Link Scheduling in Wireless Networks with Successive Interference Cancellation

Shaohe Lv[1], Weihua Zhuang[2], Xiaodong Wang[1], and Xingming Zhou[1]

[1]*National Laboratory of Parallel and Distributed Processing*
National University of Defense Technology, ChangSha, Hunan, 410073, P. R. China
[2]*Department of Electrical and Computer Engineering*
University of Waterloo, Waterloo, Ontario, Canada
[1]*{shaohelv, xdwang, xmzhou}@nudt.edu.cn,* [2]*wzhuang@bbcr.uwaterloo.ca*

Abstract

Successive interference cancellation (SIC) is an effective way of multipacket reception (MPR) to combat interference at the physical layer. To understand the potential MPR advantages, we study link scheduling in an ad hoc network with SIC at the physical layer. The fact that the links detected sequentially by SIC are correlated at the receiver poses key technical challenges. A link can be interfered indirectly when the detecting and removing of the correlated signals fail. We characterize the link dependence and propose a simultaneity graph (SG) to capture the effect of SIC. Then *interference number* is defined to measure the interference of a link. We show that scheduling over SG is NP-hard and the maximum interference number bounds the performance of a maximal greedy scheme. An independent set based greedy scheme is explored to efficiently construct a maximal feasible schedule. Moreover, with careful selection of link ordering, we present a scheduling scheme that improves the bound. The performance is evaluated by both simulations and measurements in a testbed. The throughput gain is on average 40% and up to 120% over IEEE 802.11. The complexity of SG is comparable with that of conflict graph, especially when the network size is not large.

Key words: Link scheduling; successive interference cancellation
PACS:

1. Introduction

Modern wireless communication is interference-limited. Due to the broadcast nature, what arrives at the receiver is a composite signal consisting of all near-by transmissions. However, the receiver tries to decode only one transmission by regarding all the others as interference and noise. When the arrivals of multiple transmissions overlap, collision occurs and the reception fails.

There are two major ways to combat the interference. The first is interference avoidance, which is at the high layers to arrange the transmission to avoid the harmful interference. This way is easy to adopt but inherently unable to provide high throughput. Many recent works show that, even with perfect network coordination, the performance is still poor [10]. The second way is to embrace the interference, i.e., all packets in a composite signal are decoded. Such capability of multiple packet reception (MPR) is a significant progress in signal processing. Recently, theoretic analysis [6] has verified the effectiveness of MPR.

Following the second approach, we further ask: with the MPR techniques, do the non-MPR upper-layer protocols work well, or how to coordinate the transmissions?

Though significant progress has been made in MPR techniques, little attention has been paid to the design of support protocols. As not all composite signals are decodable, it is indispensable to avoid harmful collisions (i.e., when the involved signals cannot be separated). In particular, there are specific requirements to ensure the feasibility of an MPR method. It is necessary to coordinate the transmissions carefully to meet the requirements.

We focus on link scheduling in an ad hoc network with successive interference cancellation (SIC) at the physical layer. SIC is a simple but powerful technique to perform MPR. With SIC, the receiver tries to detect multiple received signals using an iterative approach. In each iteration, the strongest signal is decoded, by treating the remaining signals as interference. If a required SINR (signal to interference and noise ratio) is

⋆ This work will be presented in part at the *IEEE INFOCOM* 2011 [16]. The authors would like to acknowledge the support of the National Foundation of Science of China (grant 61070203).

satisfied, this signal can be decoded and removed from the received composite signal. In the subsequent iteration, the next strongest signal is decoded, and the process continues until either all the signals are decoded or a point is reached where an iteration fails.

The fact that the reception of the desired signal relies on the successful detection of the stronger interfering signal(s) results in the dependence between the signals and introduces a new type of interference. Consider three signals, S_1, S_2 and S_3. Suppose the signal strength of S_2 is similar to that of S_3 and much greater than that of S_1. As a result, the detection of S_1 depends on that of S_2 and S_3. When there are S_3 (or S_2) and S_1 only, S_3 (or S_2) can be decoded and removed first and does not interfere the reception of S_1. How does S_3 affect the detection of S_1? In fact, when all three signals coexist, S_3 can interfere the detection of S_2, which further prevents the reception of S_1. The major feature of such interference (e.g., from S_3) is that it does not act directly on the desired signal (e.g., S_1), but on the correlated signal (e.g., S_2) and propagates to the detection of the desired signal. We term the new type of interference as *indirect* interference. Previously, e.g., in [10, 23], only direct interference was characterized. To facilitate efficient scheduling, both direct and indirect interference should be captured accurately.

To achieve the objective, we make the following contributions. First, we characterize the link dependence and propose a new network graph model, *simultaneity graph* (SG),which is built on the conflict graph (CG) model [13] with a key new component, *super vertex*, to capture the impact of SIC. A super vertex contains both a link and its correlated links(s), so that indirect interference can be characterized accurately. Second, *interference number* is defined to fully capture the effect of both direct and indirect interference. As scheduling over SG is NP-hard, we resort to an efficient heuristic solution. We show that the maximum interference number bounds the performance of a maximal greedy scheme and propose several efficient greedy scheduling schemes. In particular, an independent set based greedy scheme is proposed to efficiently construct maximal feasible schedule. Moreover, with careful selection of link ordering, we present a scheduling scheme that achieves better bound. Finally, the performance is evaluated by simulations and testbed measurements. It is shown that the newly proposed scheduling schemes perform very well, e.g., the throughput gain is on average 40% and up to 120% over IEEE 802.11. The overhead of SG is comparable to CG, especially when the network size is not large, which is a typical scenario for time-division multiple access (TDMA) scheduling.

The rest of the paper is organized as follow. Section 2 overviews the related work and Section 3 describes the system model and motivation of our work. Section 4 presents the new interference model. Section 5 discusses the scheduling schemes. Sections 6 and 7 present the results of simulation and measurement in testbed, respectively. Finally, we conclude the research in Section 8. The proofs of the claims and properties are given in Appendix.

2. RELATED WORK

It is a fundamental issue to handle transmission interference in wireless networks. Here we summarize only some of the works closely related to ours.

MPR: The earliest MPR technique is multi-user detection (MUD) [1], which attempts to extract all individual transmissions from a composite received signal. There are several different ways to perform interference cancellation (IC) [1]: parallel, successive, and hybrid. The parallel IC iteratively detects all signals at once, while the successive IC detects the signals sequentially. The hybrid IC is a combination of the two. Recently, Zigzag decoding is proposed to resolve the collision by using multiple composite signals [8]. In addition, how to combine the interference alignment and cancellation techniques to improve the network throughput is studied in [14]. Among them, successive interference cancellation (SIC) is the simplest one, whose effectiveness in an ad hoc network has been already verified experimentally [12].

Wireless network with MPR: Capacity analysis shows that MPR, or SIC, improves the performance significantly in an ad hoc network [2, 6]. To realize the potential of MPR, network protocols must be designed carefully. There are some studies to support MPR in a centralized network [26] and in a distributed scenario, e.g., distributed MAC [4] and joint routing and scheduling [24]. Recently, in [7], topology control is examined to maximize the capacity in a MIMO network with SIC.

Scheduling with MPR: Interference-aware scheduling is a classic issue [3, 5, 9, 13, 22, 23]. The famous link interference models, e.g., the protocol and physical models [10], are proposed originally for networks with single packet reception. To deal with the MPR, the protocol model is extended by growing the number of permitted interferer from zero to N ($N \geq 1$) [24], while the physical model is enhanced by allowing reception with a lower SINR (signal to interference plus noise ratio) threshold [4]. The model used in [26] correlates the reception probability with the number of concurrent transmissions, while neglecting any difference among transmissions.

Network graph is a powerful way to model the network interference. Most of the existing models (such as CG) fail to capture the impact of SIC. Though there are many scheduling schemes over CG, most of them [13, 23] are not directly applicable to the newly proposed simultaneity graph.

3. SYSTEM MODEL AND MOTIVATION

Consider a wireless network of N stationary nodes and n links. A link with transmitter S_i and receiver R_i is denoted by $L_{S_iR_i}$ (or simply, L_i). Table 1 summarizes the important notations used in this paper. We assume that: (i) the signal removal of SIC is perfect; (ii) the reception threshold, i.e., required SINR to assure the successful detection, is always larger than one at all nodes; and (iii) each node has an omni-directional antenna, operates in the half duplex mode, and is not able to transmit multiple packets simultaneously. Assumption (iii) was previously stated as primary interference, e.g., in [23].

Table 1
Summary of the main notations

Notation	Definition
$P_X(Y)$	the received signal strength at node X from node Y
β_{XY}	the reception threshold at X for the signal from Y
$LL_L(X)$	the set of links interfering link L at X
$DL_L(X)$	the set of links on which link L depends at X
SG	simultaneity graph, e.g., SG=(V, E)
IVS(LS)	the incident vertex set of link set *LS*
ILS(VS)	the incident link set of vertex set *VS*
RVS(L)	the relevant vertex set of link L
$IN^i_{SG}(L)$	the incoming number of link L in SG
$IN^o_{SG}(L)$	the outgoing number of link L in SG
$IN_{SG}(L)$	the interference number (*IN*) of link L in SG
$IN^d_{SG}(L)$	the IN difference of link L in SG
$\Delta^o(SG)$	the maximum outgoing number in SG
$\Delta^{IN}(SG)$	the maximum interference number in SG

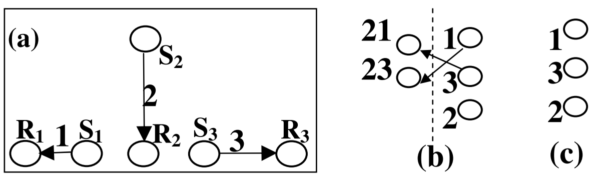

Fig. 1. (a) A three-link network; (b) The simultaneity graph; (c) The conflict graph.

Consider two links $L_{S_1R_1}$ and $L_{S_2R_2}$, at node X, and let $P_X(Y)$ be the received power at node X of the signal from node Y. Suppose $P_X(S_1) > P_X(S_2)$. Node X can decode correctly the signal of $L_{S_1R_1}$ if

$$\frac{P_X(S_1)}{P_{No} + P_X(S_2)} \geq \beta_{XS_1} \tag{1}$$

where P_{No} denote the power of noise and β_{XY} specifies the reception threshold at node X for the signal from Y.

When SIC is applied, node X is able to decode the signals of both links. First, if (1) holds, node X can decode the signal of $L_{S_1R_1}$. Second, after the decoded signal is removed, node X can decode the signal of $L_{S_2R_2}$ if the remaining signal satisfies

$$P_X(S_2)/P_{No} \geq \beta_{XS_2}. \tag{2}$$

The fact that the links detected sequentially by SIC are correlated at the receiver poses key technical challenges. Consider three links $L_{S_1R_1}$, $L_{S_2R_2}$ and $L_{S_3R_3}$, as shown in Fig. 1 (a). At node R_2, suppose $P_{R_2}(S_2) < P_{R_2}(S_1) \approx P_{R_2}(S_3)$ and the following holds:

$$\begin{aligned} \frac{P_{R_2}(S_1)}{P_{No} + P_{R_2}(S_2)} &\geq \beta_{R_2S_1} \\ \frac{P_{R_2}(S_3)}{P_{No} + P_{R_2}(S_2)} &\geq \beta_{R_2S_3}. \end{aligned} \tag{3}$$

Obviously, the detection of the $L_{S_2R_2}$ signal depends on that of $L_{S_1R_1}$. When the three links are all active, as $P_{R_2}(S_1) \approx P_{R_2}(S_3)$ and $\beta_{R_2S_1} > 1$, we have

$$\frac{P_{R_2}(S_1)}{P_{No} + P_{R_2}(S_2) + P_{R_2}(S_3)} < \beta_{R_2S_1}. \tag{4}$$

As a result, the interference from $L_{S_3R_3}$ prevents the detection of the $L_{S_1R_1}$ signal. Eventually, there is no way to obtain the desired signal (i.e., from $L_{S_2R_2}$). However, from (3), when $L_{S_3R_3}$ and $L_{S_2R_2}$ coexist, node R_2 is able to extract the $L_{S_2R_2}$ signal after removing the $L_{S_3R_3}$ signal. Therefore, the interference of $L_{S_3R_3}$ at node R_2 on $L_{S_2R_2}$ is a new type of interference, which we termed as *indirect* interference. Similarly, there is also indirect interference of $L_{S_1R_1}$ on $L_{S_2R_2}$. In comparison, there is direct interference at node R_2 between $L_{S_1R_1}$ and $L_{S_3R_3}$.

It is necessary to differentiate the two types of interference. When two links are directly interfered, they are not allowed to transmit simultaneously. In comparison, though there is indirect interference between $L_{S_2R_2}$ and $L_{S_3R_3}$, they can still transmit simultaneously. For example, the effect of $L_{S_3R_3}$ on $L_{S_2R_2}$ is harmful only when $L_{S_1R_1}$ is also active. What $L_{S_3R_3}$ interrupts is the detection of the $L_{S_1R_1}$ signal. However, as $L_{S_2R_2}$ depends on $L_{S_1R_1}$, the interference of $L_{S_3R_3}$ on $L_{S_1R_1}$ eventually prevents the detection of the $L_{S_2R_2}$ signal. To capture the indirect interference, we should characterize the link correlation.

4. MODEL THE INTERFERENCE

We model the interference at two levels. A link interference model specifies at a given node the relation among signals of different links, while a network interference model specifies whether a set of links can transmit concurrently.

4.1. *Link Interference Model*

In [10], the protocol model is defined to model the relation between two links. The model ignores the accumulated effect of multiple interfering signals and defines two possible inter-link relation: two links are *independent* if they can transmit simultaneously, or *interfered* otherwise. We based on the protocol model propose an interference model to capture the effect of SIC.

Our model: Consider, at node X, there are two links, $L_{S_1R_1}$ and $L_{S_2R_2}$. Three relations are defined between them. "L-" is used to refer to the locality (i.e., the relation is valid at node X). Suppose the signal is detectable individually, i.e., $P_X(S_1)/P_{No} \geq \beta_{XS_1}$ and $P_X(S_2)/P_{No} \geq \beta_{XS_2}$.

- $L_{S_1R_1}$ is L-independent of $L_{S_2R_2}$ if when $L_{S_2R_2}$ is transmitting, the $L_{S_1R_1}$ signal is still detectable, i.e., $P_X(S_1)/(P_{No} + P_X(S_2)) \geq \beta_{XS_1}$.
- $L_{S_1R_1}$ is L-dependent on $L_{S_2R_2}$ if the detection of the $L_{S_2R_2}$ signal is successful in the presence of the $L_{S_1R_1}$ signal, i.e., $P_X(S_2)/(P_{No} + P_X(S_1)) \geq \beta_{XS_2}$. Afterwards, the $L_{S_2R_2}$ signal is removed to enable the detection of the $L_{S_1R_1}$ signal.
- $L_{S_1R_1}$ is L-interfered by $L_{S_2R_2}$ if none of the two signals can be extracted when they transmit simultaneously.

The difference of the above model to the classic protocol model is that, when the interfering signal is sufficiently strong, the detection of the desired signal is not interfered by, but depends on the removal of the interfering signal.

4.2. *Network Interference Model*

Whether or not a set of links can concurrently transmit is determined by the local interference at all intended receivers. Let

$LL_L(X)$ denote the set of links by which link L is L-interfered at node X and $DL_L(X)$ the set of links on which link L is L-dependent at node X. Link L is *feasible* at node X only when no link in $LL_L(X)$ is active and every link in $DL_L(X)$ is either not transmitting a signal or feasible at node X. A link set LS is *feasible* only if every link is feasible at the intended receiver when all links in LS are active.

Conflict graph [13] is a widely used method to model network interference. It takes every link as vertex, and connects two vertices if the corresponding links are interfered. In a CG, it only states whether two links can transmit concurrently, but does not provide any information to distinguish the two types of interference and differentiate the simultaneous transmission due to capture effect from that due to SIC.

To model the link dependence and indirect interference, we introduce a new type of vertex, *super vertex*, which contains both the link and the correlated one(s). For example, if $L_{S_1R_1}$ is L-dependent at R_1 on $L_{S_2R_2}$, a super vertex $(L_{S_1R_1}L_{S_2R_2})$ is constructed to capture the dependence.

As the L-dependent relation may be transferable, there can exist a dependent chain, e.g., at R_1, $L_{S_1R_1}$ depends on $L_{S_2R_2}$, and $L_{S_2R_2}$ depends on $L_{S_3R_3}$, etc. Then a problem arises: How many links should a super vertex at least contain? The following claim helps us to answer the question.

Claim 1 Under the proposed link interference model, a link set consisting of more than three links is feasible, if and only if all subsets of two links and of three links are feasible.

It is feasible to set a limit that there are only *two* links in a super vertex. With Claim 1, no matter how large the network is, it is sufficient to model the relation of every subset of two or three links. When there are two links, only direct interference can exist. For indirect interference, consider a three-link set, denoted by $\{L_1, L_2, L_3\}$. Without loss of generality, suppose L_1 depends on L_2, and L_3 interferes the detection on L_2 at the receiver of L_1. To capture the indirect interference, it is sufficient to model the dependence of L_1 on L_2 and the local interference of L_3 on L_2. The super vertex to describe the dependence contains only two links (i.e., L_1 and L_2).

4.3. *Simultaneity Graph: Construction*

Graph is a common way to model wireless networks. We present a new model, *simultaneity graph* (SG). Let $SG=(V,E)$ denote a simultaneity graph, where V collects all vertices and E all edges. Table 2 summarizes the rules to construct a simultaneity graph.

There are two types of vertex in SG. An *ordinary* vertex (OV) contains a single link. For every link L, there is an OV (L) in SG. A *super* vertex (SV) contains two links that are L-dependent. SV is order-aware. For example, there are two SVs in SG, $(L_{S_1R_1}L_{S_2R_2})$ and $(L_{S_2R_2}L_{S_1R_1})$, if $L_{S_1R_1}$ is L-dependent at R_1 on $L_{S_2R_2}$ and $L_{S_2R_2}$ is L-dependent at R_2 on $L_{S_1R_1}$.

There are three types of *directed* edges in SG. First, there is an edge from $(L_{S_1R_1})$ to $(L_{S_2R_2})$ if $L_{S_1R_1}$ is L-interfered at R_1 by $L_{S_2R_2}$. The connection between two OVs captures direct interference, the same as in a conflict graph. Second, there is an

Table 2
Rules to construct a simultaneity graph

Rule	Description
Vertex Rule I	There is an OV for every link, i.e., (L) for link L.
Vertex Rule II	There is an *SV* for an L-dependent pair, i.e., $(L_{S_1R_1}L_2)$, if $L_{S_1R_1}$ is L-dependent on L_2 at node R_1.
Edge Rule I	Connect (L_2) to $(L_{S_1R_1})$ if $L_{S_1R_1}$ is L-interfered at R_1 by L_2.
Edge Rule II	Connect (L_3) to $(L_{S_1R_1}L_{S_2R_2})$ if $L_{S_2R_2}$ is L-interfered by L_3 at node R_1.
Edge Rule III	Connect $(L_{S_1R_1})$ to $(L_{S_2R_2})$ bidirectionally if $S_1=S_2$, or $S_1=R_2$, or $R_1=S_2$.

edge from (L_3) to $(L_{S_1R_1}L_{S_2R_2})$ if $L_{S_2R_2}$ is L-interfered at R_1 by L_3. The connection between OV and SV characterizes the indirect interference. Finally, to capture the primary interference, two ordinary vertices are connected bidirectionally if the transmitters are the same or the transmitter of one link is the same as the receiver of the other.

One can draw the simultaneity graph for the scenarios in Fig. 1 (a). Besides three ordinary vertices, as L_2 depends on both L_1 and L_3, two super vertices are constructed. Then two edges are established between (L_2L_1) and (L_3) and between (L_2L_3) and (L_1). The resulting graph is shown in Fig. 1 (b). For comparison, the conflict graph is presented in Fig. 1 (c), which fails to capture the indirect interference. As a result, L_2 can be starved since it is allowed that all three links transmit simultaneously.

Given $SV=(V,E)$, an adjacent relation is defined between two links or between a single link and a link set. A vertex is *adjacent* to another vertex if there is at least one edge between them. Note that the vertex can be an ordinary vertex or super vertex. Link L is adjacent to link L' when (L') and (L) are adjacent. Furthermore, link L is adjacent to link set LS if: (i) there is a link $x \in LS$ such that x is adjacent to L; or (ii) there are two links $x, y \in LS$ such that (xy) is adjacent to (L); or (iii) there are two links $x, y \in LS$ such that (xL) is adjacent to (y) or (Lx) is adjacent to (y).

4.4. *Simultaneity Graph: Properties*

The remaining question is, based on a simultaneity graph, how to quickly determine the feasibility of a link set. The claims below show that, to determine the feasibility of a link set is equivalent to determine the connectivity of a specific vertex set. For link set LS, let $IVS(LS)$ denote the *incident vertex set* of LS, which contains all ordinary vertices (L) if $L \in LS$ and all super vertices (L_1L_2) if $L_1, L_2 \in LS$. Conversely, for vertex set VS, let $ILS(VS)$ denote the *incident link set* of VS, which contains all links L if L is in at least one vertex in VS (i.e., there is a vertex such as (L), (Lx) or (xL) in VS). A vertex set VS is an *independent* set if every two vertices in VS are not adjacent, and a *complete* set if there is no vertex such that it is not in VS but only contains the link(s) in $ILS(VS)$. The following property holds immediately.

Property 1 Let LS be a link set, then $IVS(LS)$ is a complete set. For vertex set VS, $VS = IVS(ILS(VS))$ if VS is complete.

It is known that, in a CG, the fact that a link set is feasible is equivalent to that the incident vertex set is independent. In an SG, we have the following property.

Property 2 If link set LS is feasible, then $IVS(LS)$ is an independent set.

When a vertex set is independent, it is not always true that the incident link set is feasible. To determine the feasibility of a link set from a vertex set, we have the following claims.

Claim 2 (Soundness) If vertex set VS is an independent and complete set, then $ILS(VS)$ is feasible.

Claim 3 (Completeness) Link set LS is feasible if and only if $IVS(LS)$ is a complete and independent set.

Let LS be a link set, Property 1 states the uniqueness of the incident vertex set for LS. Claim 3 further indicates that, to determine the feasibility of LS is equivalent to determining whether $IVS(LS)$ is independent. From this aspect, LS is represented uniquely by $IVS(LS)$ in a simultaneity graph.

4.5. *Determine the Feasibility*

Algorithm 1: Determine whether link L can be added into a feasible link set LS

```
   Data: SG (V, E): simultaneity graph;
   LS: a feasible link set;
   L: a link not in LS
   Result: TRUE if LS∪{L} is feasible, FALSE otherwise
1  foreach vertex e in IVS(LS) do
2      if (L) is adjacent to e then
3          return FALSE;
4      end
5  end
6  Let SV_L ← V ∩ {(Lx), (xL)|x ∈ LS};
7  foreach link e in LS do
8      foreach vertex v in SV_L do
9          if (e) is adjacent to v then
10             return FALSE;
11         end
12     end
13 end
14 return TRUE;
```

The most important task in scheduling is to find a feasible link set, which can be translated to find a complete and independent vertex set. Usually, as the set of scheduled links is constructed gradually, we often need to decide whether a link (e.g., L) can be added into a feasible link set (LS). In a CG, for every $L' \in LS$, we need to check whether (L) and (L') are adjacent. In an SG, it is slightly more complex. Both direct and indirect interference should be examined carefully. The process is shown in Algorithm 1. The first *foreach* block serves to find all possible direct interference, which is the same as in the CG. Further, there are three classes of indirect interference. Namely, for $L_1, L_2 \in LS$,

- L_1 is interfered: At receiver of L_1, L_1 is L-dependent on L_2 and L_2 is L-interfered by L;
- L_1 is interfered: At receiver of L_1, L_1 is L-dependent on L and L is L-interfered by L_2;
- L is interfered: At receiver of L, L is L-dependent on L_1 and L_1 is L-interfered by L_2.

The first class is checked in the first *foreach* block, while the last two classes are examined in the second *foreach* block. The time complexity of Algorithm 1 is polynomial. To determine the feasibility of LS, we should check the connectivity between every two vertices in $IVS(LS)$, i.e., between any two OVs and between any OV to any SV. Let $|LS|$ be the number of links in LS, as there are at most O($|LS|$) OVs and O($|LS|^2$) SVs, the total number of testing (i.e., lines 2 and 9) is upper bounded by O($|LS|^3$).

5. SCHEDULING POLICY

Link scheduling has been studied extensively [11, 22, 23]. However, most of them are not SIC-aware. As scheduling over the SG is NP-hard, in the following, we present several heuristic solutions with theoretic performance bounds.

For packet transmission, time is partitioned to slots of a constant duration. Each slot is for transmission of one packet. To measure the performance of a scheduling scheme, *schedule length* is defined as the total number of time slots used by the scheme. The objective of a scheduling policy is, given an *SG* = (V, E), to assign each link the required number of time slots while assuring the schedule length is as short as possible.

5.1. *Characterizing the Interference*

Previously, in the CG, any link is contained by only one vertex and the degree of the vertex specifies the interaction of the link and the remaining part of the network. Hence, the design of many scheduling schemes over the CG is based on node degree. However, in an SG, it is possible that a link resides in more than one vertices, i.e., one ordinary vertex and several super vertices. Each vertex characterizes link interference from a unique aspect. Hence, to fully understand the interference of any given link, one should take all related vertices into account. In particular, given *SG*= (V, E), let *RVS*(L) be *relevant vertex set* of L, which denotes the set of all vertices containing link L, i.e., $RVS(L) = \{(L)\} \cup (V \cap \{(Lx), (xL)|x \in ILS(V)\})$. The interference of L is specified by all vertices in *RVS*(L) together.

The *interference number* (*IN*) estimates the amount of transmissions affected by a link. Concretely, for link L, both the links interfering and interfered by L should postpone their transmissions when link L is active. Given SG=(V, E), we define *incoming* number, $IN^i_{SG}(L)$, as the number of links interfering L, while *outgoing* number, $IN^o_{SG}(L)$, as the number of links interfered by L. Putting them together, the interference number of L is defined as $IN_{SG}(L) = IN^i_{SG}(L) + IN^o_{SG}(L)$. The incoming number of link L can be determined as follow:

- For every $(y) \in V$, if there is an edge from (y) to (L), increase $IN^i_{SG}(L)$ by one;
- For every $(Lx) \in V$ and every $(y) \in V$, if there is an edge from (y) to (Lx) but no edge from (y) to (L), increase $IN^i_{SG}(L)$ by one;
- If there are two edges connecting (y) to (Lx) and (x) to (Ly), decrease $IN^i_{SG}(L)$ by one.

The first two items estimate the number of links that interfere L directly and indirectly, respectively. Consider two links, x and y, which are L-interfered at the receiver of L. If L depends on both of them, to ensure successful detection on L, it is sufficient to postpone transmission on one of x and y. Nevertheless, the second item counts twice for such indirect interference. To compensate for this, in the last item the incoming number is decreased by one.

Similarly, $IN^o_{SG}(L)$ can be determined as follow:

– For every $(y) \in V$, if there is an edge from (L) to (y), increase $IN^o_{SG}(L)$ by one;
– For every $(xL) \in V$ and every $(y) \in V$, if there is an edge from (y) to (xL) but no edge from (y) to (x), increase $IN^o_{SG}(L)$ by one.

There is no need for the outgoing number to count for the edge from (L) to (xy). Such edge corresponds to the fact that, at the receiver of x, x depends on y and y is L-interfered by L. The interference of L has already been counted. If x is L-interfered directly by L, it is captured by the first item. Otherwise, if x is L-dependent on L, the indirect interference introduced by L is captured by the second item. When L is active, either x or y should be silent to avoid the indirect interference; yet, there is no need to postpone the transmissions on both of them. As there is at most one link delayed by L, we only increase the outgoing number by one in the second item.

It is easy to see that, for a link L, the incoming number bounds the number of links interfering L directly or indirectly. Meanwhile, the outgoing number bounds the number of links interfered directly or indirectly by L. Therefore, the interference number of L completely captures the interference of L. That is, the following property holds.

Property 3 Let $SG = (V, E)$ be a simultaneity graph, for $L \in ILS(V)$, the number of links postponed by L is upper bounded by $IN_{SG}(L)$.

Remark: The interference number of a link L can be larger than the number of links delayed by L in practice. There may be redundance between the incoming and outgoing numbers. Consider two links, L_1 and L_2, which are interfered by each other. Link L_2 is counted in both the incoming number and outgoing number of L_1. As a result, L_2 is counted twice in the interference number of L_1. By recording and comparing the links in the two categories (i.e., incoming and outgoing links), the redundance can be removed, which we leave as a future work.

Given an $SG = (V, E)$, the following property establishes the relation between the sum of incoming numbers and that of the outgoing number.

Property 4 Let $SG = (V, E)$ be a simultaneity graph, then $\sum_{L \in ILS(V)} IN^i_{SG}(L) \le \sum_{L \in ILS(V)} IN^o_{SG}(L)$.

Let $LS \subseteq ILS(V)$, the subgraph of SG generated by LS is $SG_{LS} = (V', E')$, where V' is the incident vertex set of LS in SG and E' collects all edges in E between any two vertices in V'. The interference number of L to LS is defined as the interference number of L in $SG_{LS \cup \{L\}}$.

Property 5 Let $SG_1 = (V_1, E_1)$ be a simultaneity graph and $SG_2 = (V_2, E_2)$ a subgraph of SG_1. Then, $IN_{SG_2}(L) \le IN_{SG_1}(L)$, $\forall L \in ILS(V_2)$. This also holds if replace IN by IN^i or IN^o.

Given $SG = (V, E)$, let $\Delta^{IN}(SG)$ denote the *maximum* interference number of SG, i.e., $\Delta^{IN}(SG) = \max_{x \in ILS(V)}\{IN_{SG}(x)\}$. Similarly, $\Delta^i(SG)$ and $\Delta^o(SG)$ are the maximum incoming number and outgoing number of SG, respectively. The following property is an immediate deduction of Property 5.

Property 6 Let $SG_1 = (V_1, E_1)$ be a simultaneity graph and $SG_2 = (V_2, E_2)$ a subgraph of SG_1. Then, $\Delta^o(SG_2) \le \Delta^o(SG_1)$.

SIC-aware scheduling over the SG is at least NP-hard. It is clear that, after removing all super vertices and the incident edges, the SG reduces to the CG. That is, for a network where there is no need to apply SIC, the SG is the same as the CG. Scheduling over the CG is known as NP-hard [13]. Hence, it is unlikely to design an optimal scheduling scheme over the SG within a polynomial time. Otherwise, the same method can be applied to derive an optimal solution over the CG. Therefore, we resort to a heuristic solution. Next, three scheduling schemes are presented with theoretic performance bounds.

5.2. *Independent Set based Policy*

We study one class of schemes, referred to as independent set (IS) based schemes, where the interference number is used in place of node degree. A general process of an IS-based greedy scheme is given in Algorithm 2. Let $\mathcal{S}$ be the next scheduled link set (initially empty) to be constructed, $\mathcal{U}$ be the set of unscheduled links (initially containing all the currently unscheduled links) that can be placed in $\mathcal{S}$, and $\mathcal{R}$ the set (initially empty) of unscheduled links that cannot be placed in $\mathcal{S}$. The greedy scheme proceeds as follows:

– Choose the first link $L \in \mathcal{U}$ that has the maximum interference number. Place L in $\mathcal{S}$ and move all the links $L' \in \mathcal{U}$ that are adjacent to L from $\mathcal{U}$ to $\mathcal{R}$;
– While $\mathcal{U}$ remains nonempty, do the following: choose the first link $L \in \mathcal{U}$ with respect to a specific metric, add L to $\mathcal{S}$ and move all the links that are adjacent to $\mathcal{S}$ from $\mathcal{U}$ to $\mathcal{R}$.

When $\mathcal{U}$ has been emptied, $\mathcal{S}$ is exactly a link set that will be scheduled in a time slot. For a link requiring more than one slot, the procedure repeats several times until sufficient slots are assigned to the link. It is clear that Algorithm 2 computes a feasible schedule S_{IS}.

The performance of an IS-based greedy scheme is determined by the way how the independent set (i.e., in steps 6-9) is generated. Two policies are considered in choosing the next link (step 8). The first policy, called *Smallest Degree First* (SDF), favors the link in $\mathcal{U}$ which *minimizes* the interference number to $\mathcal{U}$. The goal of the procedure is, by minimizing the number of links deleted from $\mathcal{U}$, to make current $\mathcal{S}$ as large as possible. The second policy, called *Recursive Largest First* (RLF), favors the link in $\mathcal{U}$ that has the *maximum* interference number to $\mathcal{R}$. Let G_R be the residual graph induced by the vertices except those in $IVS(\mathcal{S})$ after $\mathcal{S}$ is finally formed (i.e., when $\mathcal{U}$ is empty). The goal of the procedure is to make $\mathcal{S}$ large while assuring that G_R has as many edges eliminated from it as possible. In principle, RLF attempts to eliminate the interference at an early stage to enable more simultaneous transmissions for

Algorithm 2: SCH_{IS} - Independent set based scheduling

Data: Simultaneity Graph $SG = (V, E)$, link demands
Result: A feasible schedule S_{IS}

1 SG'←SG;
2 **repeat**
3 Set $\mathcal{S}$ empty set and $\mathcal{U}$ the set of all links in SG';
4 Choose a link in $\mathcal{U}$ that has the *maximum* interference number in SG', and move it from $\mathcal{U}$ to $\mathcal{S}$;
5 $\mathcal{R} \leftarrow \emptyset$;
6 **while** $\mathcal{U} \neq \emptyset$ **do**
7 Move all links not feasible with $\mathcal{S}$ from $\mathcal{U}$ to $\mathcal{R}$;
8 Choose a link from $\mathcal{U}$, and move it from $\mathcal{U}$ to $\mathcal{S}$;
9 **end**
10 Add $\mathcal{S}$ to the schedule S_{IS};
11 For each link $L \in \mathcal{S}$, decrease by one the demand and when the demand becomes zero, remove from SG' all vertices in $RVS(L)$ and the incident edges;
12 **until** *All links in SG' are removed*
13 return the schedule S_{IS};

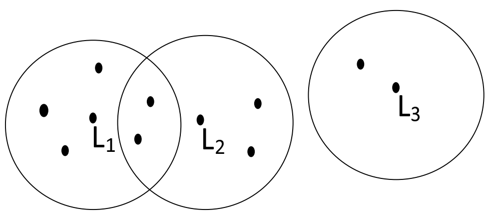

Fig. 2. Example to show the difference between RLF and SDF. The three circles show the interference ranges of links L_1, L_2 and L_3, respectively.

future. Concerning the average performance, RLF is usually the best among available graph coloring algorithms [17].

To clearly illustrate the difference of the two polices, see the example shown in Fig. 2, where each point represents a link. The three circles show the interference ranges of links L_1, L_2 and L_3, respectively (e.g., the links within the circle around L_1 interfere or are interfered by L_1). There are five, four and one links affected by L_1, L_2 and L_3, respectively. First, L_1 is chosen in step 4 of Algorithm 2. Afterwards, SDF and RLF make different decision in choosing the next link. After the links affected by L_1 are removed, as there are two links affected by L_2 and one by L_3, L_3 is preferred by SDF. In comparison, as there are two links affected by both L_1 and L_2 but none by both L_1 and L_3, L_2 is preferred by RLF.

The next claim states the worst case performance bound of a maximal greedy scheme. For simplicity, we assume that the required number of slots is at most one for every link. The case for demand larger than one is similar. A greedy scheme being *maximal* means that the set of scheduled links at any time slot except the last one is maximal, i.e., no more active link can be added. A link is *active* if it has packets to transmit (i.e., requiring at least one slot). It is easy to show that our IS-based greedy scheme is maximal.

Claim 4 Let *SG*= (*V*, *E*) be a simultaneity graph and T_X the schedule length of a greedy scheduling scheme *X*. If *X* is maximal, then $T_X \leq \Delta^{IN}(SG) + 1$.

The role of interference number in the SG is similar to that of node degree in the CG. It is well-known that the schedule length of a maximal greedy scheme over the CG is upper bounded by the maximum node degree. Claim 4 shows that the maximum interference number also bounds the schedule length for maximal greedy scheduling over the SG.

5.3. *The Third policy: Achieving Better Bound*

It is known that, for graph coloring, the optimal coloring can be obtained by using a greedy approach on a certain ordering of vertices. To improve the general bound given in Claim 4, we present a new centralized scheduling scheme with a careful selection of link ordering.

Towards this, given $SG=(V, E)$, let $IN^d_{SG}(L)$ denote the *IN difference* of link *L*, which is the difference between the outgoing and incoming *IN*, i.e., $IN^d_{SG}(L) = IN^o_{SG}(L) - IN^i_{SG}(L)$.

Property 7 Let *SG*= (*V*, *E*) be a simultaneity graph and $\Delta^d(SG)$ the *maximum IN* difference of SG, then $\Delta^d(SG) \geq 0$.

The scheduling scheme is summarized in Algorithm 3. There are two major procedures:

- Link ordering: The first link that has the *maximum IN* difference is chosen. While not all links are scheduled, do the following: remove all relevant vertices and incident edges of already chosen links, and select the link with the maximum *IN* difference in the residual SG. The process of selection provides a particular ordering of all links.
- Slot allocation: Time slots are assigned for each link from the last one to the first. When the demand of a link is larger than one slot, multiple slots are assigned to meet the demand. If currently available slots are not enough, new slots are allocated to schedule the link. Finally, at every time slot, a feasible link set is constructed.

Algorithm 3: SCH_{LO} - Scheduling with link ordering

Data: Simultaneity Graph $SG = (V, E)$, link demands
Result: A feasible schedule S_{LO}

SG'←SG;
$\mathcal{U} \leftarrow ILS(V)$;
repeat
Find a link L in $\mathcal{U}$ that has the *maximum IN* difference and let L_{n-m+1} denote the *m*th chosen link.
$\mathcal{U} \leftarrow \mathcal{U} - \{L_{n-m+1}\}$;
Remove from SG' all vertices in $RVS(L_{n-m+1})$ and the incident edges;
until $\mathcal{U} == \emptyset$
for i=1 **to** n **do**
Schedule link L_i in the first d_i available slots such that the resulting set of scheduled link is feasible, where d_i is the number of slots required by L_i;
If currently available slots are not sufficient to schedule d_i slots for L_i, add new slots at the end of the schedule S_{LO} and schedule link L_i in these slots
end
return the schedule S_{LO};

The following claim states the performance of Algorithm 3. For simplicity, we assume that the required number of slots is at most one for every link. The case for demand larger than one is similar.

Claim 5 Let *SG*= (*V*, *E*) be a simultaneity graph, the length of the schedule constructed by Algorithm 3 is upper bounded by $O(\Delta^o(SG))$.

Algorithm 3 shows that in-depth understanding of link interference can help us to achieve better scheduling performance in theory. Claim 4 bounds by $\Delta^{IN}(SG)$ the schedule length of the maximal greedy scheduling scheme, while Claim 5 shows that Algorithm 3 needs at most $2\cdot\Delta^{o}(SG)+1$ slots. The new scheme improves the bound from $\Delta^{IN}(SG)$ to $\Delta^{o}(SG)$. It is likely that $\Delta^{IN}(SG)$ is arbitrarily larger than $\Delta^{o}(SG)$. Though in the worst case the performance bound is still up to $O(n)$, the probability to have such a case can be low when the *IN* difference is exploited.

6. SIMULATION RESULTS

We evaluate the performance of the scheduling schemes by simulations in network simulator (NS-2) [19]. Each data point is obtained by averaging the results from multiple runs. Each run lasts for 5 minutes or until at least 5000 packets have been transmitted at every node.

Though there are several proposals to perform scheduling with MPR, the interference model used is not suitable for SIC. As the proposals such as those in [26] heavily depend on model parameters, it is difficult to implement them in our scenarios. Similar to [3], we compare the performance of our schemes to that of IEEE 802.11. Note that IEEE 802.11 is a CSMA MAC protocol and not originally designed for MPR. We use it as reference to show the capability of SIC and the effectiveness of the proposed interference model and scheduling schemes. In the following, the results of the three proposed scheduling schemes are denoted by "LO", "SDF" and "RLF", respectively.

The performance metric is per-link throughput. At each receiver X, both the packets destined to X and those forwarded by X are counted. Throughput gain is defined as $(T_S - T_{802})/T_{802}$, where T_S is the throughput when scheme S is used and T_{802} is that when IEEE 802.11 with carrier sensing is deployed.

6.1. *Simulation Setup*

Consider a single-channel network in a 140m×140m area. The two-way ground propagation model [19] is used. Without interference, the transmission range is 70m. A TDMA protocol is deployed at the MAC layer. The transmission rate is fixed at 11Mbps (i.e., the highest data rate in IEEE 802.11b) for all nodes if the required SINR threshold (i.e., *CPThresh*, a parameter in NS-2) can be satisfied. For how to simulate the IEEE 802.11b channel, please refer to [20]. We also performed the simulation with different settings of *fixed* transmission rate and transmission range, and found out that the results are quite similar. We leave the work for future to investigate the joint design of SIC and rate adaptation [15].

NS-2 provides support of capture effect. When collision occurs between two transmissions, the stronger one survives if the SINR is larger than CPThresh. We modify the function to support SIC and set CPThresh to the reception threshold. In particular, when collision occurs, for any involved signal, the reception succeeds only if (i) there is no L-interfered signal; (ii) the receptions of all L-dependent signals succeed; and (iii) the SINR is no less than CPThresh. Obviously, this check runs recursively.

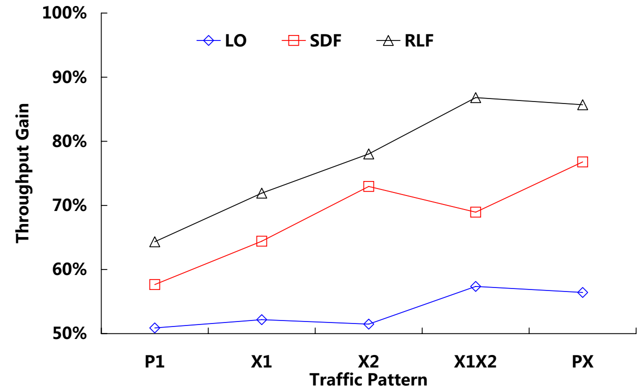

Fig. 3. Throughput gain over IEEE 802.11 under various flow patterns in a network with 49 nodes at uniform grids.

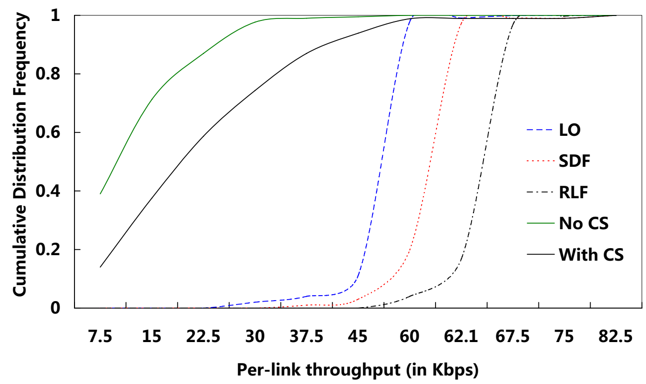

Fig. 4. Cumulative distribution of throughput under flow pattern $X1X2$ in a network with 49 nodes at uniform grids.

Unless otherwise specified, each node is backlogged, i.e., it always has packets to send, with packet size 1500 Bytes. For each source node, the destination is chosen randomly and a shortest multi-hop path is found.

6.2. *Grid Networks*

To investigate the potential advantage of SIC in a large network, consider a network with 49 nodes at uniform grids, and let (x, y) $(1 \le x \le 7,\ 1 \le y \le 7)$ denote the nodes at different positions. Three flow patterns are examined: (i) $P1$: $(x, 1) \rightarrow (x, 7)$, (ii) $X1$: $(x, 2) \rightarrow ((x+5) \bmod 7, 6)$, and (iii) $X2$: $(x, 7) \rightarrow ((x+5) \bmod 7, 2)$. Here, $v_1 \bmod v_2$ return the the remainder obtained when dividing v_1 by v_2. There are seven multi-hop flows for each pattern. Fig. 3 shows the average per-link throughput gain, where "$X1X2$" refers to that all the flows of pattern $X1$ and $X2$ are active, and "PX" to that all flows are active.

It is clearly observed that a significant gain is obtained. With a larger number of active links, there are also more opportunities of simultaneous transmissions. Along with the increase of traffic load, a higher gain is achieved. Among the three schemes, "RLF" performs the best. When the load is heavy (e.g., $X1X2$ or PX), it almost doubles the throughput as compared to IEEE 802.11 with carrier sensing.

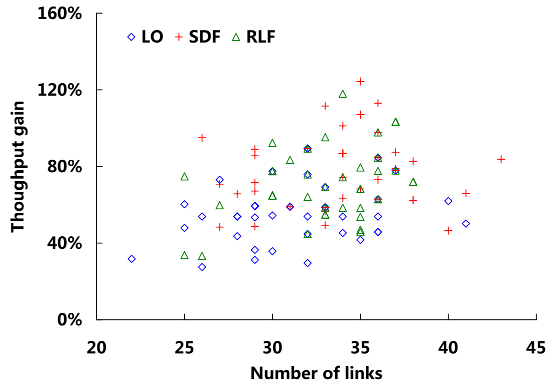

Fig. 5. Throughput gain over IEEE 802.11 verse the number of links in a network with 36 to 64 nodes.

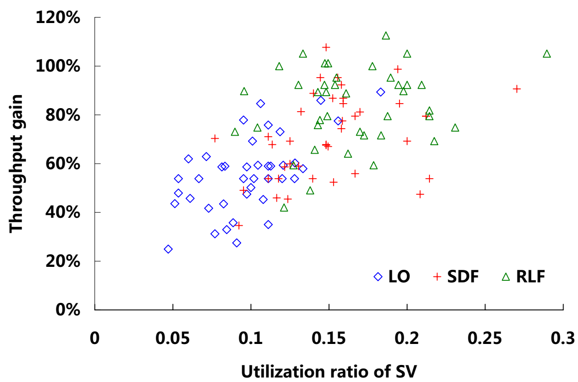

Fig. 6. Throughput gain over IEEE 802.11 verse the utilization ratio of super vertex in a network with 36 to 64 nodes.

To analyze fairness, the cumulative distribution of per-link throughput is shown in Fig. 4 for pattern $X1X2$. The results of IEEE 802.11 are also plotted. The performance of IEEE 802.11 is very poor when carrier sensing is disabled. In fact, when the number of links is large, without any coordination, the interference is very high at most receivers. It is therefore not surprising that many links receive nearly zero throughput, and no link achieves relatively high performance. In comparison, with carrier sensing, though there are still some starved links, the link number is greatly reduced. Nevertheless, per-link throughput is very low, e.g., less than 40Kbps for almost 80% links. Carrier sensing also fails to ensure the fairness, i.e., there are a very small number of links receive a very high throughput. With link scheduling, no matter which scheme is used, the throughput is the same for more than 90% links. The results verify that, with SIC in place, the proposed schemes are able to achieve both high throughput and good fairness in a relatively large network.

6.3. *Random Networks*

We also carry out simulations of networks with a dynamic topology. The number of nodes is varied between 36 to 64. All nodes are randomly positioned. For each node X, a transmission probability p_X is assigned to determine whether it issues a packet or not at the beginning of a slot. In simulations, p_X is the same for all nodes and varies from 0.5 to 0.9. Fig. 5 shows the average throughput gain verse the number of links. For "LO",

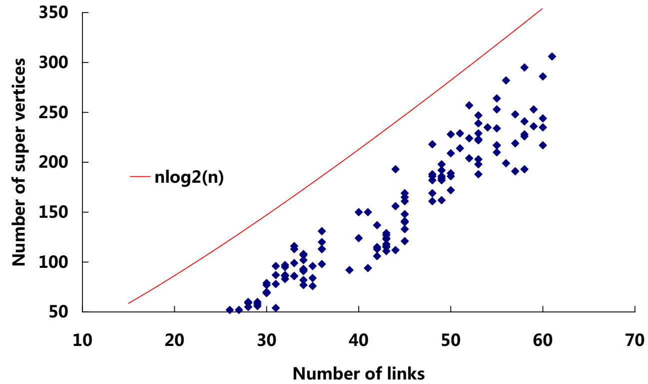

Fig. 7. The number of super vertices verse the number of links in a network with 36 to 64 nodes.

the gain is at least 30%. It is usually larger than 50% and up to 80%. The performance is even better for the two independent set based schemes. For example, when there are more than 30 links, the throughput is often double of or even higher (e.g., up to 120%) than that of IEEE 802.11 with carrier sensing. Further, for all three schemes, it is expected that the performance gain increases with the number of links.

To further understand the performance gain, *utilization ratio* of super vertex is defined as the ratio of the number of *used* super vertices to that of all super vertices. A super vertex (L_1L_2) is used if both L_1 and L_2 are scheduled in the same slot when they are active. Fig. 6 shows the throughput gain verse the utilization ratio of SV. The correlation coefficient between them is given at row "Simulation" in Table 3. The fact that all coefficient values are close to 0.5 indicates the essential correlation between the gain and the usage of super vertex. In general, with a larger utilization ratio, a higher gain can be expected. Moreover, there are much more super vertices used by "SDF" and "RLF", which explains the additional gain of them as compared to that of "LO". Though with a better theoretic bound, "LO" is not a superior choice. Recall that "LO" orders all links at once, while "SDF" and "RLF" only order the remaining links which are feasible with the set of scheduled links. It is an interesting topic to study how to effectively make use of interference information, e.g., whether or not global ordering is a good option for a greedy scheme.

To capture the effect of SIC, the super vertex is adopted to model link dependence. The complexity of scheduling relies on the number of super vertices in SG. Let n denote the number of links in the network. The number of super vertices is in fact bounded by $O(n \log_2 n)$. Note that in the worst case, the number can be up to $O(n^2)$. Fig. 7 shows the number of super vertices verse the number of links. It is clear to see that though the number increases quickly, it is always bounded by the curve of function $n \log_2 n$.

7. EXPERIMENTAL RESULTS

We now present preliminary experimental results on a software radio platform, the Universal Software Radio Peripheral (USRP) [21], built with Zigbee at the physical layer. We imple-

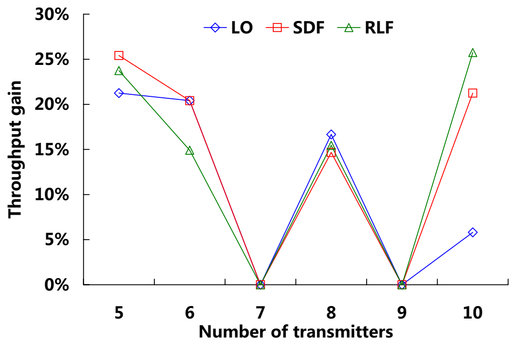

Fig. 8. Throughput gain verse the number of transmitter nodes under traffic pattern $C1$.

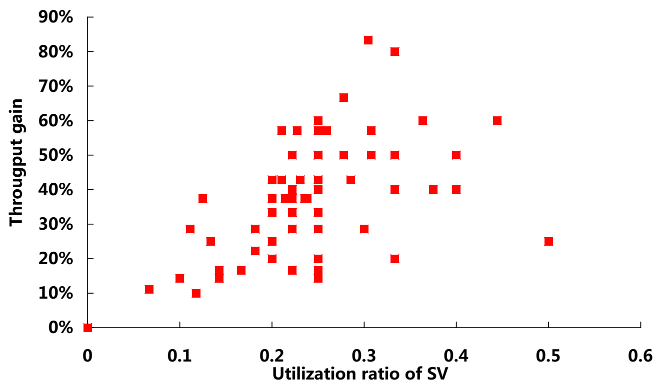

Fig. 9. Throughput gain verse the utilization ratio of super vertex for scheme "SDF" under pattern $D1$.

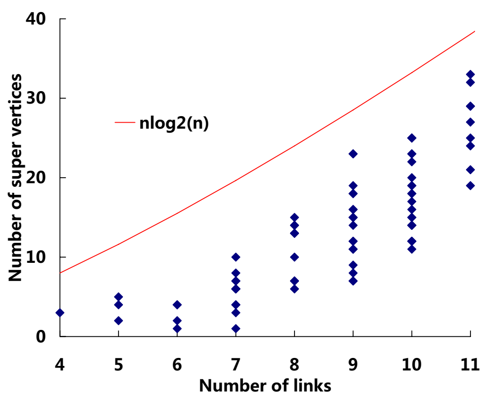

Fig. 10. The number of super vertices verse the number of links under traffic pattern $D1$.

mented a reference MAC and SIC, using a combination of standard and custom-built GNU radio processing blocks. The IEEE 802.15.4 standard provides a physical layer for sensor networks and other wireless personal area networks. At 2.4GHz, it sends up to 128byte packets at a low rate of 250kbps. The true bitrate of the ZigBee PHY is 2Mbps. Low rate 802.11 modes share many features with ZigBee including comparable data rate and the use of DSSS. For the details of implementation, please refer to [12] and the references therein.

The testbed is deployed in a large lab room with up to 11 USRP nodes randomly positioned. When there is no interference, every two nodes can communicate directly. Evidently, if all nodes do not perform SIC, all transmissions should take place in turn. Let U_i denote the ith node, where $i \in \{1, 2, \ldots, 11\}$. A transmission probability is assigned to each node when it acts as a transmitter (e.g., p_i for U_i). For all nodes, p_i is the same and varies from 0.5 to 0.9. Two different patterns are examined.

– $C1$: U_1-U_{10} send packet to U_{11}.
– $D1$: Every node is a transmitter and randomly chooses another node as the destination, which is changed randomly after every 100 packets are sent.

Table 3
Correlation coefficient between the throughput gain over IEEE 802.11 and the utilization ratio of super vertex

#	LO	SDF	RLF
Simulation	0.572	0.442	0.469
Experiment	0.612	0.616	0.553

The first pattern ($C1$) is similar to the infrastructure based WLAN. Among $U_1, U_2, \ldots, \ldots, U_{10}$, we randomly choose N ($5 \le N \le 10$) of them to be transmitters. Fig.8 plots the throughput gain verse the number of transmitters when p_i=0.9. Though slightly lower than that in simulations, a significant gain is obtained for most of the N values. In general, the gain is between 15% and 30%. Interestingly, the three schemes give similar performance. In some cases (e.g. N = 7 or 9), there is no need to apply SIC (i.e., every two signals are interfered at U_{11}). As all transmissions must be performed sequentially, there is no gain obtained from link scheduling.

Consider the second pattern ($D1$), which is similar to an ad hoc network. Due to the large number of receiver nodes, one can expect much more opportunities to apply SIC. The throughput gain verse the utilization ratio of SV for "SDF" is shown in Fig. 9. The results for "LO" and "RLF" are quite similar. Besides, the correlation coefficient for the three schemes is given at row "Experiment" in Table 3. Compared to the pattern $C1$, the advantage is much more significant. For example, the throughput gain is on average 40% and up to 80%. It is consistent with simulations that a larger gain is achieved with a higher utilization ratio of SV. For example, when more than 30% super vertices are used, the throughput gain is usually larger than 40%. To investigate the overhead of SG, Fig. 10 shows the number of super vertices verse the number of links (i.e., n). It is the same as in simulations that the number of SV is always bounded by $O(n \log_2 n)$. We have observed that, in experiments, the utilization ratio often reaches or exceeds 20%, which is much higher than in simulations (Fig. 6).

In summary, the simultaneity graph accurately characterizes the impact of SIC with reasonable overhead. TDMA scheduling is capable to capture the opportunity of simultaneous transmissions and therefore provide significant performance gain over the classic CSMA protocol (e.g., IEEE 802.11).

8. CONCLUSIONS AND FUTURE WORK

Though SIC is a simple way to perform multipacket reception, scheduling in an ad hoc network with SIC is nontrivial. The fact that the links detected sequentially by SIC are correlated at the receiver poses key technical challenges. We characterize the new link relation and propose a simultaneity graph to capture the effect of SIC. We show that scheduling over the SG is NP-hard and the maximum interference number bounds the performance of a maximal greedy scheme. Three policies are explored to efficiently construct a maximal feasible schedule. The performance is verified in both simulations using NS-2 and measurements in testbed.

In this work, the assumptions made in Section 3 help to keep the problem complexity at a reasonable level. However, in practice, the signal removal is imperfect [25]. Moreover, in a CDMA-based network, the reception threshold may be less than one [18]. As a result, the link relation can become much more complicated. For future studies, it is interesting to investigate the design of a scheduling scheme when these assumptions are removed. In addition, to combat the effect of both interference and attenuation, it is necessary to integrate interference cancellation and rate adaptation. Finally, an efficient distributed scheme to achieve good scheduling performance requires further investigation.

Appendix A. Proofs of the Claims and Properties

Proof of Claim 1: The necessary condition holds intuitively. We proof the sufficient condition by contradiction. Let LS denote the whole link set. We need to show that, when LS is not feasible, there must be at least an infeasible subset of two or three links. If LS is not feasible, at least one link $L_{SR} \in LS$ is not feasible. Then we have one (or both) of following:

– There is a link $L_{S'R'} \in LS$ and $L_{S'R'} \in LL_{L_{SR}}(R)$;
– There is a link $L_{S_1R_1} \in LS$ and $L_{S_1R_1} \in DL_{L_{SR}}(R)$; yet node R cannot decode the $L_{S_1R_1}$ signal.

For the first case, $\{L_{SR}, L_{S'R'}\}$ is not feasible. For the second case, similarly, we have:

– There is a link $L_{S'_1R'_1} \in LS$ and $L_{S'_1R'_1} \in LL_{L_{S_1R_1}}(R)$;
– There is a link $L_{S_2R_2} \in LS$ and $L_{S_2R_2} \in DL_{L_{S_1R_1}}(R)$; yet node R cannot decode the $L_{S_2R_2}$ signal.

For the first case, $\{L_{SR}, L_{S_1R_1}, L_{S'_1R'_1}\}$ is not feasible. The indirect interference of $L_{S'_1R'_1}$ prevents the detection on L_{SR}. For the second case, similarly, we have an $L_{S'_2R'_2}$ or $L_{S_3R_3}$. The chain is extended until an $L_{S'_eR'_e}$ is encountered for $L_{S_eR_e}(e \geq 1)$. As the number of links is finite, such e must exist.

As the reception threshold is larger than one, it is easy to see that $P_R(S) \leq P_R(S_1) \ldots \leq P_R(S_e)$. Therefore, L_{SR} must not be L-independent of $L_{S_kR_k}(k \leq e)$. If L_{SR} is L-interfered by $L_{S_eR_e}$, $\{L_{SR}, L_{S_eR_e}\}$ is not feasible. Otherwise, L_{SR} is L-dependent on $L_{S_eR_e}$, then $\{L_{SR}, L_{S_eR_e}, L_{S'_eR'_e}\}$ is not feasible because the indirect interference of $L_{S'_eR'_e}$ prevents the detection on L_{SR}.

In summary, when the link set is not feasible, there must be a two-link or three-link subset that is not feasible. ■

Proof of Property 2: We proof by contradiction, i.e., if $IVS(LS)$ is not an independent set, LS must be infeasible. When $IVS(LS)$ is not an independent set, $IVS(LS)$ must contain:

– Two adjacent OVs, e.g., (L'_1) and (L'_2); or
– an OV (L_1) and an SV (L_2L_3) which are adjacent.

Then $\{L'_1, L'_2\}$ or $\{L_1, L_2, L_3\}$ is not feasible. Observe that $\{L'_1, L'_2\} \subseteq LS$ and $\{L_1, L_2, L_3\} \subseteq LS$, according to Claim 1, we can therefore conclude that LS is not feasible. ■

Proof of Claim 2: We proof by contradiction, i.e., if $ILS(VS)$ is infeasible, VS is incomplete or not an independent set. According to Claim 1, there is a subset (e.g., Sub_L) of $ILS(VS)$ that is not feasible and contains two or three links. When $Sub_L = \{L'_1, L'_2\}$, (L'_1) and (L'_2) must be adjacent. Otherwise, Sub_L becomes feasible. When $Sub_L=\{L_1, L_2, L_3\}$, let L_1 be the failed link. In the presence of indirect interference, L_1 should be L-dependent on a link (e.g., L_2), which is L-interfered by the third link (e.g., L_3) at the receiver of L_1. As a result, (L_1L_2) is constructed by the vertex rule II, which is adjacent to (L_3) according to the edge rule II. If VS is incomplete, we complete the proof. Otherwise, as $\{L'_1, L'_2\} \subseteq LS$ and $\{L_1, L_2, L_3\} \subseteq ILS(VS)$, we have $\{(L'_1), (L'_2)\} \subseteq VS$ or $\{(L_1L_2), (L_3)\} \subseteq VS$, implying that VS is not an independent set. ■

Proof of Claim 3: The sufficient condition is due to Property 2 and Claim 2, while the necessary one is from Property 3. ■

Proof of Property 4: For any link L, the incoming number counts the links interfering L. There are two types of interference: direct or indirect. For direct interference, i.e., the edge from (y) to (L), it is captured in $IN^o_{SG}(y)$. For indirect interference, i.e., the edge from (y) to (Lx), it is captured in $IN^o_{SG}(x)$. Therefore, the sum of the outgoing number is no less than that of the incoming number. ■

Proof of Property 5: For a link L, the incoming number counts the number of links connecting to (L) or an SV such as (Lx). When there is a vertex (y) in E_2 and it connects to $v \in \{(\mathrm{L})\} \cup \{(Lx)|(Lx) \in E_2\}$, as SG_2 is a subgraph of SG_1, we have the following: both (y) and v must be in E_1 and they are connected in SG_1. Therefore, $IN^i_{SG_2}(L) \leq IN^i_{SG_1}(L)$. For the outgoing number, similarly, we have $IN^o_{SG_2}(L) \leq IN^o_{SG_1}(L)$. According to the definition of interference number, we have $IN_{SG_2}(L) \leq IN_{SG_1}(L)$. ■

Proof of Property 7: Let $L_1, \ldots, L_n$ denote all the links in $ILS(V)$.

$$\begin{aligned}
\Delta^d(SG) &= \max_{1\leq t\leq n}\{IN^o_{SG}(L_t) - IN^i_{SG}(L_t)\} \\
&\geq \frac{1}{n}\sum_{1\leq t\leq n}(IN^o_{SG}(L_t) - IN^i_{SG}(L_t)) \\
&= \frac{1}{n}(\sum_{1\leq t\leq n} IN^o_{SG}(L_t) - \sum_{1\leq t\leq n} IN^i_{SG}(L_t)) \\
&\geq 0.
\end{aligned} \tag{A.1}$$

The last inequality holds due to Property 4. ■

Proof of Claim 4: We proof by contradiction, i.e., if $T_X > \Delta^{IN}(SG) + 1$, then the greedy scheme X is not maximal. Let $LS_1, \ldots, LS_{T_X}$ denote the schedule constructed by the scheme X. $\forall L \in LS_{T_X}$, we have $IN_{SG}(L) \leq \Delta^{IN}(SG) < T_X$-1. From the Pigeon hole principle, among $LS_1, \ldots, LS_{T_X-1}$, there must be at least one link set which does not contain any link postponed by L. Let LS_i denote the link set. It is safe to add L into LS_i. As L is active at time slot i (i.e., when LS_i is scheduled), the greedy scheme X is not maximal. ■

Proof of Claim 5: Let n denote the total number of links and $L_1, \ldots, L_n$ the ordered links. Scheduling (i.e., steps 8-11) starts from L_n to L_1. After L_t is scheduled, let TS_t denote the schedule length and $SG_t = (V_t, E_t)$ the simultaneity graph generated by the set of all scheduled links. It is obvious that $TS_1 \geq TS_2 \geq \ldots TS_n$ and TS_1 equals to the length of the schedule constructed by Algorithm 3.

Observe that in SG_t, the *IN* difference of L_t is the maximum one. According to Property 7, $\Delta^d_{SG_t}(L_t) \geq 0$. That is, $IN^o_{SG_t}(L_t) \geq IN^i_{SG_t}(L_t)$. Thus, $IN_{SG_t}(L_t) = IN^i_{SG_t}(L_t) + IN^o_{SG_t}(L_t) \leq 2 \cdot IN^o_{SG_t}(L_t)$.

When we attempt to assign a time slot to L_t, there are two possible cases:

- $TS_{t+1} > IN_{SG_t}(L_t)$: Observe that the number of links postponed by L_t is upper bounded by $IN_{SG_t}(L_t)$. From the Pigeon hole principle, there must be at least one time slot, for which the set of scheduled links does not contain any link postponed by L_t. It is safe to assign such slot to L_t. As a result, the schedule length is unchanged after L_t is scheduled, i.e., $TS_t = TS_{t+1}$.
- $TS_{t+1} \leq IN_{SG_t}(L_t)$: In the worst case, one new time slot is allocated to L_t, implying $TS_t \leq TS_{t+1} + 1 \leq IN_{SG_t}(L_t) + 1 \leq 2 \cdot IN^o_{SG_t}(L_t) + 1$.

In summary, $TS_t \leq \max\{TS_{t+1}, 2 \cdot IN^o_{SG_t}(L_t) + 1\}$. Observe that $SG = SG_1$ and SG_{t+1} is a subgraph of SG_t, we show $TS_t \leq 2 \cdot \Delta^o(SG_t) + 1$ via two steps.

(i) Evidently, $TS_n = 1 \leq 2 \cdot \Delta^o(SG_n) + 1$.

(ii) Suppose $TS_{k+1} \leq 2 \cdot \Delta^o(SG_{k+1}) + 1$ for $k \leq n$, then $TS_k \leq 2 \cdot \Delta^o(SG_k) + 1$ must hold. In fact,

$$\begin{aligned} TS_k &\leq \max\{TS_{k+1}, 2 \cdot IN^o_{SG_k}(L_k) + 1\} \\ &\leq \max\{2 \cdot \Delta^o(SG_{k+1}) + 1, 2 \cdot IN^o_{SG_k}(L_k) + 1\} \\ &\leq \max\{2 \cdot \Delta^o(SG_k) + 1, 2 \cdot IN^o_{SG_k}(L_k) + 1\} \\ &= 2 \cdot \Delta^o(SG_k) + 1. \end{aligned} \tag{A.2}$$

The third inequality holds due to Property 6.

Put the results in the two steps together, we conclude that $TS_t \leq 2 \cdot \Delta^o(SG_t) + 1$. Then $TS_1 \leq 2 \cdot \Delta^o(SG_1) + 1 = 2 \cdot \Delta^o(SG) + 1$. Therefore, the length of the schedule constructed by Algorithm 3 is upper bounded by $O(\Delta^o(SG))$. ■

References

[1] J. Andrews. Interference cancellation for cellular systems: a contemporary overview. *IEEE Wireless Communications*, 12(2):19–29, 2005.

[2] J. Blomer and N. Jindal. Transmission capacity of wireless ad hoc networks: Successive interference cancellation vs. joint detection. In *Proc. IEEE ICC'09*, pages 1–5, 2009.

[3] G. S. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proc. ACM MOBICOM'06*, pages 2–13, 2006.

[4] G. D. Celik, G. Zussman, W. F. Khan, and E. Modiano. MAC for networks with multipacket reception capability and spatially distributed nodes. In *Proc. IEEE INFOCOM'08*, pages 1436–1444, 2008.

[5] M. Dinitz. Distributed algorithms for approximating wireless network capacity. In *Proc. IEEE INFOCOM'10*, pages 1397–1405, 2010.

[6] J. J. Garcia-Luna-Aceves, H. R. Sadjadpour, and Z. Wang. Challenges: towards truly scalable ad hoc networks. In *Proc. ACM MOBICOM'07*, pages 207–214, 2007.

[7] E. Gelal, K. Pelechrinis, T.-S. Kim, I. Broustis, S. V. Krishnamurthy, and B. Rao. Topology control for effective interference cancellation in multi-user MIMO networks. In *Proc. IEEE INFOCOM'10*, pages 2357–2365, 2010.

[8] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *Proc. ACM SIGCOMM'08*, pages 159–170, 2008.

[9] O. Goussevskaia and R. Wattenhofer. Complexity of scheduling with analog network coding. In *Proc. ACM FOWANC'08*, pages 77–84, 2008.

[10] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. Information Theory*, 46(2):388–404, 2000.

[11] R. Gupta, J. Musacchio, and J. Walrand. Sufficient rate constraints for QoS flows in ad-hoc networks. *Ad Hoc Networks*, 5(4):429–443, 2007.

[12] D. Halperin, T. E. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *Proc. ACM MOBICOM'08*, pages 339–350, 2008.

[13] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. ACM MOBICOM'03*, pages 66–80, 2003.

[14] E. L. Li, R. Alimi, D. Shen, H. Viswanathan, and Y. R. Yang. A general algorithm for interference alignment and cancellation in wireless networks. In *Proc. IEEE INFOCOM'10*, pages 1774–1782, 2010.

[15] S. Lv, X. Wang, and X. Zhou. On the rate adaptation for IEEE 802.11 wireless networks. *Computer Networks*, 54(17):3173–3186, 2010.

[16] S. Lv, W. Zhuang, X. Wang, and X. Zhou. Scheduling in wireless ad hoc networks with successive interference cancellation. In *Proc. IEEE INFOCOM'11*, to appear.

[17] E. Malaguti and P. Toth. A survey on vertex coloring problems. *International Transactions in Operational Research*, 17(1):1–34, 2009.

[18] G. D. Nguyen, A. Ephremides, and J. E. Wieselthier. Comments on "capture and retransmission control in mobile radio". *IEEE Journal on Selected Areas in Communications*, 24(12):2340–2341, 2006.

[19] NS-2. Network simulator version 2. *http://www.isi.edu/nsnam/ns/*, 2010.

[20] J. Robinson. Making ns-2 simulate an 802.11b link. *http://www.joshuarobinson.net*, 2011.

[21] USRP. Ettus research. *http://www.ettus.com*, 2010.

[22] P.-J. Wan, Z. Wang, H. Du, S. C.-H. Huang, and Z. Wan. First-fit scheduling for beaconing in multihop wireless networks. In *Proc. IEEE INFOCOM'10*, pages 2205–2212, 2010.

[23] W. Wang, X.-Y. Li, O. Frieder, Y. Wang, and W.-Z. Song. Efficient interference-aware TDMA link scheduling for static wireless networks. In *Proc. ACM MOBICOM'06*, pages 262–273, 2006.

[24] X. Wang and J. J. Garcia-Luna-Aceves. Embracing interference in ad hoc networks using joint routing and scheduling with multiple packet reception. In *Proc. IEEE INFOCOM'08*, pages 843–851, 2008.

[25] S. P. Weber, J. G. Andrews, X. Yang, and G. de Veciana. Transmission capacity of wireless ad hoc networks with successive interference cancellation. *IEEE Trans. Information Theory*, 53(8):2799–2814, 2007.

[26] Q. Zhao and L. Tong. A dynamic queue protocol for multiaccess wireless networks with multipacket reception. *IEEE Trans. Wireless Communications*, 3(6):2221–2231, 2004.