Improved Measurement-Based Frequency Allocation Algorithms for Wireless Networks

Jeremy K. Chen, Gustavo de Veciana, and Theodore S. Rappaport
Wireless Networking and Communications Group (WNCG), The University of Texas at Austin, USA
Emails: {jchen, gustavo, wireless}@ece.utexas.edu

***Abstract*— This paper presents three algorithms that outperform all other published work for allocating a limited number of orthogonal frequency channels to access points (APs) in wireless networks. Unlike other work, we minimize interference seen by both *users* and *APs*, we use a *physical* rather than *binary* model for interference, and we mitigate the impact of rogue RF interference. Our three algorithms have different mechanisms of switching the channels of APs based on the in-situ interference measured at clients and/or APs. The convergence of the algorithms is proven and characterized. Our algorithms consistently yield high throughput gains irrespective of network topology, the level of AP activity, and the number of controlled APs, rogue interferers, and available channels. We outperform the best published work by 10% and 9.3% for mean and median user throughputs respectively, and 28%, 55%, 160%, and 7690% for 25, 20, 15, and 10 percentiles of user throughputs, respectively.**

I. Introduction

In wireless LANs (WLANs), a number of orthogonal frequency channels are allocated to APs so that each AP is allocated one channel. When the number of channels is limited relative to the number of APs, some APs inevitably use the same channel and induce co-channel interference. Judicious channel reuse mechanisms are necessary to reduce such interference. The same problem exists in cellular networks.

A number of WLAN frequency allocation schemes have been proposed thus far. The work in [1] assumes each AP has a different fixed traffic load, and defines the effective channel utilization of an AP as the fraction of time the channel is used for data transmission or is sensed busy due to interference from other APs; then, the maximum effective channel utilization among all APs is minimized. AP placement and frequency allocation are jointly optimized in [2] with the same objective of minimizing the max channel utilization as in [1]. The frequency allocation problem is modeled as a minimum-sum-weight vertex-coloring problem in [3] where vertices are APs, and the weight of each edge between two APs denotes the number of clients that are associated with either one of these two APs and are interfered by the other AP. The work in [4] minimizes the number of clients whose transmissions suffer channel conflicts; a client associated with an AP suffers conflicts if other clients or other APs interfere with the client or the AP under consideration. The definition of channel conflict in [4] is more comprehensive than those in [1]–[3]; the work in [4] has been shown to outperform [1]–[3].

Only [5] presents mechanisms to detect and reduce the negative impact from *rogue interferers*, i.e., intentional or unintentional RF interferers, microwave ovens, or other RF devices that also operate on the same unlicensed bands as WLAN. In [5], each AP senses interference and independently selects a channel whose measured interference power is below a predefined threshold, without coordinating with other APs. In networks with high interference, it may not be feasible to find a channel allocation so that every AP senses interference below the threshold; in this case, the algorithm in [5] does not converge. One could in principle set a higher threshold for the algorithm in [5] to work in high-interference regimes, but [5] does not mention methods to adapt the threshold. It is not trivial to adapt this threshold, since a high threshold will degrade network performance, but a low threshold will yield no feasible solutions. By contrast, two of our proposed algorithms converge irrespective of the overall interference level. The non-convergence result of [5] in the high interference regime is due to the *binary model for interference*, which is also used in [1]–[4]. Our work considers a *physical* model for interference; that is, we assume that interference power is a continuous quantity, which properly represents the real world.

Most traffic in WLANs is downlink [6]; hence, maximizing *downlink* throughputs and signal-to-interference-and-noise ratio (SINR) seen by *users* are key to proper network design. The work in [1], [2], [5] focuses on minimizing the interference at *APs* rather than that seen by *users*, as is done in [3], [4], and thus often perform poorer than [3], [4].

The *main contribution* of this work is our three new algorithms that outperform all other published work, i.e., those in [1]–[5]. The proposed algorithms perform well mainly because they: (1) minimize interference seen by *users* rather than that seen by *APs*; (2) use a *physical* model rather than a *binary* model for interference; and (3) have the ability to deal with *rogue interferers*. We propose that all or a subset of clients measure the *in-situ* interference power on all frequency channels periodically when their associated APs are idle, and report the average measured power to their associated APs. This technique is used in mobile-assisted hand-off (MAHO) in the cellular field [7], and results in this paper may also be applied to cellular networks. APs also measure *in-situ* interference power. Since the measurements at APs or clients are performed during their idle time, the overhead is negligible. Each AP then computes a metric called *weighted interference* which captures the overall interference

[1]This research is sponsored by NSF Grant ACI-0305644.

as seen by itself and its clients, by placing different weights on its and the clients' in-situ measurements according to the clients' traffic loads, signal strengths, and uplink and downlink traffic volume. Section II introduces the system model and notation, and describes the *weighted interference* in detail. The three proposed algorithms, denoted *No-Coord*, *Local-Coord*, and *Global-Coord*, have different mechanisms for iteratively switching frequency channels in order to reduce the weighted interference seen in a single cell, a group of nearby cells, or all cells, respectively, where a *cell* means an AP (or base station) and its associated users. Section III presents the mechanisms used by the three algorithms and their convergence. Section IV shows by simulation that our algorithms substantially outperform [1]–[5].

II. System Model and Notation

We first describe basic notation; then the first subsection describes *weighted interference*, a metric used in the three proposed algorithms to capture the overall interference of each cell. The second subsection defines notation used exclusively for the proposed *Local-Coord* algorithm.

Suppose M APs, indexed by $\mathbb{M} = \{1, 2, \ldots, M\}$, operate on K orthogonal frequency channels, indexed by $\mathbb{K} = \{1, 2, \ldots, K\}$. We index users (or clients) by $\mathbb{L} = \{1, 2, \ldots, L\}$. We denote the identity of an AP and a client by a_m ($m \in \mathbb{M}$) and c_l ($l \in \mathbb{L}$), respectively. We assume for this work that the locations of the APs and the clients do not vary with time, and assume that no APs or users are at the same location, although the algorithms given here also apply for mobile APs and/or clients. Let $\mathbb{L}_m$ ($\mathbb{L}_m \subseteq \mathbb{L}$) denote the set of users that are associated with the AP a_m. We assume every user is associated with a single AP, and define a *cell* $\mathbb{Z}_m = \{a_m\} \cup \{c_l : l \in \mathbb{L}_m\}$. Let f_m ($f_m \in \mathbb{K}$) denote the channel that a_m operates on, and let $\vec{f} = (f_1, f_2, \ldots, f_M)$ denote the channels of all M APs.

A. Weighted Interference

In brief, the weighted interference of each cell (say $\mathbb{Z}_m$) is intended to capture the overall interference in the cell, and is therefore defined as a weighted sum of the average in-situ measurements at a_m and at all clients associated with a_m, i.e., at every $u \in \mathbb{Z}_m$. We propose that a_m or the clients associated with a_m measure their in-situ interference power when there is no traffic within Cell $\mathbb{Z}_m$, i.e., a_m is neither transmitting or receiving data. The average in-situ measured interference power at u (for every $u \in \mathbb{Z}_m$) on channel k is denoted $I_k^u(\vec{f})$. The averaging period is a design choice and could be the same as the period that an AP switches its channel, say 1, 2, or 5 minutes. $I_k^u(\vec{f})$ is lower-bounded by noise floor. The *weighted interference function* of $\mathbb{Z}_m$ on channel k is defined by

$$W_k^m(\vec{f}) = \sum_{u \in \mathbb{Z}_m} B_k^u(I_k^u(\vec{f})), \quad k \in \mathbb{K}, \tag{1}$$

where $B_k^u(\cdot)$ is a nonnegative and non-decreasing function that captures the weight of the in-situ measurement at u. We require that $W_k^m(\vec{f}) > 0$ to capture the noise floor in the real world. $B_k^u(\cdot)$ should be designed to reflect the difference of clients' traffic demands, signal strengths, and uplink and downlink traffic volume. In practice, clients report the measurements to a_m either periodically or upon request from a_m; then, $W_k^m(\vec{f})$ can be computed at a_m.

In Section III-E we show that two of our proposed algorithms (namely *Local-Coord* and *Global-Coord*) converge if the weighted interference function has the general form in (1). Below we introduce two simplified forms of $W_k^m(\cdot)$ representing practical metrics. The first form, denoted *user-based*, places different weights on the in-situ interference measurements at clients based on the traffic volume and the signal strength at each client. The *user-based* form captures the performance of downlink transmission, which is appropriate for WLANs since traffic measurements show that downlink traffic volume accounts for more than 84% of total (uplink plus downlink) traffic volume [6]. The second form, denoted *AP-based*, includes the interference measurements at APs only. The *AP-based* form can be viewed as a simplified version of the *user-based* one by considering all users have the same traffic volume and signal strength.

1) User-based: The *user-based* weighted interference function for $\mathbb{Z}_m$ is defined by

$$W_k^{(\mathrm{C}),m}(\vec{f}) = \sum_{l \in \mathbb{L}_m} \frac{Y_{c_l,a_m}}{S_{c_l,a_m}} \cdot I_k^{c_l}(\vec{f}), \tag{2}$$

where S_{c_l,a_m} denotes the average received signal power[2] from a_m to c_l, and Y_{c_l,a_m} denotes the average traffic volume from a_m to c_l. We incorporate the inverse of S_{c_l,a_m} in (2) because a client with a stronger S_{c_l,a_m} has higher tolerance to interference and thus should contribute less to the overall weighted interference. Y_{c_l,a_m} is included in (2) as a scaling factor, since a client with higher traffic volume should be more important for the weighted interference. In practice, some users may be sampled to reduce the complexity of computing (2), i.e., the summation in (2) may be over a subset of $\mathbb{L}_m$.

2) AP-based: The *AP-based* weighted interference function for a_m is defined by

$$W_k^{(\mathrm{A}),m}(\vec{f}) = I_k^{a_m}(\vec{f}). \tag{3}$$

B. Interfering Cells

When an AP switches its channel, some nearby cells see changes in their weighted interference. The *Local-Coord* algorithm examines the cells that see such changes; the notation of such cells are presented below. Cell $\mathbb{Z}_n$ is said to be interfered by Cell $\mathbb{Z}_m$ (or $\mathbb{Z}_m$ interferes with $\mathbb{Z}_n$) if and only if a_m or a user associated with a_m induces non-negligible interference (e.g., the interference power at the receiver is higher than the noise floor) at a_n or a user associated with a_n. We define $\mathbb{G}_m$ (*the set of cells interfered by* $\mathbb{Z}_m$) such that $n \in \mathbb{G}_m$

[2]Note c_l cannot measure S_{c_l,a_m} directly but can estimate S_{c_l,a_m} as follows. The average in-situ SINR at c_l can be measured at c_l when a_m is transmitting to c_l, and is denoted γ_l. We assume the interference at c_l is the same whether a_m is transmitting to c_l or a_m is idle, i.e., the interference at c_l is always $I_{f_m}^{c_l}(\vec{f})$. Then we estimate $S_{c_l,a_m} = \gamma_l \cdot I_{f_m}^{c_l}(\vec{f})$.

if and only if $\mathbb{Z}_m$ interferes with $\mathbb{Z}_n$ given that a_m and a_n are on the same channel. The subset of $\mathbb{G}_m$ that is on channel k is denoted $\mathbb{G}_{m,k}(\vec{f})$. Suppose a_m switches from channel k to k', the cells that see changes in their weighted interference are $\mathbb{Z}_m$ and the cells indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$; hence the weighted interference of the cells indexed by $\mathbb{H}_{m,k,k'}(\vec{f}) \equiv \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})$ are examined by *Local-Coord* if a_m switches from channel k to k'.

Another AP a_n can run *Local-Coord* simultaneously with a_m if the channel switching of a_n induces negligible change of the weighted interference of the cells that may be examined by *Local-Coord*, i.e. $\mathbb{Z}_m$ and the cells indexed by $\mathbb{G}_m$. We define $\mathbb{V}_m$ as the set of the indices of cells that interfere with $\mathbb{Z}_m$ or the cells indexed by $\mathbb{G}_m$, i.e. $i \in \mathbb{V}_m$ if and only if there exists $j \in \{m\} \cup \mathbb{G}_m$ such that $\mathbb{Z}_i$ interferes with $\mathbb{Z}_j$. The cells indexed by $\mathbb{V}_m$ include all the cells that cannot simultaneously change channels with a_m. The notation of $\mathbb{V}_m$ is used for the distributed protocol of *Local-Coord*. Suppose we are given the locations of all controlled APs and possible locations of clients; then the sets of $\mathbb{G}_m$ and $\mathbb{V}_m$ can be pre-computed and pre-configured in the controlled APs or a central network controller that communicates with the controlled APs, using radio propagation prediction models as described in [7]–[9].

III. Three Measurement-Based Algorithms

The three proposed algorithms all have an iterative nature. At each point in time (predefined, randomly chosen, or determined at runtime), say every 1, 2, or 5 minutes, one *iteration* of channel switching takes place where one or more APs switch their frequency channels according to mechanisms that are specific to the proposed algorithms, while other APs stay on their current channels. The channel switching time in hardware is several milliseconds and is thus negligible as compared to the interval between two iterations. APs and clients measure and average their in-situ interference between every two successive iterations. Iterations keep taking place on different AP(s) until the channel allocations converge. Below we describe the different conditions of the three algorithms that a representative AP a_m can switch from channel $k = f_m$ to $k' = f'_m$. Throughout this paper, $\vec{f'} \in \mathbb{K}^M$ denotes a vector of channels selected by APs after the representative AP a_m moves from channel f_m to f'_m. Hence $\vec{f'}$ differs from $\vec{f}$ in only the m-th element.

A. The ***No-Coord*** Algorithm

A representative AP a_m switches from its current channel f_m to f'_m only if the weighted interference on the new channel f'_m is lower, i.e., the following condition holds:

$$\textit{No-Coord}\text{ Condition:} \qquad W^m_{f_m}(\vec{f}) > W^m_{f'_m}(\vec{f'}). \qquad (4)$$

This algorithm is denoted *No-Coord*, because a_m makes a greedy channel selection without coordination with other APs.

B. The ***Local-Coord*** Algorithm

If a_m switches from channel k to k', only $\mathbb{Z}_m$ and the cells indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$ see changes in their weighted interference. AP a_m switches from channel k to k' if the max weighted interference seen by these cells decreases after the channel switching, i.e., the following *Local-Coord* condition holds:

$$\max_{i \in \mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f_i}(\vec{f}) > \max_{i \in \mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f'_i}(\vec{f'}), \qquad (5)$$

where $\mathbb{H}_{m,k,k'}(\vec{f})$ has been defined in Section II-B. This algorithm is denoted *Local-Coord*, since a_m needs to *locally* coordinate with the APs indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$ via wired backbone network for the channel switching.

For example, Fig. 1 depicts the cells that see changes in weighted interference before and after AP-1 switches its channel. Since the max weighted interference seen by Cells $1-4$ decreases, AP-1 can switch to the new channel.

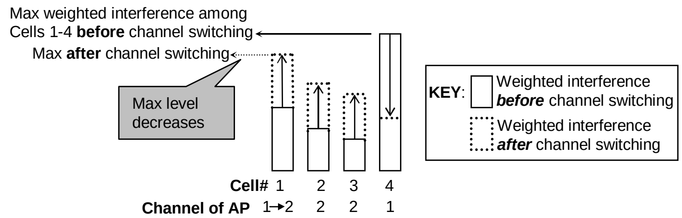

Fig. 1. Decrease of the max weighted interference seen by Cells $1-4$ *before* and *after* AP-1 switches from Channel 1 to 2.

TABLE I
A VARIABLE ψ_m USED IN THE DISTRIBUTED PROTOCOL IN FIG. 2.

ψ_m	Channel switching at a_m	Can a_m be locked?
-1	a_m is in the process of switching its channel	No
0	a_m can initiate the process of channel switching	Yes
1 or more	a_m cannot initiate the process of channel switching	Yes

Since coordination among APs is confined in a local area, multiple APs that are far apart enough can change their channels simultaneously if a proper inter-AP protocol is employed. In general, the number of APs that can simultaneously change channels grows with the number of total APs; hence, *Local-Coord* is scalable. Fig. 2 presents a distributed protocol implementing *Local-Coord*. We say an AP a_m is *locked*, if a_m is not allowed to switch its channel per other APs' requests; if a_m is *unlocked*, a_m may switch its channel. First we suppose that each AP has an independent random timer that triggers the AP to initiate the process of switching its channel as described in Fig. 2(a). If a_m is locked, a_m will ignore this trigger and wait for next trigger. The key idea of this protocol is described in Phases 1 and 2 in Fig. 2(a) that a_m needs to lock all the APs indexed by $\mathbb{V}_m$ (as defined in Section II-B) before a_m switches to a new channel; then a_m unlocks those APs after the channel switching. If any AP indexed by $\mathbb{V}_m$ cannot be locked, a_m cannot switch its channel. The procedure to handle locking and unlocking requests are described in Fig. 2(b) and (c), respectively. An AP can be locked for multiple times by different APs; Table I describes ψ_m, which denotes the number of times that a_m was locked. Only when $\psi_m = 0$ can a_m initiate the process of channel switching. When a_m is in the

(a) Suppose a timer triggers a_m to consider initiating a channel switching. Then a_m will do the following procedure.

1: **if** $\psi_m = 0$ **then**
2: *Phase 1*: Set $\psi_m = -1$ and send requests to lock all APs indexed by $\mathbb{V}_m$, i.e., $\{a_n : n \in \mathbb{V}_m\}$.
3: *Phase 2*: Wait for replies from $\{a_n : n \in \mathbb{V}_m\}$.
4: **if** If the replies indicate that $\{a_n : n \in \mathbb{V}_m\}$ were all successfully locked by a_m **then**
5: a_m switches its channel from k to k', and stays at k' if (5) is satisfied; otherwise, a_m switches back to channel k.
6: Send messages to unlock $\{a_n : n \in \mathbb{V}_m\}$.
7: **else**
8: Send messages to unlock the APs among $\{a_n : n \in \mathbb{V}_m\}$ that were just successfully locked by a_m. (Do not need to unlock the APs that could not be locked by a_m.)
9: **end if**
10: Set $\psi_m = 0$.
11: **end if**

(b) Upon receiving a *locking request* from a_m, a_n will do the following procedure.

1: **if** $\psi_n \neq -1$ **then**
2: Increase ψ_n by one.
3: Reply to a_m that a_n was successfully locked by a_m.
4: **else**
5: Reply to a_m that a_n could not be locked.
6: **end if**

(c) Upon receiving an *unlocking request* from a_m, a_n will decrease ψ_n by one.

Fig. 2. A protocol for the distributed implementation of *Local-Coord*.

process of switching its channel (denoted by $\psi_m = -1$), it cannot be locked.

Deadlock is a problem that needs to be avoided in distributed computing; in this context *deadlock* means that two or more APs that have initiated the process of switching their channels are waiting for one other, and thus none of these APs can ever finish. In the 6th and 8th steps of Fig. 2(a), a_m unlocks other APs immediately no matter whether a_m can switch its channel; hence, *deadlock never arises in the protocol in Fig. 2*.

*C. The **Global-Coord** Algorithm*

AP a_m will switch to a new channel only if the *sum* interference on the new channel is lower (after a_m switches there) than the *sum* interference on its current channel, i.e., the following condition holds.

$$\textit{Global-Coord}\text{ Condition: } \sum_{n:f_n=k} W_k^n(\vec{f}) > \sum_{n:f'_n=k'} W_{k'}^n(\vec{f'}). \tag{6}$$

This algorithm requires global coordination among APs using a central network controller that communicate with all APs, and is thus denoted *Global-Coord*.

D. Implementation Concerns

Note that in the descriptions of the three proposed algorithms, some terms of weighted interference are unknown before a_m switches to the new channel. An implementation may require a_m to switch to a new channel by trial, and then require one or more cells to measure and compute their weighted interference after a_m switches to the new channel. Only when all the quantities needed for the channel decisions are known can a_m decide whether switching to the new channel complies with the condition described for each algorithm. If the condition is satisfied, a_m stays on the new channel; otherwise, a_m switches back to the old channel or tries another channel. *No-Coord* requires the weighted interference at cell $\mathbb{Z}_m$, *Local-Coord* at cells indexed by $\mathbb{H}_{m,k,k'}(\vec{f})$, and *Global-Coord* at all cells.

E. Convergence and Characterization of Convergence Points

Theorem 1. *Consider a particular realization of the locations of APs and users and a weighted interference function of the form of (1). Given any set of initial AP channel choices, the channel selection process converges for* Local-Coord *and* Global-Coord *in a finite number of steps.*

Before characterizing the convergence points for *No-Coord*, *Local-Coord*, and *Global-Coord*, we need some definitions described below. A vector of frequency allocations denoted by $\vec{f}$ is a *Nash equilibrium* (a concept widely used in game theory [10]), if no single cell can lower its weighted interference by changing only its own channel.

Let $\vec{u} = (u_1, \ldots, u_N)$ and $\vec{u'} = (u'_1, \ldots, u'_N)$ denote the non-increasing sorted versions of two arbitrary vectors $\vec{v} = (v_1, v_2, \ldots, v_N)$ and $\vec{v'} = (v'_1, v'_2, \ldots, v'_N)$, respectively. We say that $\vec{v}$ *lexicographically dominates* $\vec{v'}$ (or $\vec{v} \succ \vec{v'}$) if there exists some index j, where $N \geq j \geq 1$ for which $u_j > u'_j$ and $u_i = u'_i$ for all $i < j$. Vectors $\vec{v}$ and $\vec{v'}$ have *the same lexicographic order* if $\vec{u}$ and $\vec{u'}$ are element-wise the same. We say $\vec{v} \succeq \vec{v'}$ if $\vec{v} \succ \vec{v'}$ or $\vec{v}$ and $\vec{v'}$ have the same lexicographic order. We say that a vector of frequency allocations denoted by $\vec{f}$ is a *local lexicographic minimum* with respect to a vector function $\vec{\theta}(\cdot)$, if for any vector of frequency allocations $\vec{f'} \in \mathbb{K}^M$ that differs from $\vec{f}$ in only one element, $\vec{\theta}(\vec{f'}) \succeq \vec{\theta}(\vec{f})$ holds true.

Theorem 2. *Suppose* No-Coord *converges to a frequency allocation $\vec{f}$. Then, $\vec{f}$ is a Nash equilibrium.*

Note that *No-Coord* does not always converge, although simulation results show that *No-Coord* converges in most cases. Theorem 2 is for the cases where *No-Coord* converges. One may limit the number of iterations or specify a minimum gradient slope to implement *No-Coord*. Below we state a technical assumption useful in proving Theorem 3.

Assumption 1. Since the weighted interference in (1) takes a continuum of values, it is reasonable to assume that the weighted interference values at different cells or channels are distinct, i.e., $\forall k, j \in \mathbb{K}, \forall m, n \in \mathbb{M}$ such that $k \neq j$ or $m \neq n$, we have $W_k^m(\vec{f}) \neq W_j^n(\vec{f})$ with probability one.

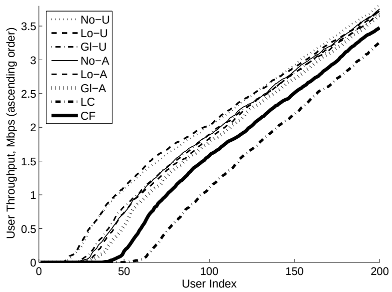

Fig. 3. *User throughput (in Mbps)* comparison in a setting with APs on a *uniform* 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with lower throughputs are shown.

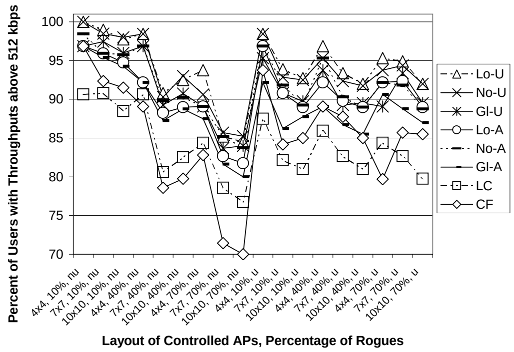

Fig. 4. Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted 'nu' and 'u', respectively.

Theorem 3. *Suppose* Local-Coord *or* Global-Coord *converge to a frequency allocation* $\vec{f}$*. Then with probability one,* $\vec{f}$ *is a* local lexicographic minimum *with respect to the vector function* $\vec{\alpha}(\cdot)$ *as defined in (7) for* Local-Coord*, or with respect to* $\vec{\beta}(\cdot)$ *as defined in (8) for* Global-Coord*, where*

$$\vec{\alpha}(\vec{f}) = \left(W^1_{f_1}(\vec{f}), W^2_{f_2}(\vec{f}), \ldots, W^M_{f_M}(\vec{f})\right) \tag{7}$$

$$\vec{\beta}(\vec{f}) = \Big(\sum_{n:f_n=1} W^n_1(\vec{f}), \sum_{n:f_n=2} W^n_2(\vec{f}), \ldots, \sum_{n:f_n=K} W^n_K(\vec{f})\Big). \tag{8}$$

IV. Simulation Results

No-Coord, *Local-Coord*, and *Global-Coord*, along with the *user-based* weighted interference function in (2) and the *AP-based* in (3), yield six combinations, *No-U*, *Lo-U*, *Gl-U*, *No-A*, *Lo-A*, and *Gl-A*. The algorithm in [4], denoted *CF*, has been shown to outperform [1]–[3]. Hence, we compare our proposed algorithms (the six combinations above) against *CF* and the algorithm in [5], denoted *LC*. The number of orthogonal channels (K) is set to 3 to represent 802.11b/g; other larger values of K produce very similar trends as to those shown in Fig. 3-5, making our approach applicable to cellular networks and 802.11a. We assume each AP can source up to 54 Mbps per the 802.11g standard. We consider 3 network sizes, 3 levels of rogue interference, and 2 network topologies, and thus have 18 combinations, as shown in the x-axis of Fig. 4. The 3 network sizes include a 4-by-4 AP layout with 64 users, a 7-by-7 layout with 196 users, and a 10-by-10 layout with 400 users; an AP is associated with 4 users in average. We consider a uniform topology where APs are regularly located, and a nonuniform topology where APs are perturbed from the uniform layout with a random distance up to 25% of separation. The number of rogues is 10%, 40%, or 70% as compared to the number of controlled APs. First, we consider a saturated network where all APs are transmitting downlink traffic, and found that *Lo-U* outperforms the best published work, *CF*, by 6.8% and 7.2% for mean and median user throughputs respectively, and 28%, 55%, 160%, and 7690% for 25, 20, 15, and 10 percentiles of user throughputs, respectively, as shown in Fig. 3. Fig. 3 also shows that *No-U* outperforms *CF* by 10% and 9.3% for mean and median respectively, and 28%, 48%, 151%, and 7480% for 25, 20, 15, and 10 percentiles of user throughputs, respectively; our algorithms yield significant throughput gains *especially for users with low throughputs*. Fig. 4 shows that our algorithms enable more users to operate above 512 kbps irrespective of the number of APs and rogues; this trend is also true for other throughput thresholds. Second, we set the ratio of downlink to uplink traffic to be 5:1 [6], and found that our algorithms consistently yield throughput gains (including both downlink and uplink) irrespective of the probability of AP activity, as shown in Fig. 5. Fig. 5 also shows *No-U* is slightly higher than *Lo-U* for 50 and 25 percentiles of users' throughputs. Details of the simulation setup and more results are presented in [11].

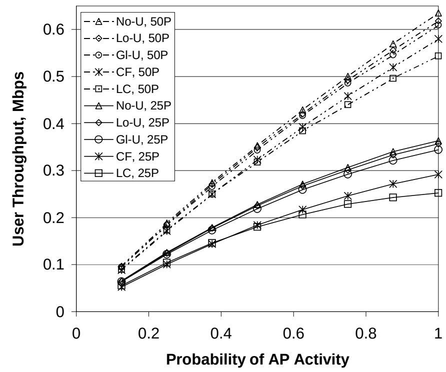

Fig. 5. 50 and 25 percentiles of users' throughputs (50P and 25P) respectively, including both downlink and uplink traffic, for 400 users on a 10-by-10 uniform AP layout with 70 rogues.

V. Conclusions

The three proposed algorithms substantially outperform all other published ones. Among the three algorithms, *Local-Coord* is the best in uplifting the throughputs of users that suffer low throughputs. For *Local-Coord*, a scalable distributed protocol is given, and the convergence is guaranteed; hence, *Local-Coord* should be the best algorithm for frequency allocation in wireless networks. If coordination among APs cannot be realized as required in *Local-Coord*, *No-Coord* is also a good option, since it does not need coordination among APs. Although *No-Coord* is not guaranteed to converge, simulations show that it converges in most cases and has comparable throughput gain as *Local-Coord*, and practical way to implement is given. Ongoing and future work considers using knowledge of building layouts and locations of APs and users to further improve the frequency allocations [8], [9], [11].

Appendix: Proofs of Theorems 1-3

Lemma 1. *Suppose two vectors* $\vec{v} = (v_1, v_2, \ldots, v_N)$ *and* $\vec{v'} = (v'_1, v'_2, \ldots, v'_N)$ *differ in at least one element. Assume all elements in* $\vec{v}$ *are distinct, and so are those in* $\vec{v'}$*. Let* $\mathbb{D}$ *denote indices where* $\vec{v}$ *and* $\vec{v'}$ *differ, i.e.,* $\mathbb{D} = \{i : v_i \neq v'_i\}$*. Then we have* $\vec{v} \succ \vec{v'}$ *if* $\max_{i\in\mathbb{D}} v_i > \max_{i\in\mathbb{D}} v'_i$*.*

Sketch. We sort the elements of $\vec{v}$ and $\vec{v'}$ respectively in descending order, and compare their elements one by one from the largest to the smallest. Then the first different pair of elements between the two sorted vectors is $\max_{i\in\mathbb{D}} v_i$ and $\max_{i\in\mathbb{D}} v'_i$ respectively. Since $\max_{i\in\mathbb{D}} v_i > \max_{i\in\mathbb{D}} v'_i$, we have $\vec{v} \succ \vec{v'}$ according to the definition of lexicographic order in Section III-E. The detailed proof is given in [11]. □

Lemma 2. *Suppose* a_m *is a representative AP switching its channel from* k *to* k' *according to the* Local-Coord *Condition in (5) or* Global-Coord *Condition in (6), and the channels of all the other APs remain unchanged. Then we have* $\vec{\alpha}(\vec{f}) \succ \vec{\alpha}(\vec{f'})$ *for* Local-Coord*, or* $\vec{\beta}(\vec{f}) \succ \vec{\beta}(\vec{f'})$ *for* Global-Coord *(*$\vec{\alpha}(\vec{f})$ *defined in (7) and* $\vec{\beta}(\vec{f})$ *defined in (8)).*

Proof. If a_m switches from channel k to k', only the cells indexed by $\mathbb{H}_{m,k,k'}(\vec{f})$ see changes in their weighted interference. Note the n-th element of $\vec{\alpha}(\vec{f})$ signifies the weighted interference of $\mathbb{Z}_n$. Hence, the different elements between $\vec{\alpha}(\vec{f})$ and $\vec{\alpha}(\vec{f'})$ are those indexed by $\mathbb{H}_{m,k,k'}(\vec{f})$. According to Lemma 1, it suffices to show that the maximum of these different elements in $\vec{\alpha}(\vec{f})$ is greater than the maximum of those in $\vec{\alpha}(\vec{f'})$, i.e., $\max_{i\in\mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f_i}(\vec{f}) > \max_{i\in\mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f'_i}(\vec{f'})$, which is equal to the *Local-Coord* condition in (5). Hence, the proof for *Local-Coord* is done. The proof for *Global-Coord* is similar and is omitted for the sake of brevity (see [11] for the proof). □

Proof of Theorem 1. We will first prove the convergence of *Local-Coord*. We form a directed graph $\mathcal{G}$ with all possible channel vectors $\vec{f}$ as *nodes* (hence the number of nodes is finite), and all channel adjustments that satisfy *Local-Coord* Condition in (5) as *edges*, assuming only one AP switches its channel at any point of time. We will show that this graph is *acyclic*; then since $\mathcal{G}$ is acyclic and finite, any initial node will converge to a sink in a finite number of steps of channel adjustments. Note that lexicographic order possesses the transitive property, that is, if $\vec{v} \succ \vec{v'}$ and $\vec{v'} \succ \vec{v''}$, then $\vec{v} \succ \vec{v''}$ [12]. Suppose *there exists a cycle on* $\mathcal{G}$, and $\vec{f}^0, \vec{f}^1, \vec{f}^2, \ldots$ are nodes on this cycle. As we travel through this cycle once, we will see that $\vec{\alpha}(\vec{f}^0) \succ \vec{\alpha}(\vec{f}^1) \succ \vec{\alpha}(\vec{f}^2) \succ \ldots \succ \vec{\alpha}(\vec{f}^0)$ according to Lemma 2. This implies $\vec{\alpha}(\vec{f}^0) \succ \vec{\alpha}(\vec{f}^0)$ according to the transitive property, which is a contradiction since $\vec{\alpha}(\vec{f}^0)$ does *not* lexicographically dominate itself. Therefore $\mathcal{G}$ is acyclic, and the proof is done. The proof of *Global-Coord* is the same as the proof above, except that the edges of $\mathcal{G}$ are the channel adjustments satisfying the *Global-Coord* Condition in (6), and $\vec{\alpha}(\cdot)$ is replaced with $\vec{\beta}(\cdot)$. □

Proof of Theorem 2. Suppose *No-Coord* converges at a frequency allocation $\vec{f}$, but $\vec{f}$ is *not* a Nash equilibrium. Then there exists at least one AP, say a_m, and one channel f'_m ($f'_m \neq f_m$) so that a_m can switch from its current channel f_m to f'_m to strictly decrease the weighted interference of $\mathbb{Z}_m$. Then, the frequency allocation should not have converged, since a_m can switch to channel f'_m according to *No-Coord* condition in (4). This proof is done by contradiction. □

Proof of Theorem 3. Recall from the proof of Lemma 2 that $\vec{\alpha}(\vec{f})$ differs from $\vec{\alpha}(\vec{f'})$ only in the elements indexed by $\mathbb{H}_{m,k,k'}(\vec{f})$. In order to prove that $\vec{\alpha}(\vec{f'}) \succeq \vec{\alpha}(\vec{f})$ holds with probability one, it suffices to show that

$$\max_{i\in\mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f'_i}(\vec{f'}) > \max_{i\in\mathbb{H}_{m,k,k'}(\vec{f})} W^i_{f_i}(\vec{f}). \tag{9}$$

holds with probability one, according to Lemma 1. Since *Local-Coord* converges at $\vec{f}$, no AP can move to a new channel so that *Local-Coord* condition in (5) is satisfied. Hence, for every AP a_m (say it is currently on channel k) and every new channel k' ($k' \neq k$), the converse of (5) holds. The inequality in the converse of (5) holds with probability one according to Assumption 1, and is the same as (9); thus, the proof is done. The proof for *Global-Coord* is similar and is omitted for the sake of brevity (see [11] for the proof). □

References

[1] K. K. Leung and B.-J. Kim, "Frequency assignment for IEEE 802.11 wireless networks," in *Proceedings of IEEE Vehicular Technology Conference*, 2003, pp. 1422 – 1426.

[2] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP placement and channel assignment in wireless LANs," in *Proc. IEEE Conf. on Local Computer Networks (LCN)*, Nov. 2002, pp. 831 – 836.

[3] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *ACM Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 19 – 31, 2005.

[4] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless LANs," in *Proceedings of IEEE Infocom*, 2006.

[5] D. J. Leith and P. Clifford, "A self-managed distributed channel selection algorithm for WLANs," in *Proceedings of International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Apr. 2006, pp. 1 – 9.

[6] C. Na, J. K. Chen, and T. S. Rappaport, "Measured traffic statistics and throughput of IEEE 802.11b public WLAN hotspots with three different applications," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3296 – 3305, Nov. 2006.

[7] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.

[8] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74 – 80, Oct. 2003.

[9] J. K. Chen, T. S. Rappaport, and G. de Veciana, "Site specific knowledge for frequency allocations in wireless LAN and cellular networks," submitted to *IEEE Vehicular Technology Conference*, Oct. 2007.

[10] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[11] J. K. Chen, T. S. Rappaport, and G. de Veciana, "Frequency allocation and load balancing with site specific knowledge for wireless data networks," PhD Dissertation, to appear, University of Texas at Austin, May 2007.

[12] (2007, Mar.) Lexicographical order. [Online]. Available: http://en.wikipedia.org/wiki/Lexicographical_order