F_G^{D2}. For arbitrary graphs, a $\Delta + 1\text{-coloring}$ can be computed in time $O(\log^* n + \Delta^2)$ or $O(\Delta \log n)$ [8], [20]. Recall that all previous distributed algorithms work for the general graph. By taking advantage of special properties of conflict graph defined here, we are able to obtain a deterministic distributed coloring algorithm that colors the links with $O(\Delta(F_G^{D2}))$ colors in almost constant time when the interference ranges are homogeneous. On the other hand, as shown in our centralized algorithm, the optimal color is $\Theta(\delta(F_G^{D2}))$ which could be much smaller than $\Delta(F_G^{D2})$ when interference ranges are heterogeneous. Thus, simply applying a coloring algorithm with ratio $\Theta(\Delta(F_G^{D2}))$ may not achieve a good performance. The first instinct is to design a distributed version of Algorithm 1. However, finding the node with the global maximum degree iteratively does not seem promising for distributed algorithm. Thus, we need to find some lower bound for the optimal color other than $O(\delta(F_G^{D2}))$.

Given two nodes v_i and v_j, we say that v_i *precedes* v_j if and only if $r_i > r_j$ or $r_i = r_j$ and $i > j$. Given a pair of links $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ with different heads $h_{i,j} \neq h_{p,q}$, we say that $\mathbf{L}_{i,j}$ *precedes* $\mathbf{L}_{p,q}$ if $r_{i,j} > r_{p,q}$ or $r_{i,j} = r_{p,q}$ and $h_{i,j} > h_{p,q}$. Recall that $r_{i,j} = \max\{r_i, r_j\}$. We also say that the corresponding vertex $\mathbf{L}_{i,j}$ precedes $\mathbf{L}_{p,q}$ in the conflict graph in this case. For a vertex $\mathbf{L}_{i,j}$ in graph F_G^{D2}, let $d_{i,j}^{\geq}(F_G^{D2})$ be the number of adjacent vertices that precede $\mathbf{L}_{i,j}$, which is called *efficient degree* of $\mathbf{L}_{i,j}$. From Theorem 5, there are at least $d_{i,j}^{\geq}(F_G^{D2})/(2C_1)$ vertices adjacent to and preceding $\mathbf{L}_{i,j}$ that form a clique in which each vertex (i.e., the corresponding link in the communication graph) interferes with each other. Let $\phi(F_G^{D2}) = \max_{\mathbf{L}_{i,j}} d_{i,j}^{\geq}(F_G^{D2})$, then Theorem 5 shows that optimal coloring algorithm needs at least $\phi(F_G^{D2})/(2C_1)$ colors. Thus, finding a coloring algorithm using at most $\Theta(\phi(F_G^{D2}))$ colors is a constant-ratio approximation algorithm. Unlike the centralized Algorithm 1 in which the lower bound of $\delta(F_G^{D2})$ could not be found by using only local information, the lower bound of $\phi(F_G^{D2})$ could be easily obtained by any link $\mathbf{L}_{i,j}$ by simply counting the number of interfering links that precede itself, i.e., with larger link interference radius. Algorithm 3 presents our distributed coloring method that uses at most $\phi(F_G^{D2})$ colors.

Algorithm 3: Distributed coloring algorithm for RTS/CTS model

Input: A communication graph $G = (V, E)$.

Output: A valid coloring of all links.

```
Each node $v_i$ collects all communication links, say $H_i$,
  that contain $v_i$ as the head, i.e., all links $\mathbf{L}_{i,j}$ with $r_i \geq r_j$.
Each node $v_i$ collects all communication links, denoted
  by $M_i$, that are not in $H_i$ and interfere with some links $H_i$.
Node $v_i$ finds $M_i^+$, which is the subset of links in $M_i$
  that precedes every link in $H_i$ and let $M_i^- = M_i - M_i^+$.
Node $v_i$ sets all links in $M_i^+$ as uncolored.
while some links in $M_i^+$ are uncolored do
    Node $v_i$ listens messages from other nodes.
    if $v_i$ receives a message $\mathrm{Color}(p, q, k)$ then
        Node $v_i$ marks $\mathbf{L}_{p,q}$ with color ID $k$ if $\mathbf{L}_{p,q}$ is in $M_i^+$.
for each node $v_j$ in $H_i$ do
    Find the color with minimum color ID, say $k$, that
      is not used by any link that is conflicted with $\mathbf{L}_{i,j}$.
      Color link $\mathbf{L}_{i,j}$ with color ID $k$.
    Sends the message $\mathrm{Color}(i, j, k)$ to all heads of the
      links adjacent to $\mathbf{L}_{i,j}$ in $M_i^-$.
```

Theorem 12. *Algorithm 3 computes a valid coloring using at most $\phi(F_G^{D2})$ colors, which is asymptotically optimal.*

Proof. First, we show that the algorithm does terminate. Since it is straightforward that the number of nodes in H_i is bounded by $\phi(F_G^{D2})$, the **for** loop terminates in $O(n)$ iterations. Thus, the maximum time needed for all other processes other than **while** loop is bounded by a finite time T and our main focus is to show that the **while** loop does terminate for any node v_i. Let $(v_{\sigma_1}, v_{\sigma_2}, \ldots, v_{\sigma_n})$ be the sorted list of nodes in the decreasing order of their interference range. Thus, v_{σ_i} precedes v_{σ_j} if and only if $i < j$. Since v_{σ_1} precedes every other nodes, $M_{\sigma_1}^+$ is empty and v_{σ_1} colors all links that are adjacent to v_{σ_1} in time T. Now, consider the nodes v_{σ_2} and $M_{\sigma_2}^+$. If $\mathbf{L}_{p,q} \in M_{\sigma_2}^+$, then either v_p or v_q is v_{σ_1}. Thus, all links in $M_{\sigma_2}^+$ are colored. Therefore, all links that are adjacent to v_{σ_2} are colored before time $2T$. Similarly, all links that are adjacent to v_{σ_j} are colored before time $j \cdot T$. Thus, all links are colored in time $n \cdot T$. It is straightforward to show that, by assuming color one link takes a unit time, the running time of this algorithm is at most m, where m is the number of directed communication links.

Second, we show that the computed coloring is valid, i.e., no two conflict links have the same color. Consider conflict links $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$, as discussed in the following cases:

Case 1. $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ have the same head. Without loss of generality, we assume that $v_i = v_p$ is the head of the links. Thus, both $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ are in H_i. Therefore, $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ have different colors.

Case 2. $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ have different heads. Then, without loss of generality, we can assume that $h_{i,j} = i$, $h_{p,q} = p$ and v_i precedes v_p. Since $\mathbf{L}_{i,j} \in M_p^+$, $\mathbf{L}_{i,j}$ is colored before M_p^+ becomes empty. Thus, $\mathbf{L}_{p,q}$ is colored after $\mathbf{L}_{i,j}$ is. Therefore, when v_p colors $\mathbf{L}_{p,q}$, it uses a color that is different from the color of $\mathbf{L}_{i,j}$ based on our algorithm.

Third, it is straightforward that Algorithm 3 uses at most $\phi(F_G^{D2})$ colors, i.e., it has a constant approximation ratio. □

Notice that in Algorithm 3, we start to color a link after all interfering links preceding it are colored. Thus, in the worst case, it may take time $O(n)$ to color all the links, where n is the number of nodes in the network. Here, we assume that in one time unit, a node can color all its incident links. Comparing with previous polylogarithmic time distributed coloring algorithms that color the graph using $\Delta(F_G^{D2})$ colors, Algorithm 3 may take longer time. However, following example shows that $\Delta(F_G^{D2})$ could be as large as $O(n)$ times of the color used by Algorithm 3,

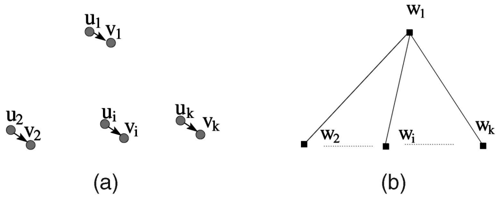


Fig. 5. Δ could be $\Theta(n)$ of number of colors used by Algorithm 3. (a) The original network. (b) The conflict graph.

where n is the number of the nodes in original network. In Fig. 5a, there are k pairs of transmission links $u_1 v_1, \ldots, u_n v_n$. Nodes u_1, v_1 have interference range 1, and all other nodes have interference range ϵ, where ϵ is a small positive constant such that node u_i does not interfere v_j for $i, j > 1$. The corresponding conflict graph is shown in Fig. 5b. It is not difficult to see that we only need two colors while the degree of $\mathbf{L}_{1,1}$ is $n-1$. In other words, compared with previous polylogarithmic time methods with $\Omega(n)$ approximation ratios, our method has a constant approximation ratio using a larger worst case running time.

5.2 Faster Scheduling under RTS/CTS Model

Although Algorithm 3 computes a coloring that is at most constant times of the optimal, it may need linear number of rounds to compute the coloring. In certain circumstances, we would prefer the distributed algorithms that run fast to the distributed algorithms that have good performance as long as the fast distributed algorithm does not perform much worse. In the following, we present another distributed algorithm that computes the coloring very fast with a good performance guarantee of $O(\log(\psi)+1)$, where ψ is the ratio between the maximum interference range over the minimum interference range among all nodes.

Algorithm 4: Fast distributed coloring algorithm for RTS/CTS model
Input: A communication graph $G=(V,E)$.
Output: A valid coloring of the communication graph.

```
Node $v_i$ computes a subset, say $H_i$, of all communication
  links containing $v_i$ such that link $\mathbf{L}_{i,j} \in H_i$ if and only if
  $r_i > r_j$.
while node $v_i$ failed to obtain the channel do
    Node $v_i$ monitors the channel and competes for it.
for each link $\mathbf{L}_{i,j} \in H_i$ do
    Color link $\mathbf{L}_{i,j}$ with the smallest color ID, say $k$, that
      is not used by any link that conflicts with $\mathbf{L}_{i,j}$.
    Broadcasts the message Color$(i,j,k)$ to each head of
      links that conflict with $\mathbf{L}_{i,j}$.
```

Algorithm 4 assumes that there is a certain competition-based MAC layer (e.g., 802.11 with RTS/CTS) available for a node to obtain the channel. We use this MAC mechanism to obtain a link scheduling that is efficient and interference free. Algorithm 4 is very simple and can be implemented without much additional computation on each node. However, the proof of the performance guarantee is not straightforward. To prove the main theorem, we need some notation in order to extend Theorems 4 and 5. For a given node v_k, let $N^{\geq}(v_k, \alpha, \beta)$ be a node set composed of the nodes satisfying that 1) each of their interference radius is at least $\frac{r_k}{\beta}$ and 2) each of them interferes some nodes in $R(v_k, \alpha)$. Let $N^{\geq}(\mathbf{L}_{i,j}, \alpha, \beta)$ be the union of $N^{\geq}(v_i, \alpha, \beta)$ and $N^{\geq}(v_j, \alpha, \beta)$. The proofs of Lemmas 13 and 14 are similar to the proofs of Theorems 4 and 5, respectively, and, thus, are omitted here.

Lemma 13. *For any node v_k and any set $V_k \subseteq N^{\geq}(v_k, \alpha, \beta)$, there exists a subset V_k' of V_k with cardinality at least $\lceil |V_k| / C_{\alpha,\beta} \rceil$ such that nodes in V_k' interfere with each other, where $C_{\alpha,\beta} = (6\alpha\beta + 1)^2 + 11$.*

Lemma 14. *For any link $\mathbf{L}_{i,j}$ and any set $V_{ij} \subseteq N^{\geq}(\mathbf{L}_{i,j}, \alpha, \beta)$, there exists a subset V_{ij}' of V_{ij} with cardinality at least $\lceil V_{ij} / (2C_{\alpha+1,\beta}) \rceil$ such that links in V_{ij}' interfere with each other.*

Let $\Delta(\alpha, \beta) = \max_{\mathbf{L}_{i,j}} |N^{\geq}(\mathbf{L}_{i,j}, \alpha, \beta)|$ and $\chi(F_G^{D2})$ be the optimal number of colors. Based on Lemma 14, the following theorem is straightforward, for any fixed α, β:

Theorem 15. $\chi(F_G^{D2}) \geq \lceil \Delta(\alpha, \beta) / (2C_{\alpha+1,\beta}) \rceil$.

We then present our main theorem for our fast distributed coloring method.

Theorem 16. *Algorithm 4 computes a coloring that is at most $O(\log(\psi)+1)$ times of optimum $\chi(F_G^{D2})$.*

Proof. Without loss of generality, let link $\mathbf{L}_{i,j}$ be the link that has the maximum color ID, say $\mathbf{g}$. To prove the theorem, we will show that $\mathbf{g} \leq 2C_{1,2} \cdot (\log(\psi)+1) \cdot \chi$. In the following, we prove it by contradiction and, for the sake of contradiction, assume that $\mathbf{g} > 2C_{1,2} \cdot (\log(\psi)+1) \cdot \chi$.

We first argue that for any $0 \leq k \leq \log(\psi)$, there exists a link $\mathbf{L}_{i^{(k)},j^{(k)}}$ such that $r_{i^{(k)},j^{(k)}} < r_{i,j}/2^k$ and its color ID is not smaller than $\mathbf{g} - 2C_{1,2} \cdot k \cdot \chi$. We prove this argument by induction on k. If $k=0$, then the argument trivially holds. Assume for $k \leq p$, the argument holds. From Theorem 15, by letting $\alpha = 0$ and $\beta = 2$, $\chi \geq \Delta(0,2)/(2C_{1,2})$. In other words, the number of links, which interfere or are interfered by link $\mathbf{L}_{i^{(p)},j^{(p)}}$ and whose radius is not smaller than $r_{i^{(p)},j^{(p)}}/2$, is at most $2C_{1,2} \cdot \chi$. Thus, there must exist a link $\mathbf{L}_{i^{(p+1)},j^{(p+1)}}$ such that

1. $\mathbf{L}_{i^{(p+1)},j^{(p+1)}}$ interferes or is interfered by $\mathbf{L}_{i^{(p)},j^{(p)}}$;
2. $r_{i^{(p+1)},j^{(p+1)}} < r_{i,j}/2^{p+1}$; and
3. $\mathbf{L}_{i^{(p+1)},j^{(p+1)}}$'s color ID is at least $\mathbf{g} - 2C_{1,2} \cdot (p+1) \cdot \chi$.

This finishes the induction.

Thus, let $k = \lfloor \log(\psi) \rfloor$, link $\mathbf{L}_{i^{\lfloor \log(\psi) \rfloor},j^{\lfloor \log(\psi) \rfloor}}$ has the color ID not smaller than $\mathbf{g} - 2C_{1,2} \cdot \lfloor \log(\psi) \rfloor \cdot \chi$. This implies that $\mathbf{L}_{i^{\lfloor \log(\psi) \rfloor},j^{\lfloor \log(\psi) \rfloor}}$ has at least $2C_{1,2} \cdot \chi + 1$ adjacent links. Since, $r_{i^{(\lfloor \log(\psi) \rfloor)},j^{(\lfloor \log(\psi) \rfloor)}} < r_{i,j}/2^{\lfloor \log(\psi) \rfloor)}$ and $r_{p,q} \geq r_{i,j}/2^{\log(\psi)}$, all links that interfere or are interfered by link $\mathbf{L}_{i^{\lfloor \log(\psi) \rfloor},j^{\lfloor \log(\psi) \rfloor}}$ have interference radius at least $r_{i^{\lfloor \log(\psi) \rfloor},j^{\lfloor \log(\psi) \rfloor}}/2$. From Lemma 14, $\chi \geq \lceil \frac{2C_{1,2} \cdot \chi + 1}{2C_{1,2}} \rceil \geq \chi + 1$, which is a contradiction. Thus, $\mathbf{g} \leq 2C_{1,2} \cdot (\log(\psi)+1) \cdot \chi$. This finishes the proof. □

Algorithm 4 essentially is a **First-Fit** coloring method. It has been proved in [21] that, any **First-Fit** coloring of an d-inductive graph with n nodes will produce a coloring using colors of, at most, $O(d \log n)$ times of the optimum. Here, a graph G is d-inductive if we can number the vertices such that each node has at most d edges connected to the nodes with larger numbers. We essentially proved previously that graphs F_G^P and F_G^{D2} are d-inductive graphs for some constants d. Thus, we have the following theorem:

Theorem 17. *Algorithm 4 computes a coloring that is at most $O(\min(\log n, 1 + \log \psi)$ times of optimum $\chi(F_G^{D2})$.*

Notice that, in Algorithm 4, a node can start assigning timeslots to its incident links as long as it obtained the communication channel. Thus, the time complexity of this algorithm will be much close to the node coloring number of the communication graph G, in which two interfering nodes should be assigned different colors. Notice that, it was proved in [22] that for disk graphs, the tight bound for approximation ratio for online coloring of disk graphs is $\min\{\log n, \log \psi\}$. Thus, we know that it is impossible to design distributed algorithm for link scheduling with better asymptotic approximation ratio when no any ordering are allowed among links.

5.3 Scheduling under fPrIM Model

From Theorem 11, any coloring algorithm that uses $O(\Delta^{in}(F_G^P))$ colors under the fPrIM model has a constant approximation ratio. Here, we give a distributed algorithm (Algorithm 5) that bears the similar idea of our centralized method (Algorithm 2).

Algorithm 5: Distributed scheduling for fPrIM model

```
Input: A communication network $G = (V, E)$.
Output: A valid coloring of all links.
Assign each communication link a label WHITE.
The header of each communication link $\mathbf{L}_{i,j}$ collects all
  incoming links and outgoing links, denoted by $M_{i,j}^{in}$
  and $M_{i,j}^{out}$.
while link $\mathbf{L}_{i,j}$ is WHITE do
    Link $\mathbf{L}_{i,j}$ monitors the channel.
    If some link $e$ in $M_{i,j}^{in} \bigcup M_{i,j}^{out}$ announces that it
      becomes GRAY with timestamp $k$, link $\mathbf{L}_{i,j}$ locally
      stores the label of link $e$ as GRAY and the
      timestamp $k$.
    if the number of WHITE links in $M_{i,j}^{in}$ is not smaller
      than the number of WHITE links in $M_{i,j}^{out}$ then
        Link $\mathbf{L}_{i,j}$ competes for the channel.
        if Link $\mathbf{L}_{i,j}$ obtains the channel then
            Link $\mathbf{L}_{i,j}$ labels itself GRAY with a timestamp
              $t + 1$ where $t$ is the maximum timestamp of all
              GRAY links stored locally. Here $t = 0$ is no
              GRAY links are stored. Link $\mathbf{L}_{i,j}$ send to all
              adjacent links in $F_G^P$ the message that $\mathbf{L}_{i,j}$
              becomes GRAY with the timestamp $t + 1$. Link
              $\mathbf{L}_{i,j}$ makes a list of links $S_{i,j}$ composed of the
              current WHITE links in $M_{i,j}^{in} \bigcup M_{i,j}^{out}$.
while there exists some links in $S_{i,j}$ not colored do
    Link $\mathbf{L}_{i,j}$ listens to the announcement. If a link $e'$
      in $S_{i,j}$ announces its color, then link $\mathbf{L}_{i,j}$ locally
      updates the status of $e'$ as colored together with
      the color of $e'$.
Link $\mathbf{L}_{i,j}$ colors itself using the smallest color available
  that will not produce any conflict with links in $S_{i,j}$. It
  then sends to all adjacent links in $F_G^P$ without a color
  the message about its current color assigned.
```

Theorem 18. *Algorithm 5 computes a valid coloring with at most $2 \cdot \Delta^{in}(F_G^P) + 1$ colors with $O(m)$ messages, where m is the number of communication links.*

Proof. Notice that for each link $\mathbf{L}_{i,j}$, Algorithm 5 uses the smallest color that is not used by any links in $S_{i,j}$. Since the number of incoming links is not smaller than the outgoing links in $S_{i,j}$, link $\mathbf{L}_{i,j}$ is colored with a color not greater than $2 \cdot d_{i,j}^{in}(F_G^P) + 1$. Thus, Algorithm 5 computes a valid coloring with at most $2 \cdot \Delta^{in}(F_G^P) + 1$ colors. Note that each link $\mathbf{L}_{i,j}$ only announces twice in our distributed scheduling algorithm: when it becomes GRAY and when it is colored. Thus, the overall message complexity is $O(m)$. □

Notice that our faster distributed algorithm for RTS/CTS interference model can also be used for the fPrIM here. Using similar proof techniques, we can also prove the following result: our faster distributed coloring algorithm computes a coloring that is at most $O(\min(\log n, 1 + \log \psi)$ times of optimum $\chi(F_G^P)$.

6 Scheduling with Traffic and Schedulable Flows

6.1 Scheduling with Traffic Load

In TDMA system, the minimization of the number of colors is closely related to the maximization of the network throughput. One intrinsic assumption behind the idea of coloring is that each communication link has the same packet arrive rate, i.e., the number of traffics that need to go through each communication link is same. However, this is not likely to be true, and it is possible that some communication link carries more traffic than others, e.g., when joint routing and link scheduling is performed. In [14], we show that simple adaptation of minimum coloring to schedule link transmissions will produce a network throughput that is arbitrarily smaller than the optimum. Thus, we need to generalize the coloring that can take the traffic rate on each communication link into account. In this paper, we use the *weighted coloring* to capture this, which is defined as follows:

Definition 1. *Given a graph $G = (V, E)$ where V is the set of vertices and E is the set of links. Every link $e_i \in E$ has an integral weight $w_i \geq 0$. A weighted link coloring is an assignment of at least w_i distinct colors to each link e_i such that no two links sharing the same color interfere with each other.*

By introducing the notation of weighted coloring, we can assign different weight to different communication links. For example, given a set of k flow requirements f_i from s_i to t_i, $1 \leq i \leq k$, a certain routing algorithm will determine the routing path for each flow. The weight of a link e is then the total flow passing through e divided by the bandwidth $\mathbf{c}(e)$ of link e. In the following, we show how to obtain a valid weighted coloring based on the unweighted coloring (Algorithm 6).

Algorithm 6: Weighted coloring algorithm based on unweighted coloring algorithm $\mathcal{A}$

```
Input: A communication graph $G = (V, E)$ with weight on
  each link and an unweighted coloring algorithm $\mathcal{A}$.
```

```
Output: A valid coloring of the links.
Build the conflict graph $F_G$ based on original graph $G$ and interference model. Assign weight $w_{i,j}$ to vertex $\mathbf{L}_{i,j} \in F_G$.
Construct a new conflict graph $F'_G$ from $F_G$ as follows: for each vertex $\mathbf{L}_{i,j}$ with weight $w_{i,j}$, we create $w_{i,j}$ vertices, $\mathbf{L}^1_{i,j}, \mathbf{L}^2_{i,j}, \ldots, \mathbf{L}^{w_{i,j}}_{i,j}$ and add them to $F'_G$. Add to graph $F'_G$ the edges connecting $\mathbf{L}^a_{i,j}$, $\mathbf{L}^b_{i,j}$ for $1 \le a < b \le w_{i,j}$. Add to graph $F'_G$ an edge between $\mathbf{L}^a_{i,j}$ and $\mathbf{L}^b_{p,q}$ if and only if there is an edge between $\mathbf{L}_{i,j}$ and $\mathbf{L}_{p,q}$ in graph $F_G$.
Run the unweighted vertex coloring algorithm $\mathcal{A}$ on $F'_G$.
Assign link $\mathbf{L}_{i,j}$ all the colors that are used by $\mathbf{L}^k_{i,j}$ for $1 \le k \le w_{i,j}$ in $F'_G$.
```

We show that Algorithm 6 has a performance guarantee that is not worse than that of the unweighted coloring algorithm $\mathcal{A}$.

Theorem 19. *If $\mathcal{A}$ uses at most α times of the optimal colors for unweighted coloring, then Algorithm 6 also needs at most α times of the optimal colors for weighted coloring.*

Proof. Notice that for any valid weighted coloring for F_G, $\mathbf{L}_{i,j}$ is assigned at least $w_{i,j}$ colors. By assigning each vertex $\mathbf{L}^k_{i,j}$ in F'_G a distinct color that is assigned to $\mathbf{L}_{i,j}$, we obtain a valid unweighted coloring for F'_G. Thus, $\chi(F'_G) \le \chi(F_G)$. Here, $\chi(F'_G)$ is the minimum number of colors needed for unweighted coloring in F'_G and $\chi(F_G)$ is the minimum number colors needed for weighted coloring in F_G. Since $\mathcal{A}$ will return a coloring with at most $\alpha \cdot \chi(F'_G)$ colors, Algorithm 6 produces a coloring with at most $\alpha \cdot \chi(F'_G) \le \alpha \cdot \chi(F_G)$ colors. This finishes the proof. □

The basic idea of Algorithm 6 is to create a clique of size $w_{i,j}$ for each link $\mathbf{L}_{i,j}$ and color the new graph using unweighted coloring method $\mathcal{A}$. Although this gives a general framework to design weighted coloring, its time complexity could be large if the weight is large. Fortunately, Algorithm 6 could be simplified without much overhead compared to the unweighted algorithm: the main idea is to assign colors for one link at once. Instead of assigning one timeslot to a link $\mathbf{L}_k$, we assign w_k timeslots to link $\mathbf{L}_k$ when process link $\mathbf{L}_k$. As an example, we modify Algorithm 4 to obtain a fast weighted coloring (Algorithm 7). In the following, we show that Algorithm 7 has the same performance guarantee as Algorithm 4:

```
Algorithm 7: Fast distributed weighted coloring algorithm
Input: A communication graph $G = (V, E)$.
Output: A valid coloring of links in the communication
        graph.
Node $v_i$ computes a subset, say $H_i$, of all communication links containing $v_i$ such that link $\mathbf{L}_{i,j} \in H_i$ if and only if $r_i > r_j$.
while node $v_i$ failed to obtain the channel do
    Node $v_i$ monitors the channel and competes for the channel.
for each link $\mathbf{L}_{i,j} \in H_i$ do
    Color link $\mathbf{L}_{i,j}$ with the first fit $w_{i,j}$ colors that are not used by any link that interferes or is interfered by $\mathbf{L}_{i,j}$. Here, the assigned colors are not required to be continuous.
    Broadcasts the message $\mathrm{Color}(i, j, k)$ to each head of links that conflict with $\mathbf{L}_{i,j}$.
```

Theorem 20. *Algorithm 7 finds a coloring that needs at most $O(\log(\psi) + 1)$ times of optimum.*

Proof. Let $\mathcal{A}_w$ be the coloring algorithm by applying Algorithm 6 based on Algorithm 4. Observe that the coloring of $\mathcal{A}_w$ is nondeterministic, i.e., the output could be different because of the randomization introduced by the different processing time of different nodes. However, it is true that the output of Algorithm 7 is one of the possible outputs of $\mathcal{A}_w$. From Theorem 19, any coloring output by $\mathcal{A}_w$ is at most $O(\log(\psi) + 1)$ times the optimal. Thus, Algorithm 7 computes a coloring that needs at most $O(\log(\psi) + 1)$ times optimal color. □

Similarly, we can modify Algorithms 1 and 3 to obtain efficient weighted coloring methods with the same time complexities and approximation ratios. Theorem 1 directly follows from the above two theorems.

6.2 Necessary and Sufficient Conditions for Schedulable Flows

Similar to [1], [3], and [7], we also make the connection with flows on the links of a wireless network G and the link scheduling. We give both a necessary and a sufficient condition on the link flows such that an interference-free link scheduling is feasible. Recall that we use $f(e)$ and $\mathbf{c}(e)$ to denote the load and the capacity of a link e, respectively. From Lemma 6 and Theorem 7, it follows that:

Theorem 21. *Under the RTS/CTS model, any link flow f that permits an interference-free link scheduling must satisfy the constraint $\frac{f(e)}{\mathbf{c}(e)} + \sum_{e' \in I^{\ge}(e)} \frac{f(e')}{\mathbf{c}(e')} \le 2C_1$. On the other hand, if $\frac{f(e)}{\mathbf{c}(e)} + \sum_{e' \in I^{\ge}(e)} \frac{f(e')}{\mathbf{c}(e')} \le 1$, then any link flow f permits an interference-free link scheduling.*

Similarly, under the fPrIM Model, we have:

Theorem 22. *Under the fPrIM model, any link flow f that permits an interference-free link scheduling must satisfy the following constraint $\frac{f(e)}{\mathbf{c}(e)} + \sum_{e' \in I^{in}(e)} \frac{f(e')}{\mathbf{c}(e')} \le \lceil \frac{2\pi}{\arcsin \frac{\gamma-1}{2\gamma}} \rceil$. On the other hand, if $\frac{f(e)}{\mathbf{c}(e)} + \sum_{e' \in I^{in}(e)} \frac{f(e')}{\mathbf{c}(e')} \le 1$, then any link flow f permits an interference-free link scheduling.*

The proofs of the above theorems are similar to those of [1], [3], and [7] for other interference and networking models, and are, thus, omitted here.

7 Performance Evaluation of Scheduling

Evaluation of Our Scheduling Algorithms: We first evaluate the performances of our scheduling algorithms for RTS/CTS model via simulations with random networks.

Network settings: In these simulations, we randomly generate n wireless nodes uniformly in a 10×10 unit region. The transmission range is randomly drawn from 1.8 to 2 unit, while the interference range is randomly set to be

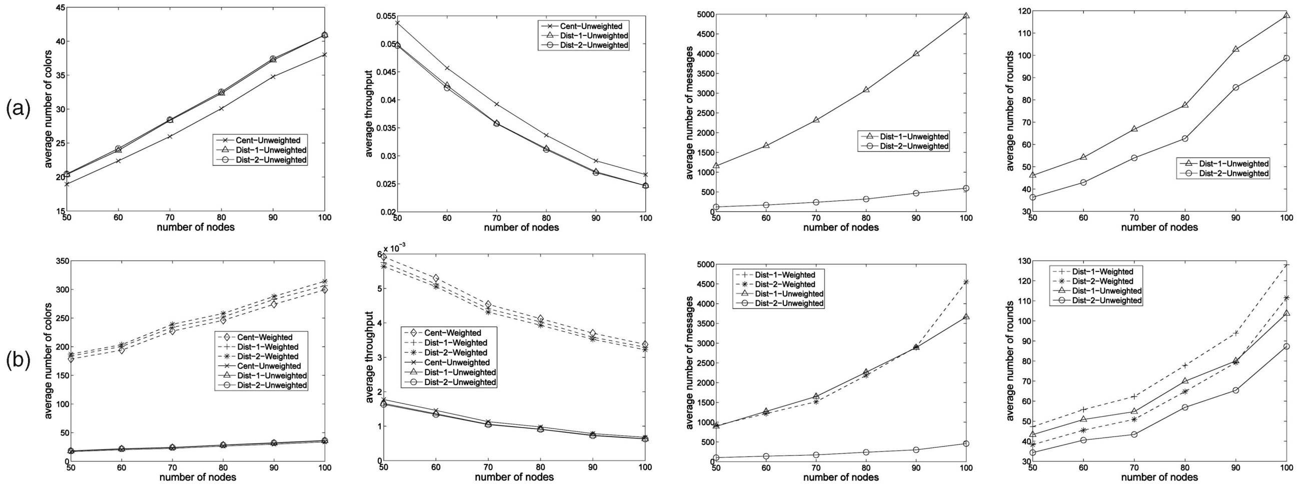


Fig. 6. (a) Scheduling without traffic load information. (b) Scheduling with nonuniform traffic load.

1.5 to 2 times of its transmission range. Typically, a unit represents about 50 m here. We assume there is a sink (or an access point) in the network, all traffics are toward it. The sink is placed in the center of the region in the simulations. We vary the node number n from 40 to 200. For each number n, 100 vertex sets (networks) are randomly generated. Given a sampled network, we not only test the number of colors and the network throughput resulted by our various link-scheduling algorithms but also count the number of messages and rounds used by the distributed algorithms. The average of these performances over all these 100 randomly sampled networks is reported. For each source, we run the classical shortest path algorithm to determine the traffic route. Notice that our scheduling algorithms do not rely on any particular routing algorithms, here the shortest path routing is used as an example.

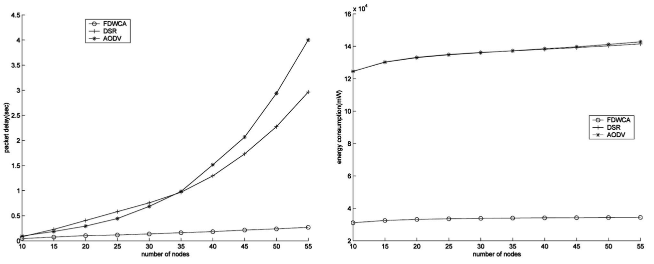
In the first scenario, we assume the system does not know the volume of each traffic. So it is an unweighted case where we need to assign one color for each link involved in the traffics. We test our centralized and two distributed algorithms (Algorithm 1 [Cent], Algorithm 3 [Dist-1], and Algorithm 4 [Dist-2]). The simulation results are reported in Fig. 6a. First, for the number of colors and the throughput, three algorithms have similar performances. When the node number increases, more colors are needed and the throughput decreases. The centralized algorithm has the best throughput, while the fast distributed algorithm has the worst, as our expectation. For both distributed algorithms, we also count the number of messages and rounds used. It shows that Dist-1 algorithm used much more messages and rounds than Dist-2 (fast distributed algorithm). The large number of rounds and messages needed by Dist-1 is due to the first two steps in Algorithm 3, which collect all communication links in H_i and M_i. The large number of rounds of Dist-1 is mainly due to conflicts among messages for collecting information. Notice that two adjacent links in the conflict graph need to compete for the channel first. After a node v_i obtained the channel, it uses a unit of time to assign colors to all links in H_i and inform other interfering links about the coloring used.

In the second scenario, we randomly draw the traffic produced by each node from 1 to 10 units. Then, for each link $\mathbf{L}_{i,j}$, its weight $w_{i,j}$ is the total volumes of traffics that need to go through it, which could be 0. The simulation results are given in Fig. 6b. The throughputs of weighted methods are much better than those of unweighted methods. Our centralized and distributed methods have similar throughput.

BENEFITS OF OUR SCHEDULING METHODS: We then evaluate the performances of our distributed link-scheduling algorithms by conducting simulations in QualNet 3.9 [23]. Notice that Algorithm 4 is a special case of Algorithm 7. Thus, we only evaluate the performance of Algorithm 7 (fast distributed weighted coloring algorithm) based on RTS/CTS model, hereafter called FDWCA, by comparing it with DSR [24] and AODV [25] approach. In FDWCA, we run the classical shortest path algorithm to determine the routing path. Here, our goal is to show that proper link scheduling can conserve energies and improve network performance. DSR and AODV are contention based, without link scheduling.

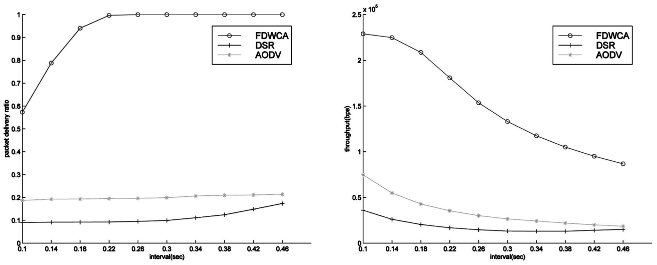
Network settings. We randomly generate n nodes in $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ square area, and adopt $802.11b$ as physical and MAC layer model. In $802.11b$ model, the transmission data rate is set to 2 Mbps, maximum transmission power is 15.0 dBm, and receive sensitivity is $-89.0\ \mathrm{dBm}$. We simulate periodical traffic from all nodes to a single sink using Constant Bit Rate (CBR) scenario, with a packet size of 128 bytes each. The slot duration is set to 10 ms in FDWCA. We evaluate different methods by comparing *packet delay*, *average energy consumption*, *packet delivery ratio*, and *network throughput*. Clearly, the delay and delivery ratio criteria reflect the network throughput. Higher packet delivery ratio and lower packet delay mean better network throughput. Average energy consumption is the average energy cost to delivery a certain number of packets from sources to sink. We calculate the average energy consumption of all nodes.

We first evaluate the packet delay and energy consumption, by fixing the reporting interval at 5 seconds and varying the network size from 10 to 55. For each specific network size, we generate 50 samples to calculate the average performance. The results are shown in Fig. 7. In Fig. 7a, as the network size grows linearly, packet delays in



(a) (b)

Fig. 7. Packet delay and energy consumption per round, when the network varies size in [10, 55] range and fixes the reporting interval at 5 seconds. (a) Packet delay. (b) Energy consumption.



(a) (b)

Fig. 8. Packet delivery ratio and network throughput per round, when a 40-node network varies reporting interval in range of [0.10, 0.46] seconds. (a) Packet delivery ratio. (b) Network throughput.

DSR and AODV approaches grow exponentially, while it increases near linearly in our FDWCA algorithm. In DSR and AODV, the large delay is mainly caused by the random resource competition of nodes, since every node is trying to send data to its parent node and eventually to the sink. The communication is not coordinated. Fig. 7b shows the comparison of average energy consumption for these three approaches. The advantage of FDWCA is obvious. DSR and AODV cost more energy than FDWCA, because enormous media contention wastes energies.

To evaluate packet delivery ratio, we fix the network size at 40 and vary the reporting interval from 0.10 to 0.46 second. The comparison of three methods is shown in Fig. 8a. As the reporting interval increases (or reporting frequency decreases), the packet delivery ratio in FDWCA algorithm reaches 100 percent after reporting interval increases to around 0.22 second, while in DSR and AODV, it reaches 100 percent fairly slow. Our simulations (figure not reported here due to space limit) show that only when the network load is very low (the node's reporting interval is around 2 seconds), AODV and DSR start to have 100 percent delivery ratio. In many applications, 100 percent delivery ratio need be guaranteed. That is to say, FDWCA can enable finer scale data collection (e.g., more frequent data sampling is possible) than DSR and AODV. Fig. 8b shows that our scheduling method FDWCA achieves much better throughput than both AODV and DSR. Notice that for reporting interval of 0.1 second with each reporting packet of size 128 bytes, the total traffic demand produced by all $n = 40$ nodes is $\frac{1}{0.1} \cdot n \cdot 1\ \text{Kbps} = 400\ \text{Kbps}$. The delivery ratio is about 58 percent, which matches the observed throughput at about 230 Kbps in Fig. 8b.

The above simulation is based on the 802.11 networks, where MAC protocol is a variation of CSMA/CA. Our scheduling algorithm may work better if a TDMA-like MAC protocol is used. Overall, the simulation shows the overwhelming advantage of proper link scheduling: it not only increases network throughput but also reduces the energy consumption.

8 Related Work

Scheduling has been studied extensively in the past few years due to its application for assigning timeslots in TDMA MAC protocols. Scheduling can be reduced to different coloring problems: *edge coloring* and *vertex coloring*.

Edge coloring, in which every edge corresponds to a valid communication link, is a natural way to capture the link-scheduling problem. An edge coloring is *valid* if no two incident edges share the same color. Vizing's theorem [26] states that a valid edge coloring for an *indirected* graph can be obtained by using at most $\Delta + 1$ colors, where Δ is the maximum node degree in the graph. On the other hand, any edge coloring needs at least Δ colors. Any edge coloring that uses $\Theta(\Delta)$ colors is close to the optimal. Panconesi and Srinivasan [27] proposed a randomized distributed edge coloring method that uses at most $2\Delta + 1$ colors. To some extent, this captures some transmission restrictions in ad hoc and sensor network in which no node can receive or send at the same timeslot, but it did not address some other interferences such as secondary interference. When one has a valid edge coloring, it can be easily mapped to a TDMA scheduling. However, it is possible that two communication links sharing the same color still interfere with each other in a wireless network. In order to remedy this, Gandham et al. [28] proposed to use a two-phase scheduling method: in the first phase, a distributed valid edge coloring is obtained, and in the second phase, a valid scheduling taken into account the secondary interference is obtained. In essence, [28] is based on the PrIM. The overall scheduling in [28] only provided a performance guarantee when the conflicting links form a tree. Jain et al. [13] proposed a new concept *conflict graph* that captures the interference in a wireless network.

Vertex coloring is one of the most fundamental NP-hard problems in graph theory and has been thoroughly studied. A vertex coloring is *valid* if and only if any two adjacent vertices receive different colors. The minimum number that is needed for a valid vertex coloring for a graph G is known as the *chromatic number* $\chi(G)$. It is known that for general graph, the chromatic number cannot be approximated within $n^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $\text{ZPP} = \text{NP}$ [29]. For vertex coloring of a general graph G, it was proved that, every graph G can be colored using $\delta(G) + 1$ colors. Then, Hochbaum [19] presented a method to find the value of $\delta(G)$ and color G using $\delta(G) + 1$ colors in $O(|V| + |E|)$ time. Ramanathan [9] proposed a unified framework for TDMA, FDMA, and CDMA-based multi-hop wireless networks. They also proposed a timeslot assignment to edges; the number of timeslots required is at most $O(\theta)$ times the optimum, where θ is the thickness of a graph, i.e., the minimum number of planar graphs into which the network can be decomposed. Krumke et al. [5] proposed efficient approximation algorithms for the distance-2 vertex coloring problem for various geometric graphs including (r, s)-civilized graphs, planar graphs, graphs with bounded genus, etc. In [6], Kumar et al. studied packet scheduling under RTS/CTS interference model and gave polylogarithmic/constant factor approximation algorithms for various families of disk graphs and randomized near-optimal approximation algorithms for general graphs.

Several distributed algorithms that use $O(\Delta)$ colors have been proposed in literatures. A $(\Delta + 1)$-coloring can be computed in time $O(\log n + \Delta)$ [30] or $O(\Delta \log n)$ [31]. In [20], Marco and Pelc proposed a distributed algorithm that computed an $O(\Delta)$-coloring in time $O(\log n)$. All of the above distributed algorithms do not take the interference into account and is based on the message passing model, which implies that the actual time used in a wireless environment could be much larger [8]. Recently, Moscibroda and Wattenhofer [8] proposed an $O(\Delta)$ distributed coloring method with time-complexity $O(\Delta \log n)$. It is worth to point out that the coloring in [8] considered a simple interference model and the time is close to time needed in practice. However, the coloring in [8] is based on the assumption that the wireless ad hoc network can be modeled as a UDG, i.e., their method will return a coloring that only guarantees that any nodes that are adjacent in the UDG will get different colors; nodes that are not adjacent in UDG may get the same color. In addition, they assumed that all nodes have the same transmission range and same interference range as its transmission range. This is different from the interference-free scheduling studied in this paper.

Kodialam and Nandagopal [2] studied the effect of interference on the achievable rate region in multihop wireless networks. They treated the interference models as linear constraints and solve the flow problem using linear program. In [3], the same authors considered the problem of jointly routing the flows and scheduling transmissions to achieve a given rate vector using the protocol model of interference. They developed necessary and sufficient conditions for the achievable rate vector. They formulated the problem as a linear programming problem and implemented primal-dual algorithms for solving the problem. The scheduling problem is solved as a graph edge-coloring problem using existing greedy algorithms. In [4], they extended their work to the multiradio multichannel wireless mesh networks.

Kumar et al. [7] developed analytical performance evaluation models and distributed algorithms for routing and scheduling which incorporate fairness, energy and dilation (path-length) requirements and provide a unified framework for utilizing the network close to its maximum throughput capacity. Alicherry et al. [1] mathematically formulated the joint channel assignment and routing problem in multiradio mesh networks, and established necessary and sufficient conditions under which interference-free link communication schedule can be obtained and designed an simple greedy algorithm to compute such a schedule. Notice that the studied network in [1] is restricted to be a UDG, i.e., the uniform interference range is assumed to be a fixed multiple of the uniform communication range.

Recently, Chen et al. [32], [33] also studied the cross-layer optimization of congestion control and routing together with scheduling problem under interference.

9 Conclusion

In this paper, we considered the problem of max-throughput (or max-fairness) routing and an interference-aware link scheduling for a wireless network. We assumed a general model for wireless networks, i.e., nodes could have different transmission ranges and different interference ranges, and a link uv may not exist even if $\|uv\|$ is less than the transmission range of node u. We presented a linear programming formulation to find a flow routing whose achieved throughput (or fairness) is at least a constant fraction of the optimum, and then used the link coloring to resolve the scheduling problem. We presented both centralized and distributed scheduling algorithms that use timeslots within a constant factor of the optimum. We also pointed out that the simple link coloring does not imply a good throughput, and then proposed efficient algorithms for general weighted link coloring, which can obtain link scheduling with proven performances. We conducted extensive simulations for our scheduling algorithms. Our theoretical results are corroborated by our simulation studies.

Challenges and future work. There are still a number of challenging questions left for future research. The first question is how to efficiently collect the information about the interfering links of a given link. This is not an issue in the previous studies since they assumed a UDG model and the same interference range for all nodes. However, when the interference range is larger than the transmission range, the information on the links within the interference area of a receiver cannot be directly collected, since these links may be outside the transmission range. Clearly, collection can only be done with helps of relaying from other nearby nodes. By assuming a fixed interference range and position information available at each node, this process can be done by collecting multihop neighborhood information. However, due to blocking or fading, fixed interference range maybe inaccurate or not practical. The second question is how to improve the overall time complexity of our distributed algorithms. The results presented in [8] may give some insights on this, but it is not obvious because the model used here is more complicated than the model used in [8]. We suspect the existence of polylogarithmic time distributed algorithms for problems studied in this paper under the unstructured environment [8]. The third question is how to solve joint routing and scheduling problem when the link capacity is not fixed. Note that here we assume that the link capacity $c(e)$ is fixed. However, it has been observed that in an interference-limited wireless network, data rates attainable in each link are not fixed and can be a function of SINR at a receiver of the link. In other words, the link capacity depends on the transmission activity of nodes around the receiver. It becomes more challenging to design efficient routing and scheduling method under such link capacity model. Recently, researchers [34], [35], [36] began to study similar joint-optimization problems under the new characteristics of the link capacity. The fourth question is to study the link scheduling in an asynchronized environment. We believe that our methods still apply with small modifications. The last but not the least problem is to study the link scheduling in a dynamic environment where the traffic load on links could change dynamically.

Acknowledgments

The work was supported in part by the US National Science Foundation (NSF) under Grants CNS-0721666 and CCR-0311174 and by the US National Aeronautics and Space Administration (NASA) under Grant ESTO 05-AIST05-0082.

References

[1] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks," *Proc. ACM MobiCom,* 2005.

[2] M. Kodialam and T. Nandagopal, "The Effect of Interference on the Capacity of Multi-Hop Wireless Networks," *Proc. IEEE Int'l Symp. Information Theory (ISIT),* 2004.

[3] M. Kodialam and T. Nandagopal, "Characterizing Achievable Rates in Multi-Hop Wireless Networks: The Joint Routing and Scheduling Problem," *Proc. ACM MobiCom,* 2003.

[4] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," *Proc. ACM MobiCom,* 2005.

[5] S. Krumke, M. Marathe, and S.S. Ravi, "Models and Approximation Algorithms for Channel Assignment in Radio Networks," *ACM Wireless Networks,* vol. 7, no. 6, pp. 575-584, 2001.

[6] A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, "End-to-End Packet-Scheduling in Wireless Ad-Hoc Networks," *Proc. 15th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA),* 2004.

[7] V.S.A. Kumar, M.V. Marathe, S. Parthasarathy, and A. Srinivasan, "Algorithmic Aspects of Capacity in Wireless Networks," *SIGMETRICS Performance Evaluation Rev.,* vol. 33, no. 1, pp. 133-144, 2005.

[8] T. Moscibroda and R. Wattenhofer, "Coloring Unstructured Radio Networks," *Proc. 17th ACM Symp. Parallelism in Algorithms and Architectures (SPAA),* 2005.

[9] S. Ramanathan, "A Unified Framework and Algorithm for Channel Assignment in Wireless Networks," *Wireless Networks,* vol. 5, no. 2, pp. 81-94, 1999.

[10] A. Ephremedis and T. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Trans. Comm.,* vol. 38, no. 4, pp. 456-460, 1990.

[11] E. Arikan, "Some Complexity Results About Packet Radio Networks," *IEEE Trans. Information Theory,* vol. 30, no. 4, pp. 190-198, 1984.

[12] S. Even, O. Goldreich, S. Moran, and P. Tong, "On the NP Completeness of Certain Network Testing Problems," *Networks,* vol. 14, no. 1, pp. 1-24, 1984.

[13] K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Proc. ACM MobiCom,* 2003.

[14] W. Wang, Y. Wang, X.-Y. Li, W.Z. Song, and O. Frieder, "Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks," *Proc. ACM MobiCom,* 2006.

[15] P. Gupta and P. Kumar, "Capacity of Wireless Networks," *IEEE Trans. Information Theory,* vol. IT-46, no. 2, pp. 388-404, 2000.

[16] J. Gronkvist and A. Hansson, "Comparison Between Graph-Based and Interference-Based STDMA Scheduling," *Proc. ACM MobiHoc,* 2001.

[17] A. Behzad and I. Rubin, "On the Performance of Graph-Based Scheduling Algorithms for Packet Radio Networks," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM),* 2003.

[18] S. Yi, Y. Pei, and S. Kalyanaraman, "On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas," *Proc. ACM MobiHoc,* 2003.

[19] D.S. Hochbaum, "Efficient Bounds for the Stable Set, Vertex Cover, and Set Packing Problems," *Discrete Applied Math.,* vol. 6, pp. 243-254, 1983.

[20] G.D. Marco and A. Pelc, "Fast Distributed Graph Coloring with $o(\Delta)$ Colors," *Proc. 12th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA),* 2001.

[21] S. Irani, "Coloring Inductive Graphs On-Line," *Algorithmica,* vol. 11, no. 1, pp. 53-72, 1994.

[22] I. Caragiannis, A.V. Fishkin, C. Kaklamanis, and E. Papaioannou, "A Tight Bound for Online Colouring of Disk Graphs," *Theoretical Computer Science,* vol. 384, nos. 2-3, pp. 152-160, 2007.

[23] *QualNet Simulator,* http://www.scalable-networks.com, 2008.

[24] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing,* Kluwer Academic Publishers, 1996.

[25] C.E. Perkins, E. Belding-Royer, and S. Das, *Ad-Hoc On-Demand Distance Vector Routing,* IETF RFC 3561, 2003.

[26] C. Berge, *Graphs and Hyper Graphs.* North-Holland, 1973.

[27] A. Panconesi and A. Srinivasan, "Improved Distributed Algorithms for Coloring and Network Decomposition Problems," *Proc. 24th Ann. ACM Symp. Theory of Computing (STOC),* 1992.

[28] S. Gandham, M. Dawande, and R. Prakash, "Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited," *Proc. IEEE INFOCOM,* 2005.

[29] U. Feige and J. Kilian, "Zero Knowledge and the Chromatic Number," *J. Computer System Science,* vol. 57, no. 2, pp. 187-199, 1998.

[30] A. Panconesi and R. Rizzi, "Some Simple Distributed Algorithms for Sparse Networks," *Distributed Computing,* vol. 14, no. 2, pp. 97-100, 2001.

[31] A. Goldberg, S. Plotkin, and G. Shannon, "Parallel Symmetry Breaking in Sparse Graphs," *Proc. 19th Ann. ACM Symp. Theory of Computing (STOC),* 1987.

[32] L. Chen, S.H. Low, and J.C. Doyle, "Joint Congestion Control and Media Access Control Design for Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2005.

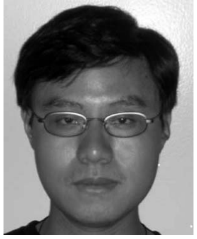
[33] L. Chen, S.H. Low, M. Chiang, and J.C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," *Proc. IEEE INFOCOM,* 2006.

[34] M.J. Neely, E. Modiano, and C.E. Rohrs, "Dynamic Power Allocation and Routing Time Varying Wireless Networks," *IEEE J. Selected Areas Comm.,* vol. 23, no. 1, pp. 89-103, 2005.

[35] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous Routing and Resource Allocation via Dual Decomposition," *IEEE Trans. Comm.,* vol. 52, no. 7, pp. 1136-1144, 2004.

[36] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint Asynchronous Congestion Control and Distributed Scheduling for Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM,* 2006.

Yu Wang received the PhD degree in computer science from the Illinois Institute of Technology in 2004 and the BEng and MEng degrees in computer science from Tsinghua University, Beijing, in 1998 and 2000, respectively. He has been an assistant professor of computer science in the Department of Computer Science, University of North Carolina, Charlotte since 2004. His current research interests include wireless networks, ad hoc and sensor networks, mobile computing, and algorithm design. He has published more than 60 papers in peer-reviewed journals and conferences. He has served as a program chair, a publicity chair, and a program committee member for several international conferences (such as IEEE IPCCC, IEEE GLOBECOM, IEEE ICC, IEEE INFOCOM, and IEEE MASS). He is the program cochair of the first ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC 2008), the program cochair of the 26th IEEE International Performance Computing and Communications Conference (IEEE IPCCC '07), the program cochair of the Fourth Workshop on Wireless Ad hoc and Sensor Networks (WWASN '07), and the program vice-chair of the First International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense '06). He is an editorial board member of the International Journal of Ad Hoc and Ubiquitous Computing and an associate editor of the International Journal of Mobile Communications, Networks, and Computing. He is a recipient of the Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities. He is a member of the ACM, the IEEE, and the IEEE Communications Society.



Weizhao Wang received the BS and MS degrees from Shanghai Jiaotong University in 1999 and 2002, respectively, and the PhD degree in computer science from the Illinois Institute of Technology in 2006. He is currently with Google Inc., Irvine, California. His current research interests include wireless networks, game theory, algorithm design, and next generation Internet.

Xiang-Yang Li received the bachelor degree in computer science and the bachelor degree in business management from Tsinghua University, Beijing, both in 1995, and the MS and PhD degrees from the University of Illinois, Urbana-Champaign, in 2000 and 2001, respectively. He has been an associate professor (since 2006) and an assistant professor (from 2000 to 2006) of computer science at the Illinois Institute of Technology. He is a visiting professor of Microsoft Research Asia for one year from May 2007. He also holds visiting professorship or adjunct-professorships at the following universities in China: TianJing University, WuHan University, and NanJing University. He is with the Department of Computer Science, Illinois Institute of Technology, Chicago, and also with Microsoft Research Asia, Beijing. He was a member of the special class (with 20 students) in China prepared for the International Mathematics Olympics (IMO) from 1988 to 1990. His research interests include wireless ad hoc networks, game theory, computational geometry, and cryptography and network security. He has published about 80 conference papers in top-quality conferences such as ACM MobiCom, ACM MobiHoc, ACM SODA, ACM STOC, and IEEE INFOCOM. He has more than 40 journal papers published or accepted for publishing. He has served various positions (such as conference chair, local arrangement chair, financial chair, session chair, and TPC member) at a number of international conferences such as AAIM, IEEE INFOCOM, ACM MobiHoc, ACM STOC, and ACM MobiCom. He is an editor of *Ad Hoc and Sensor Wireless Networks: An International Journal*. He recently also co-organized a special issue of *ACM MONET* on noncooperative computing in wireless networks and a special issue of the *IEEE Journal of Selected Area in Communications*. He is a member of the ACM and the IEEE. For more information, please see http://www.cs.iit.edu/~xli.

Wen-Zhan Song received the BS and MS degrees from the Nanjing University of Science and Technology in 1997 and 2000, respectively, and the PhD degree from Illinois Institute of Technology in 2005. He has been an assistant professor in computer science in the School of Engineering and Computer Science, Washington State University, Vancouver, since 2005. He was in the telecommunications industry from 1997 to 2001 as a system engineer. His current research interests include network protocol and algorithm design, especially in wireless networks, sensor networks, and peer-to-peer overlay networks. He is a member of the AGU, the IEEE, and the ACM.