Weighted Coloring based Channel Assignment for WLANs

Arunesh Mishra, Suman Banerjee, William Arbaugh

***Abstract*— We propose techniques to improve the usage of wireless spectrum in the context of wireless local area networks (WLANs) using new channel assignment methods among interfering Access Points (APs). We identify new ways of channel re-use that are based on realistic interference scenarios in WLAN environments. We formulate a weighted variant of the graph coloring problem that takes into account realistic channel interference observed in wireless environments, as well as the impact of such interference on wireless users. We prove that the weighted graph coloring problem is NP-hard and propose scalable distributed algorithms that achieve significantly better performance than existing techniques for channel assignment. We evaluate our algorithms through extensive simulations and experiments over an in-building wireless testbed.**

I. Introduction

With the increased popularity and deployment of WLANs, efficient management of wireless spectrum is becoming increasingly important. In this paper we focus on a specific resource sharing problem in the context of 802.11-based WLANs — channel assignment. Consider an in-building wireless environment in which multiple Access Points (APs) are operational. Each AP operates on an administrator-specified channel. In 802.11 WLANs, the wireless card of a user scans the wireless medium to identify the access point with the strongest signal and associates with it. In order to reduce interference between different APs in the same physical neighborhood administrators conduct detailed Radio Frequency (RF) site surveys, often using spectrum analyzers, prior to setting up APs within the building and assigning specific channels to them [1].

WLANs operate in unlicensed portions of the frequency spectrum (2.4 and 5 Ghz) allotted by a regulatory body, e.g., the Federal Communications Commission in the USA. Each WLAN standard (802.11/a/b/g) defines a fixed number of channels for use by APs and mobile users. For example, the 802.11b standard defines a total of 14 frequency channels of which 1 through 11 are permitted in the US. An important concept to note regarding these channels is that the channel actually represents the center frequency that the transceiver within the radio and AP uses (e.g., 2.412 GHz for channel 1 and 2.417 GHz for channel 2). There is only 5 MHz separation between the center frequencies, and an 802.11b signal occupies approximately 30 MHz of the frequency spectrum. The signal falls within about 15 MHz of each side of the center frequency. As a result, an 802.11b signal on any channel overlaps with several adjacent channel frequencies causing interference (also known as adjacent channel interference). This leaves only three channels (channels 1, 6, and 11) that can be used simultaneously without causing interference. As a basic design rule, APs within range of each other are set to to different “non-overlapping” channels. The state-of-the-art channel assignment using non-overlapping channels is performed as follows: Each AP periodically checks other data transmissions in the channel it is using. If the volume of traffic in that channel (from other APs or clients of other APs) is greater than a threshold, then the first AP tries to move over to a less congested channel. We call this technique, *Least Congested Channel Search (LCCS)* [1].

In this paper we contend that with the continued growth of WLANs, a channel assignment scheme based solely on the LCCS rule will not be efficient. Given the unlicensed nature of WLAN technologies and decreasing costs of APs, the number of APs located in a physical neighborhood has proliferated. In many cases administrators increase the number of APs within a building to improve the wireless coverage. Additionally multiple organizations co-resident in the same building deploy independent wireless LANs, and the channel assignments made for the respective APs are made independent of each other. Hence the assignment of channels to this potentially large set of APs needs to be carefully coordinated, or else the broadcast nature of WLANs can lead to serious performance degradation of the wireless users.

Considerations for Efficient Channel Allocation

In this paper, we define scalable distributed algorithms for the channel assignment problem that tries to optimize user performance in wireless LAN environments with multiple APs. Our formulation and solution to the WLAN channel assignment problem addresses two important problems which, to the best of our knowledge, have not been explicitly addressed in prior literature in the context of WLANs. They are:

Dynamic channel re-use: Consider the simple scenario shown in Figure 1. For simplicity of illustration,

A. Mishra and W. Arbaugh are with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Emails: {arunesh,waa}@cs.umd.edu. S. Banerjee is with the Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA. Email: suman@cs.wisc.edu.

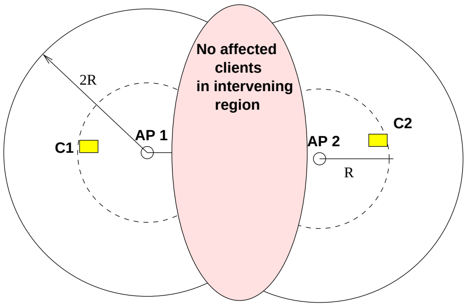


Fig. 1. Based on the impact on mobile users, some of the scenarios may not require the APs to use non-overlapping channels.

consider circular ranges. Each AP has a transmission range of R. The clients associated to a particular AP would interfere upto a region of radius $2R$. To maximize channel re-use we would like to assign the same channel to both APs. However, presence of clients in the common interference region of the two APs would result in throughput degradation due to the classic hidden terminal problem [2], [3]. In the absence of such users, our algorithms make use of such opportunities to improve channel re-use by dynamically capturing the client distribution. Unfortunately, an RF site survey at either of the AP locations or the LCCS heuristic does not expose this information to the site administrator.

Weighted channel assignments: A channel assignment problem is typically modeled as a graph-coloring problem — there is a vertex on the graph corresponding to each AP, an edge on this graph represents potential interference, and the colors represent the number of non-overlapping channels. A goal of the channel assignment problem is to cover all APs (vertices) with the minimum number of channels (colors) such that no two adjacent APs (vertices) use the same channel (color). This is the minimum graph coloring problem.

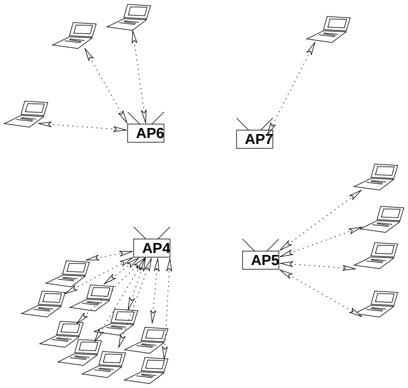


Fig. 2. All APs are in interference range of each other and assuming we have only three non-overlapping channels, we need to assign overlapping channels to APs in this case.

However, the number of non-overlapping channels available in a specific WLAN technology is bounded. For example, in 802.11b in the US there are only three independent, non-overlapping channels, 1, 6, and 11. Consider the scenario in Figure 2 in which the four APs (AP_4, AP_5, AP_6, and AP_7) are in close proximity of each other. There are 9, 4, 3, and 1 mobile clients associated with them respectively. In such scenario if we restrict our channel choices to only the non-overlapping channels (1, 6, and 11) two APs will be assigned the same channel and will experience significant interference. We demonstrate in this paper that better performance can be achieved by assigning partially overlapping channels to the APs. One possible assignment of channels to APs is: AP_4 to channel 1, AP_5 to channel 6, AP_6 to channel 11, and AP_7 to channel 9. (The channels are separated by gaps of 4,2, and 1 intervening channels.) Clearly clients of AP_5, AP_6, and AP_7, will experience some interference. AP_4 encounters no interference from other APs and their clients. This is desirable because the number of clients it is currently supporting is greater than the total number of clients in all the other APs. Note that its also possible to consider alternative channel assignments, e.g. AP_4 to channel 1, AP_5 to channel 11, AP_6 to channel 5, and AP_7 to channel 8. We discuss this further in Section II-C.

Based on these observations, we define a weighted variant of the graph-coloring problem, in which it is permissible to allocate overlapping channels to neighboring APs. Our goal in this variant is to minimize the impact of such overlapping channel assignments between neighboring APs on user performance. We discuss the exact formulation of this problem in Section II. Unlike the RF site survey based static techniques, our proposed solution enables self-configuration of channel assignment to APs based on continuously evolving channel conditions in the wireless environment.

We present two techniques in this paper that are applicable to a wide range of WLANs technologies, e.g., 802.11b, 802.11a, 802.11g, etc. The first technique does not require any collaboration among the APs and can be applied to a wireless network formed by APs belonging to different WLANs. The technique assumes that the APs are greedy in nature – they try to minimize the interference within their area of coverage [1]. We also present a second technique that drastically reduces the total number of clients suffering interference in the network as a whole. However, this technique requires collaboration among the APs. This is particularly suited for efficient channel assignment for a single wireless network, or for multiple wireless networks if the administrators are willing to co-

[1] That is, the APs are greedy yet non-malicious.

operate.

Through simulations, we show that the two techniques presented in this paper dramatically improve interference. In particular, with 3 non-overlapping channels, the techniques achieve a 45.5% reduction in interference for sparse topologies. Through simulations, we also show that even if only 10% of the total APs use the techniques, the network as whole still achieves up to 40% reduction in the clients suffering interference. We also demonstrate similar performance results on an in-building wireless network.

The rest of the paper is structured as follows: In Section II we present the formulation of our channel assignment problem for wireless LANs. In section III, we discuss a graph theoretic model for capturing interference and provide distributed algorithms in section IV. Section V discusses the simulation results, and the experiments are discussed in section VI. In Section VII we describe the related work. We conclude in section VIII.

II. Improving Channel Re-use

Apart from static radio-map based techniques [1], access points provide a method of autonomously searching for the least congested channel [4], called the Least Congested Channel Search (LCCS). Below we discuss this technique and its limitations:

A. Limitations of LCCS:

In this technique, each AP monitors packet transmissions on its channel of operation. When it detects such transmissions from other APs or clients associated to other APs (within its vicinity), it scans for an alternative "less-congested" channel. In order to be able to search for the least congested channel (channel with least number of other APs assigned in the vicinity), the AP should be within the transmission range of stations associated to other access points. Based on this observation, LCCS suffers from the following drawbacks:

1) *LCCS fails to capture certain scenarios of interference:* LCCS fails to capture situations when clients associated to two different APs interfere with each other, while the APs do not interfere among themselves. Such situations can frequently arise in practice, as APs are typically spread sufficiently far apart to maximize coverage while maintaining some overlap to remove "holes" in the coverage region.
2) *LCCS lacks ability to improve channel re-use based on client distribution:* LCCS cannot capture interference experienced at the clients, and hence cannot take advantage of opportunities of channel re-use created by client distributions. We discuss this further in the next section.

B. Our Opportunities for Efficient Channel Re-use

The techniques proposed in this paper significantly improve the channel assignment mechanisms for WLANs by automatically detecting opportunities for greater channel re-use when possible, and also correctly handling situations that would lead to greater interference between neighboring APs and their clients.

For example, as shown in Figure 1, presence of significant number of clients in the interference region would translate to a higher chance that the APs are assigned to different channels. Similarly, if the region becomes devoid of clients, our algorithms assign the same channel to both APs which improves channel re-use. Such opportunistic channel re-use enables the algorithms keep the channel assignment in sync with the client distribution.

C. Exploiting Variations in Adjacent Channel Interference

Adjacent channel interference refers to the interference caused on a particular channel by a station transmitting on a neighboring channel. Two channels are *independent* or non-overlapping if there is negligible interference between stations operating simultaneously on the two channels. Adjacent channel interference is present in IEEE 802.11 a/b/g networks.

To better understand such adjacent channel interference we performed a simple experiment with IEEE 802.11b. A transmitting station is placed on channel 6, and a receiving station is moved from channel 1 through 11. Table I shows the signal to noise ratio (SNR) (normalized to $\{0 \ldots 1\}$) at the receiving station for different receiving channel choices. The *iwconfig* utility, part of the Linux Wireless Tools package [2], was used to measure the signal strength. We can see that the interference between channels 1 and 6, and channels 6 and 11 is very low. Hence channels 1, 6, and 11 are considered non-overlapping for all practical purposes. Ideally we will like to use such non-overlapping channels to avoid any conflict in the WLAN environment.

However, there exists other channel pairs where the interference is fairly low. For example, channels 2 and 6, and channels 6 and 10. Although the interference on these channels pairs are greater than that between 1, 6, or 11, such interference can still be considered low for many practical scenarios.

We will now define a graph theoretic formulation of our problem where we present a weighted variant of the traditional graph coloring problem, in which the weights will

[2] See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

Channel	1	2	3	4	5	6	7	8	9	10	11
Normalized SNR (I-factor)	0	0.22	0.60	0.72	0.77	1.0	0.96	0.77	0.66	0.39	0

TABLE I. Table shows the signal-to-noise ratio (SNR) normalized to a scale of $0 \ldots 1$ of the transmission made on channel 6 as received on channels $1 \ldots 11$. We call this quantity *I-factor*, see Section III.

capture the impact of co-channel interference on mobile users, as well as the impact of interference between (potentially) overlapping channels on these users.

III. Graph Theoretic Formulation

In this section, we present a graph theoretic model to capture constraints on channel assignment among interfering access points.

A. Overview

The channel assignment problem for WLANs can be modeled as a graph coloring problem in which the APs are the vertices of a graph. A conflict between two APs (due to physical proximity and potential interference) is represented by an edge in the graph. The goal of this graph coloring problem is to assign a set of distinct colors (one corresponding to each available channel). To enable the most efficient re-use of these channels the objective of this problem will be to color the graph with the minimum number of colors, a problem which is known to be NP-Hard for general graphs. We call this model, the unweighted graph coloring problem.

Note that in the unweighted problem, the colors correspond to non-overlapping channels. As the number of APs within a given physical region grows, the number of colors (non-overlapping channels) needed to color the graph will exceed the number of available colors (i.e., the maximum number of non-overlapping channels).

To enable an efficient channel assignment under such circumstances we extend the above graph theoretic formulation to a *weighted* graph coloring problem with a certain *objective function*. In this weighted variant, each vertex corresponds to a distinct AP as before. However, each edge on this graph now has a weight associated with it. The weight of an edge indicates the importance of using different colors (channels) for the corresponding vertices (APs) that are connected by that edge. In the rest of this discussion we will use APs and vertices and similarly colors and channels interchangeably.

One informal strategy for assigning weights to APs can be the total number of active clients, associated to these two APs that will interfere with each other, if both these APs are assigned the same channel. We will discuss the exact computation of these weights(and how these weights translate to interference suffered by clients) in Section IV-C.

We illustrate the advantage of using weights (over the unweighted case) with an example. Consider a graph thats (say) 4 colorable and we have only 3 colors/channels. There would be many ways to 3-color this graph ; each coloring would result in a set of *conflict* edges (i.e. edges whose constituent vertices have the same channel/color). Depending on user distributions, certain overlap regions (i.e. regions within coverage of the two APs forming a conflict edge) might have a large number of users than others. Thus, to improve per-user and network throughput, it would be better to assign non-overlapping channels to such edges preferentially than to the rest. Having weights on edges as a measure of the interference between APs (eg., in terms of the number of users) assits the algorithms in taking advantage of such opportunistic scenarios. Also, partial overlap between channels is modeled as a multiplicative factor (called the *I-factor*, discussed later) in the weight of an edge; a higher weight indicating greater amount of overlap between the channels. Such a model allows the algorithms to use the 11 partially overlapping channels in 802.11b.

If two APs are connected by an edge with a high weight, we will want to assign different channels to them. If two such vertices are assigned non-overlapping channels, then the contribution of this edge to the objective function is zero (since the clients of these two APs do not interfere). However, our discussion in Section II indicates that it is possible to assign partially overlapping channels to such neighboring APs. In such a scenario, this edge contributes a positive value to the objective function, e.g., it can be the number of clients affected by interference, scaled by the degree of interference between the chosen channels. Clearly if the two APs are assigned to the same channel, the contribution of this edge to the objective function is maximum, and decreases with decreasing overlap between the assigned channels. Consequently the goal of algorithms solving the weighted graph coloring algorithm is to minimize this objective function, thereby minimizing the impact of interference on clients.

We discuss three specific objective functions for this minimum weighted graph coloring problem, and show that the problem is NP-hard for all three. Note that the graph model presented in this section and the techniques presented in section IV can be applied to *any* set of APs, sharing the same RF spectrum in the region of interest and belonging to potentially different WLANs with arbitrary topologies.

B. Formal definition

Let k denote the total number of non-overlapping channels available in the underlying wireless PHY layer[3]. Given a region of interest covered with a set of access points, define an *overlap* graph $G = (V, E)$ as follows: $V = \{ap_1, ap_2, \dots, ap_n\}$ be the set of n APs that form the network. Place an edge between APs ap_i and ap_j ($ap_i \neq ap_j$) if there is an overlap in the interference region of the BSS created by APs according to the interference model.

Let W be the *weight function* on G; $W(ap_i, ap_j)$ denotes the *normalized* weight on the edge (ap_i, ap_j). In this discussion we assume that the weight of an edge indicates the number of clients associated with the two corresponding APs that are affected if these APs are assigned the same channel.

The weighted graph coloring problem for channel assignment is stated as follows: A channel assignment $C(ap_i), ap_i \in V$ is a mapping $C : V \rightarrow \{1 \dots k\}$ from the set of vertices to the set of colors. We say that an edge (ap_i, ap_j) is *conflict free* edge if the interference between these two channels is zero (e.g. channels 1 and 6 in 802.11b). Else if the choice of colors have some positive interference (e.g. say channels 1 and 2 in 802.11b) we call the edge (ap_i, ap_j) a *conflict edge*. We define a term *Interference-factor* or I-factor, denoted by $I(ap_i, ap_j)$, for each edge, which is the interference between the colors assigned to the two APs. Note that the I-factor of all conflict-free edges is zero. The I-factor can be measured empirically, for example, as shown in Table I. We call the product, $W(ap_i, ap_j) \times I(ap_i, ap_j)$ as the I-value. The I-value represents the total effect of interference on all clients that fall in an overlapping region between two APs. We informally call this the conflict edge weight. Thus, given G and W as defined above, the channel assignment problem is to find a mapping C such that an objective function is optimized. Below, we define three different objective functions $L_{max}, L_{sum}, L_{num}$, which we shall also use as metrics to evaluate the distributed channel assignment algorithms presented in section IV.

1) *Minimize the I-value among all interfering APs:* Minimize :

$$L_{max}(G, C) = \underset{\forall e=(ap_i, ap_j) \in E}{Max} I(ap_i, ap_j) W(ap_i, ap_j)$$

We call the problem of minimizing L_{max}, as defined above, a *min-max* graph coloring problem. This minimizes the maximum I-value of an AP in the graph. Informally this implies that we are minimizing the maximum impact of interference among all overlap regions between APs.

2) *Minimize the sum of weights on all conflict edges:* Minimize :

$$L_{sum}(G, C) = \sum_{\forall e=(ap_i, ap_j) \in E} I(ap_i, ap_j) W(ap_i, ap_j)$$

L_{sum}, as defined above, minimizes the sum of the I-factors. This informally translates to minimizing the total effect of all interference experienced by clients as a consequence of channel assignments.

3) *Minimize the number of conflict edges:*

$$L_{num}(G, C) = \sum_{\forall e=(ap_i, ap_j) \in E} I(ap_i, ap_j)$$

L_{num}, as defined above, minimizes the total impact of all conflict edges, and hence translates to minimizing the total effect of interfering APs.

Non-overlapping Channels: Special Case: Note that the non-overlapping channel assignment problem is a special case of our problem formulation above. It can be obtained by defining the I-factor as $I(ap_i, ap_j) = 1$ if ap_i and ap_j are on the same channel, $I(ap_i, ap_j) = 0$ otherwise, and restricting the choice of channels to the corresponding non-overlapping channels.

Although all three objective functions translate to meaningful optimizations, we focus on the L_{max} objective function as it minimizes the maximum I-value among all overlap regions and hence maximizes available bandwidth on a per-channel basis.

Below, we show that the above problem is NP-hard for the L_{max} objective function. On similar lines, one can show NP-hardness for the other objective functions.

Theorem: Given a graph weighted undirected graph $G = (V, E)$, with n vertices, $V = \{v_1, v_2, \dots, v_n\}$, and a weight function $\Gamma : E \longrightarrow \{1 \dots 100\}$. Let $C : V \longrightarrow \{1 \dots k\}$ denote a k-coloring of G. The problem of finding P that minimizes

$$L_G = \underset{\forall e=(v_i, v_j) \in E \bigwedge C(v_i)=C(v_j)}{Max} \Gamma(v_i, v_j)$$

is *NP-hard*.

Proof: Consider the general problem of coloring an undirected graph: Given a graph $G = (V, E)$, does there exist a k-coloring $C : v \longrightarrow \{1 \dots k\}$, $1 \leq k \leq |V|$, such

[3] Equal to 3 for IEEE 802.11b/g.

that $\forall e = (v_i, v_j) \in E \Rightarrow C(v_i) \neq C(v_j)$. By assigning a weight of 1 to all edges, we formulate this as a *min-max* coloring problem. Let P be the solution obtained for the *min-max* version, it can be seen that $L_G = 0$ iff G has a k-coloring. Thus the result follows from the hardness of the general graph coloring problem.

Since the unweighted graph coloring problem is known to be NP-Hard, a number of techniques have been proposed in the literature to approximately solve the unweighted graph coloring problem [5]. The focus of this paper is to define scalable distributed and fault tolerant techniques for the weighted case which can also be applied to coexisting wireless networks and to evaluate such techniques through extensive simulations and experiments on a real testbed.

IV. Distributed Channel Allocation Algorithms

In this paper we primarily focus on the L_{max} objective function and we present two different strategies which we call *Hminmax* and *Hsum* to minimize this objective function.

Both algorithms are executed at APs in a distributed manner. Both algorithms work on *local* information (i.e. each AP collects information from its neighbors) and are hence scalable in the size of the wireless network. The algorithms are incremental in nature, hence they can adapt the channel assignment to reflect changes to the graph topology.

The first algorithm, *Hminmax*, does not require communication between the APs, and applies to multiple coexisting wireless networks sharing a limited RF spectrum. It attempts to reduce the L_{max} objective function. This algorithm applies to scenarios where the APs are greedy yet non-malicious in nature. The second algorithm, *Hsum*, tries to reduce the L_{max} objective function (like the *Hminmax*) algorithm, but additionally is also able to reduce the L_{sum} objective function to a significantly lower value than what *Hminmax* is able to achieve, without compromising the L_{max} objective function. To do this, the *Hsum* algorithm requires some cooperation between APs which results in better throughputs for the network as a whole. It uses this coordination function to operate more intelligently than the basic *Hminmax* algorithm. Such coordination can be implemented using the Inter-Access Point Protocol [6], discussed later.

```
Procedure : Hminmax($ap_i$)
1. Initialize:
    $C(ap_i) \leftarrow 1$
      { initial coloring for all nodes }

2. Optimize:
    (a) $H(c) \leftarrow \underset{\forall e=(ap_i,ap_j)\in E \wedge C(ap_j)=c}{Max} I(ap_i,ap_j)W(ap_i,ap_j)$
          { $H(c)$ is max. weight of any edge to $ap_i$
          whose neighbor has a color $c$ }
    (b) Choose $c'$ such that $H(c') = \underset{c=1\ldots k}{Min} H(c)$
      { Choose color with min. conflict weight }
    (c) $C(ap_i) \leftarrow c'$
```

Fig. 3. Algorithm *Hminmax.*

We describe the algorithms using the notation presented in Section III. We define the following additional terms: k is the number of available channels. $N(ap_i)$ denotes the set of neighbors of ap_i, i.e. $N(ap_i) = \{ap_j | e = (ap_i, ap_j) \in E \wedge ap_i \neq ap_j\}$. For simplicity of presentation, we define the following objective functions, corresponding to L_{max} and L_{sum} respectively, which are local to an AP ap_i:

$$L_{max}(G, C, ap_i) = \underset{\forall e=(ap_i,ap_j)\in E}{Max} I(ap_i, ap_j)W(ap_i, ap_j) \tag{1}$$

Informally, $L_{max}(G, C, ap_i)$ is the maximum conflict weight of an edge incident on ap_i.

$$L_{sum}(G, C, ap_i) = \sum_{\forall e=(ap_i,ap_j)\in E} I(ap_i, ap_j)W(ap_i, ap_j) \tag{2}$$

Informally, $L_{sum}(G, C, ap_i)$ is the aggregate of all weights of conflict edges incident on ap_i.

A. Algorithm Hminmax:

The algorithm, outlined in Figure 3, has two steps discussed in detail below:

Initialization step: This step selects an initial assignment for C. This can be as simple as assigning the same color to all APs, or the APs can perform a Least Congested Channel Search (LCCS) (discussed in section II) and leverage that as an initial coloring.

Optimization step: This step improves the coloring in an incremental manner. Each AP ap_i attempts to minimize interference on its maximum conflict edge by selecting the color which minimizes $L_{max}(G, C, ap_i)$. This is done in steps 2(a) and 2(b) of the algorithm. The optimization step is executed by each AP periodically. This moves the network to a better coloring and keeps the channel assignment in synchrony with changes to the graph topology.

In *Hminmax*, each AP updates its channel of operation by performing a local optimization as indicated in

Figure 3. These channel changes are performed asychronously and on a timescale of the order of 10-100 minutes. Changing the channel of operation is a primitive that has to be supported by the APs and the stations. The channel change can be performed right after a special beacon message broadcast by the AP. This latency would be expected to be around 1-2 ms[4] which is small compared to the duration that the AP stays on that channel. This makes the probability of two APs changing their channels simultaeneously to be negligible, hence allowing the APs to execute this algorithm without any coordination.

B. Algorithm Hsum:

We now present the *Hsum* algorithm which performs equally well (as *Hminmax*) in reducing the L_{max} objective function, but additionally is also able to reduce the L_{sum} objective function without compromising the L_{max} objective. *Hsum* is able to perform such an improved channel assignment by requiring some coordination between participating APs to communicate the L_{max} value during the execution of the algorithm. Hence, this algorithm particularly applies to wireless networks under the same administrative domain. The algorithm can also be applied to situations where APs from different WLANs are willing to coordinate and communicate with each. Such a communication can be achieved using the IAPP [6] protocol.

Due to space constraints, in this paper we just provide a high level description of this algorithm. Pseudo-code is outline in Figure 4.

The previous algorithm *Hminmax*, each AP performs a a greedy optimization trying to minimize its local maximum interference, i.e. minimize $L_{max}(G, C, ap_i)$. Although this reduces the maximum interference for each AP, it can create many conflict edges with low weights, which can lead to potentially large aggregate interference in the entire network, i.e. L_{sum}. Hence *Hsum* tries to additionally reduce the L_{sum} objective function as well as follows: It minimizes $L_{max}(G, C, ap_i)$ if $L_{max}(G, C) = L_{max}(G, C, ap_i)$. In other words, if the AP ap_i has a conflict edge which has the maximum weight over the entire graph, it selects a coloring which minimizes L_{max}, otherwise it minimizes the the aggregate interference within its area of coverage, i.e., $L_{sum}(G, C, ap_i)$. IAPP [6] based coordination is necessary for this algorithm because each AP needs to periodically evaluate the global value of L_{max}.

For restricting the assignment to only non-overlapping channels, as a special case, the I-factor can be set to $I(ap_i, ap_j) = 1$ if ap_i and ap_j are on the same channel and $I(ap_i, ap_j) = 0$ otherwise, in the above algorithm. This has been discussed in section III.

```
Procedure : Hsum($ap_i$)
1. Initialize:
    $C(ap_i) \leftarrow 1$
      { initial coloring for all nodes }

2. Optimize:
      { Define $w_{max}$ as current value of $L_{max}(C)$. }
    (a) $H(c) \leftarrow \underset{\forall e=(ap_i,ap_j)\in E \wedge C(ap_j)=c}{Max} I(ap_i,ap_j)W(ap_i,ap_j)$
    (b) if $H(c) \geq w_{max}$ $M(c) \leftarrow 1$ else $M(c) \leftarrow 0$
          { Mark colors with the max. conflict weight $> w_{max}$
          We do not want to choose such colors. }
    (c) $S(c) \leftarrow \sum_{\forall e=(ap_i,ap_j)\in E \wedge C(ap_j)=c} I(ap_i,ap_j)W(ap_i,ap_j)$
          { $S(c)$ denotes sum of weights of all edges to $ap_i$
          whose neighbor has a color $c$ }
    (d) Choose $c'$ such that $S(c') = \underset{1\leq c\leq k\wedge M(c)=0}{Min} S(c)$
          { Choose color with min. sum of conflict weights
          among all unmarked colors. }
    (e) $C(ap_i) \leftarrow c'$
```

Fig. 4. Algorithm *Hsum.*

C. Implementation Issues

Distributed Construction of Overlap Graph: We present a scalable and autonomous method of constructing the overlap graph in a distributed manner, called *Site-Report*. An AP can randomly and with very low probability, request a client to perform a site-report during intermittent periods of low activity with respect to the client. In a site-report, a client performs a passive scan of each channel and captures packets from APs and clients associated to different APs. For each channel, a list of APs is constructed. An AP is present in this list if a packet was destined or transmitted by that AP. Hence, this list includes APs that are in direct communication range and APs whose associated clients are in direct communication range with respect to the client performing the site-report. Such periodic scans capture the dynamic nature of interference in a wireless network exposing opportunities for channel re-use.

Implementing our site-report method is on current platforms would translate to a software update; future platforms would support the upcoming IEEE 802.11K Radio Resource Management standard [8] which includes a *Neighbor-Report* primitive that is similar to our site-report method.

In section VI, we demonstrate the construction of an overlap graph using the site-report for an in-building wireless network testbed.

[4] If implemented in the firmware of the wireless NIC [7].

Calculation of Weight: The weight of an edge created using the above method can be calculated as follows. Let Num_{ap_i} denote the number of site reports conducted by clients of AP ap_i. Let $Numap_i(ap_j)$ denote the number of site reports conducted by clients of AP ap_i which reported interference with ap_j. Then the weight $W(ap_i, ap_j)$ on the edge (ap_i, ap_j) is obtained by $W(ap_i, ap_j) = \frac{Num_{ap_i}(ap_j)}{Num_{ap_i}}$. Since the *site-report* is performed by clients associated to and within the range of an AP, this weight indicates the fraction of clients associated to ap_i that are affected by interference from ap_j. Thus higher the weight, greater would be number of clients affected and hence the channel assignment algorithms would assign non-overlapping or partially overlapping channels to these APs. Hence, the weights linearly map to interference suffered by clients. Note that using coordination between APs, a better statistical estimate for $W(ap_i, ap_j)$ would be $\frac{Num_{ap_i}(ap_j)+Num_{ap_j}(ap_i)}{Num_{ap_i}+Num_{ap_j}}$.

Inter-AP Protocol: The IAPP protocol is a UDP based communication vehicle for Inter-AP traffic. The primary purpose of the IAPP protocol is to notify and transfer a roaming station's context information in a secure manner between APs. For details regarding specifics of IAPP messages, refer [6]. In our work, we extend the IAPP protocol to include messages for the necessary coordination between APs. For example, IAPP broadcast messages can be used to trigger channel re-assignments and maintain a global value of L_{max} (for the *Hsum* algorithm). The details of implementation are outside the scope of this paper. Note that since IAPP operates at the network layer, its enables APs from different administrative domains/WLANs to communicate with each other. The IAPP messages would typically be exchanged using the wired Distribution System connecting the APs, and thus the overheads due to this communication are not relevant.

V. Simulations

We have evaluated the performance of our proposed algorithms through extensive simulations as well as through measurements on an operational wireless testbed, consisting of 20 APs distributed on two floors. We present the results from our simulation study in this section, and the results of our measurement study on the wireless testbed in the next section.

A. Methodology

The goal of the simulation study was (1) to understand the performance of our proposed techniques in a controlled manner, and (2) to study its scalability properties with increasing number and density of APs in the environment.

In this study we assumed that the APs were distributed over a three dimensional physical region. The locations of the APs were chosen uniformly at random for the different experiments. Each AP had a transmission radii and a larger interference radii which was chosen randomly as well.

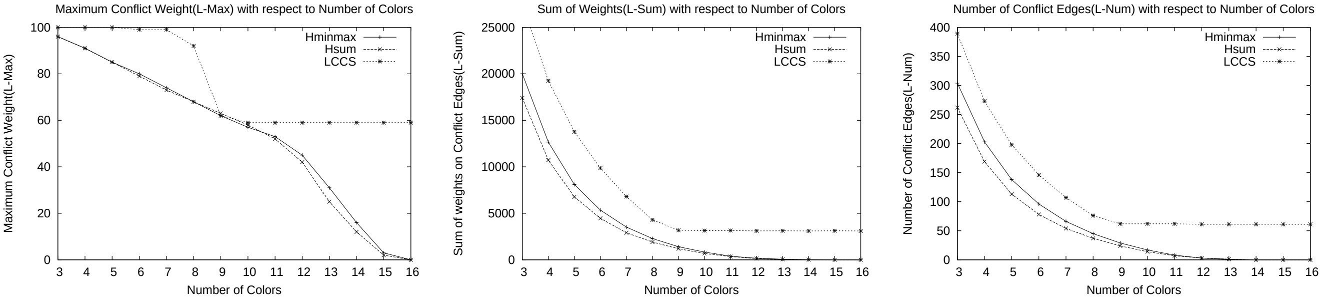
Additionally we assumed that the clients were distributed uniformly at random in the same physical space. Each client associated with a nearby AP with the strongest signal strength. Each client in this topology is an active client, i.e., it will cause and experience interference with other clients as they communicate with their respective APs. Each client also has its independent transmission and interference radii. Clearly the choice of circular radii for transmission and interference ranges is a simplifying assumption and we study the impact of more realistic physical wireless environments in the next section.

In all these experiments we compared the performance of three channel assignment algorithms, namely the current best known technique LCCS as defined in Section II, and our proposed distributed algorithms *Hminmax* and *Hsum*. The actual channel selected by the algorithms depends on the channel conditions, the number of clients associated to neighboring APs, and the network traffic going through the interfering APs. We capture this dynamic nature of the algorithm in the simulations.

Performance Metrics: We primarily examined three performance metrics for this study, as given by the three objective functions, L_{max}, L_{sum}, and L_{num}, that we defined in Section III. We discuss how these metrics directly influence user throughput in WLANs.

The weight of an edge in an overlap graph indicates (normalized with the maximum value) the fraction of users present in the overlap region of the constituent APs to the total number of users associated to both APs. As the fraction of users in the overlap region increases, the average system throughput decreases. Consider a simple setup with 2 APs shown in Figure 1. Say, there are 10 clients totally, 5 associated to each AP. Table II shows the UDP throughput obtained by NS-2 simulations on this topology with respect to increasing weight on the edge – that is, fraction of users present in the intervening region shown in Figure 1. As shown in the table, a greater fraction of users in the intervening region implies further reduction in throughput.

L_{max} is the maximum weight among all *conflict* edges in a graph. This bounds the maximum interference suffered by a user, and thus, lower the L_{max}, higher is the minimum throughput perceived by any client in the net-



(a) Hminmax, Hsum, LCCS with L_{max}

(b) Hminmax, Hsum, LCCS with L_{sum}

(c) Hminmax, Hsum, LCCS with L_{num}

Fig. 5. Figure compares the *Hminmax, Hsum* and LCCS with the number of colors. Each point is an average of simulations over 100 different graphs each with 100 vertices, and a maximum degree of 20.

Weight	0.1	0.4	0.7	1.0
Tput (Kbps)	150	80	50	20

TABLE II. Table shows the UDP throughput of a simple two-AP scenario with respect to increasing weight on the edge between the APs. The physical layer had maximum datarate of 1 Mbps.

work. Ideally, no user should be present in an overlap region, i.e., there should be no conflict edges present in the graph. The L_{num} metric captures the number of conflict edges present. For the same number of edges, it is important to note the number of users present in the overlap region. This is captured as a metric by summing up the weights over all conflict edges, or, the L_{sum} metric. Thus in any wireless environment it is always desirable to minimize each of these metrics.

Simulation Scenarios: We present two different simulation scenarios. Our first results in this section will examine the channel assignment properties of the algorithms if the objective is to only assign non-overlapping channels to APs. This is the current practice used for channel assignment today. Such a non-overlapping channel assignment problem is a special case of our problem formulation, and can be handled by both *Hminmax* and *Hsum* algorithms by appropriately choosing the I-factor, i.e., by setting $I(ap_i, ap_j) = 1$ when ap_i and ap_j are on the same channel and $I(ap_i, ap_j) = 0$ otherwise, and restricting the choice of channels to precisely the set of non-overlapping channels. In our second scenario, we describe the performance of the algorithms when it is permissible to use overlapping channels. Clearly such an approach will lead to better spatial re-use.

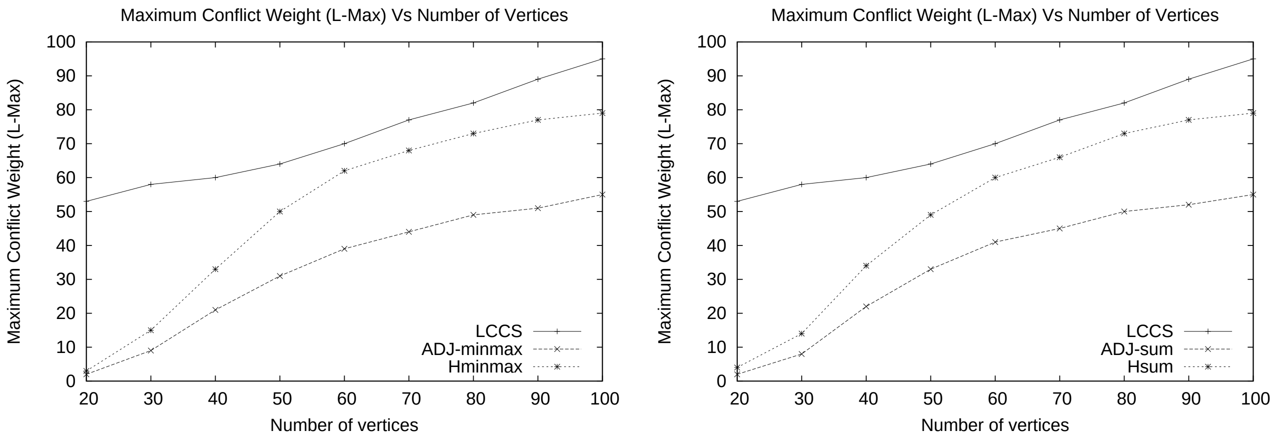
B. Using Non-overlapping channels

To simulate high interference in the wireless environment we used a large number of APs to be located in a certain physical space with same transmission and interference ranges. The number of randomly distributed APs in this case was 100, which led to a maximum degree of 20 in consequent the overlap graph. This implied that the graph is definitely colorable without conflicts using 21 colors. Our results show that *Hminmax* and *Hsum* could achieve conflict-free channel assignments to APs by using only 16 non-overlapping channels. In Figure 5 we present results comparing all the three algorithms in this environment. We can make the following observations:

1) Both *Hminmax* and *Hsum* outperform the LCCS algorithm. As the number of colors is increased from 3 to 16, the performance difference between the *Hminmax*, *Hsum* and the LCCS algorithm increases. In fact, even as the number of colors increase, LCCS does not improve as much, because it cannot detect many conflicts that exist, e.g., the Type-1 and Type-2 conflicts. Such a channel assignment cannot be obtained by performing an LCCS.
2) *Hsum* does better than *Hminmax*. This is because *Hsum* tries to minimize the L_{sum} metric in additional to L_{max} using additional coordination between APs using IAPP [6].

C. Assigning Overlapping Channels

We now present results for the channel assignment problem when we are allowed to assign partially overlapped channels. As we discussed, this flexibility enables greater spatial reuse in the wireless environment. In these simulations we assume that the number of non-overlapping channels is three, and the total number of channels is 11 (corresponding to 802.11b). For the simulations we empirically measured the I-factor, which defines the impact of assigning partially overlapping channels to neighboring APs, for all possible channel assign-



(a) Hminmax, ADJ-minmax and LCCS with L_{max}

(b) Hsum, ADJ-sum and LCCS with L_{max}

Fig. 6. *Figures shown above compare Hminmax, Hsum, ADJ-minmax, ADJ-sum and LCCS with the number of vertices. Each point is an average of simulations over 100 different graphs with the same parameters.*

ment pairs. We had presented the relative I-factor between channel 6 and all other channels in Table I.

We present the results for these experiments in Figures 6. Note that when using overlapping channel assignments, we refer to our algorithms as *ADJ-minmax* and *ADJ-sum* instead of *Hminmax* and *Hsum* respectively.

In the plots we compare the performance achieved by assigning overlapping channels to that achieved by assigning only non-overlapping channels. Clearly the performance achieved by assigning overlapping channels is expected to be better (i.e. *ADJ-minmax* is expected to achieve better performance than *Hminmax*. Figure 6(a) compares the performance of ADJ-minmax, Hminmax and LCCS. On the whole, ADJ-minmax improves the L_{max} objective function. Specifically, as the number of vertices increase, ADJ-minmax does increasingly better than *Hminmax*. The lack of non-overlapping channels causes *Hminmax* to suffer, while *ADJ-minmax* utilizes partially overlapping channels to improve the objective function.

Figure 6(b) shows the performance of *ADJ-sum Hsum* and LCCS. Like *ADJ-minmax ADJ-sum* outperforms *Hsum*, and the performance difference increases with the number of APs. Among themselves, *ADJ-sum* does better than *ADJ-minmax* with respect to L_{sum} and L_{num} and the two algorithms performs almost equally well with respect to L_{max}.

Convergence Results: Both algorithms *Hminmax* and *Hsum* converge rapidly. In particular, *Hminmax* converges in 2 *rounds* of execution of the optimization step where one round is a single execution on each AP in the topology. *Hsum* converges in 4 rounds on average.

VI. Experiments based on Testbed Measurements

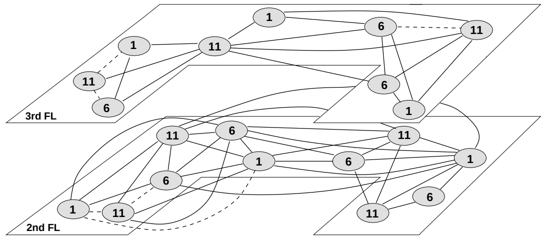


Fig. 7. Figure shows the overlap graph for the testbed network. The solid lines show the edges in the neighbor graph, and the dashed lines show the remaining edges. The number in the circle for each vertex indicates the current channel assignment in use which was performed using the LCCS algorithm.

	LCCS	Hminmax	ADJminmax
3	90	80	54
4	75	60	n/a
5	62	0	n/a

TABLE III. Table shows performance improvements achieved by Hminmax and ADJminmax over LCCS with increasing number of colors with respect to the L_{max} metric. Adjacent channel assignments using ADJminmax are applicable only with the three 802.11b non-overlapping channels.

Based on the measurements performed on an in-building testbed wireless network, we evaluate the channel assignments achieved by the LCCS and the *Hminmax* algorithms with overlapping channel re-use. Note that all physical effects of an in-building environment including path loss and multi-path effects have been captured in our measurements.

1) The Wireless Testbed: The wireless testbed network consists of 20 IEEE 802.11b APs distributed over two floors of an office building. The APs are *Soekris* boards [9] with a IEEE 802.11b *Prism II* wireless card configured as a host-based AP. All APs in the testbed operate at 100mW of transmit power. The initial channel assignment on the APs was done by searching for the least congested channel.

Figure 7 shows the overlap graph computed for the testbed network. The overlap graph was generated using the methods outlined earlier in section IV-C, namely the *Site-Report*.

2) Experiment Results: Based on the overlap graph inferred as above, we evaluate our channel assignment methods. This is compared with the current channel assignment on the testbed which based on the LCCS algorithm.

1) We first present results from assigning non-overlapping channels using *Hminmax*. In Table III, we compare *Hminmax* with the LCCS algorithm using an increasing number of non-overlapping channels. The Table shows that *Hminmax* dramatically reduces the number of users suffering interference when compared to LCCS as the number of non-overlapping channels increase (for example, with IEEE 802.11a). In particular, with 5 non-overlapping channels, *Hminmax* eliminates interference among access points while the users under the LCCS algorithm still suffer considerable interference.
2) Second we compare the performance gain using the 11 overlapping channels, when compared to using 3 non-overlapping channels. Table III shows the performance improvement over the overlap graph inferred for the testbed wireless network. In the table, *ADJ-minmax* refers to the execution of *Hminmax* with 11 overlapping channels. It can be seen that *Hminmax*, which uses 3 non-overlapping channels, reduces the number of users under interference by 11%$(= 1 - (80/90))$. *ADJ-minmax*, which uses all the 11 channels reduces the number of users under interference by 40% over the LCCS channel assignment, and by 32.5% over the channel assignment obtained from *Hminmax*. The drastic improvements stem from the fact that since the topology is fairly dense in terms of the per-AP interference, the *ADJ-minmax* algorithm uses partially overlapping channels to reduce the number of users affected.

The testbed wireless network considered here is a deployed and fully operational in-building wireless network and our improvements give an indication of the practical applicability and usefulness of the channel assignment methods discussed in this paper.

VII. Related Work

In [10], Lee et. al. provide a Linear Programming (LP) based formulation of the problem of assigning channels and AP locations using a set of *demand* points in a wireless network. The approach is centralized in nature, assumes a cooperative environment, and most importantly does not capture the dynamic nature of the channel conditions.

Channel assignment in cellular networks is a well studied problem [11]. The *cells* in a cellular network have very different characteristics when compared to IEEE 802.11 APs. Each cell has a relatively large coverage area, and a high powered base station is used to connect the cellular phones. The cells are also organized in a very regular fashion and the coverage areas are very symmetrical unlike indoor environments.

Because of these characteristics, studies such as [12], [13], focus on centralized optimization schemes such as a mixed linear integer programming based model. These centralized schemes work well in cellular networks as the channel assignment is computed once and changes rarely. Because of such fundamental differences, these techniques cannot be applied to the problem of channel assignment in WLANs.

Prior work in [14], [15] engineer MAC level extensions to take advantage of frequency diversity for throughput improvements. Channel assignment in mesh and adhoc wireless networks is typically modeled as an edge-coloring problem. This has received considerable recent attention [16], [17], [18], [19], [20]. Also there has been significant commercial interest in developing multichannel radios and corresponding higher-layer techniques to enable communication over multiple non-overlapping channels [21].

Graph coloring for general graphs is NP-hard , and it is also NP-hard to find a constant approximation. To the best of our knowledge, the weighted version of the coloring problem as described in this paper, has not been studied earlier in an algorithmic manner, although theoretic formulations of the edge-weighted coloring problem can be found in [22].

Channel assignment has also been modeled as a conflict-free coloring problem. [23], [24] provide centralized approximation algorithms for conflict free coloring. These algorithms can potentially be used for computing coloring centrally, for example, in a cellular network, but the algorithms are too complex to be implementable in a distributed manner.

In [25], Heterniemi and Jacobs present a distributed fault tolerant algorithm to $(\Delta + 1)$ color a general graph. Their algorithm also uses local information, and is scalable in nature. However, they do not assume any limit on the number of available colors, nor do they consider any weighted versions of coloring problem.

VIII. Conclusions

In this paper, we formulated channel assignment in WLANs as a weighted vertex coloring problem. We proposed two efficient, scalable and fault tolerant distributed algorithms that achieve significantly better performance than the state-of-the-art *Least Congested Channel Search* (LCCS). Through simulations, we showed that the two techniques achieve up to 45.5% and 56 % reduction in interference for sparse and dense topologies respectively with 3 non-overlapping channels. We also show that the techniques effectively use partially overlapping channels to achieve an additional 42% reduction on average for moderately sized networks. We validated these results using experiments on a fully operational in-building wireless testbed network comprising of 20 APs and achieved a 40% reduction using partially overlapping channels.

A straightforward extension to this work is to handle co-existing 802.11b/g APs in the same area of coverage. The overlap graph in such scenarios becomes directed in nature as the interference effects become asymmetric (802.11g APs would be more affected than 802.11b). The weights on the edges would reflect a measure of the asymmetric effect of the interference caused by one AP's BSS to another. We leave such extensions as future work.

References

[1] Jim Geier, "Assigning 802.11b access point channels," *Wi-Fi Planet*.

[2] P. Karn, "Maca — a new channel access method for packet radio," in *Proceedings of ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990.

[3] V. Bhargavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: A media access protocol for wirelesss lans," in *Proceedings of ACM Sigcomm*, 1994.

[4] AustWireless, "350 series ap w/captured diversity antennas and 128-bit wep," *AustWireless.com*.

[5] Vijay Vazirani, *Approximation Algorithms*, Springer Verlag, 2001.

[6] IEEE, "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," *IEEE Standard 802.1f*, July 2003.

[7] Ishwar Ramani and Stefan Savage, "Syncscan:practical fast handoff for 802.11 infrastructure networks," in *Proceedings of IEEE Infocom*, 2005.

[8] IEEE, "Lan man standards of the ieee computer society. wireless lan medium access control (mac) and physical layer(phy) specifications: Specification for radio resource measurement," *IEEE Draft 802.11K*, 2003.

[9] "Soekris Engineering," URL: http://www.soekris.com.

[10] Youngseok Lee, Kyoungae Kim, and Yanghee Choi, "Optimization of ap placement and channel assignment in wireless lans," in *Proceedings of 27th Annual IEEE Conference on Local Computer Networks (LCN)*, 2002.

[11] T. Rappaport, *Wireless Communications: Principle and Practice*, Prentice Hall, 1996.

[12] Bhaskar Krishnamachari, Stephen Wicker, Ramon Bejar, and Cesar Fernandez, "On the complexity of distributed self-configuration in wireless networks," *Journal of Telecommunication Systems*, 2003.

[13] Filipe Mazzini, Geraldo Mateus, and James Macgregor Smith, "Lagrangean based methods for solving large-scale cellular network design problems," *Journal of Wireless Networks*, 2003.

[14] Jungmin So and Nitin Vaidya, "Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proceedings of ACM MobiHoc*, 2004.

[15] Paramvir Bahl, Ranveer Chandra, and John Dunagan, "Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proceedings of ACM Mobicom*, 2004.

[16] Andrew Muir and J.J. Garcia-Luna-Aceves, "A channel access protocol for multihop wireless networks with multiple channels," 1998.

[17] Ashish Raniwala, Kartik Gopalan, and Tzi cker Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2004.

[18] Asis Nasipuri, Jun Zhuang, and Samir Das, "A multichannel csma mac protocol for mobile multihop networks," in *IEEE Wireless Communications and Networking Conference*, 1999.

[19] Jiandong Li, Zygmunt J. Haas, and Min Sheng, "Capacity evaluation of multi-channel multi-hop ad hoc networks," 2002.

[20] Rodrigo Garces and J.J. Garcia-Luna-Aceves, "Collision avoidance and resolution multiple access for multichannel wireless networks," 2000.

[21] "Commercial products," http://www.meshdynamics.com/ http://www.engim.com/ http://www.belairnetworks.com/.

[22] Bojan Mohar, "Circular colorings of edge-weighted graphs," 2002.

[23] Sariel Har-Peled and Shakhar Smorodinsky, "On conflict-free coloring of points and simple regions in the plane," 2003, http://www.uiuc.edu/ sariel/research/papers/02/coloring.

[24] Guy Even, Zvi Lotker, Dana Ron, and Shakhar Smorodinsky, "Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks," in *Proceedings of FOCS-2002*, 2002.

[25] Stephen T. Hedetniemi and David P. Jacobs, "Fault tolerant distributed coloring algorithms that stabilize in linear time," in *Proceedings of the IEEE IPDPS-2002 Workshop on Advances in Parallel and Distributed Computational Models*, 2002.