Johnny Appleseed: Wardriving to Reduce Interference in Chaotic Wireless Deployments

Tim Dasilva
UCLA
405 Hilgard Ave.
Los Angeles, CA 90095
tdasilva@cs.ucla.edu

Kevin Eustice
UCLA
405 Hilgard Ave.
Los Angeles, CA 90095
kfe@cs.ucla.edu

Peter Reiher
UCLA
405 Hilgard Ave.
Los Angeles, CA 90095
reiher@cs.ucla.edu

ABSTRACT

Many areas have dense deployments of 802.11 wireless access points, often with little or no planning of the best choices of channel assignments. As a result, there is often very high interference due to poor channel assignments. One contributing factor is that many access points are deployed with the absolute minimum of configuration effort, which means they are assigned to the channel the manufacturer has chosen, as a default. In many cases, such minimal effort deployments also mean that the access point uses the manufacturer-default password. Inspired by Johnny Appleseed, a 19th century American altruist who wandered the wilderness planting apple trees for the use of others, we investigate a method by which an altruistic wardriver moving through a dense wireless deployment could take advantage of such minimally configured access points. Where possible, he could use the default passwords to log into the system and change the channel assignment to better suite the surrounding environment, reducing interference for all. We examine this solution in simulation using real data gathered by wardrivers in several locations. We demonstrate that even with some conservative assumptions on the number of access points our Johnny Appleseed could alter, simple single-pass algorithms can result in a 10% reduction of total interference in a dense wireless deployment. We discuss the legal and ethical implications of the approach.

Categories and Subject Descriptors

C.2.3 [**Computer Communications Networks**]: Network Operations – *network management, public networks.*

General Terms

Management, Measurement, Performance.

Keywords

Wireless channel assignment, wardriving, chaotic wireless deployments.

MSWiM'08, October 27–31, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-235-1/08/10...$5.00.

1. INTRODUCTION

Wireless local area networks (WLANs) are becoming increasingly ubiquitous and enjoying wide adoption in a variety of locations, such as homes, offices, and public hot spots. *Chaotic wireless deployments*, where routers are unmanaged and their placement is unplanned, suffer from channel interference that can have a significantly negative impact on the mobile user's experience [2]. This problem is exacerbated by the average consumer's lack of knowledge regarding router configuration settings, such as the transmitting channel frequency and administrator login credentials; in particular, most routers broadcast on the same channel, creating even more wireless interference.

The inspiration for our solution to this problem traces back to 19th century America, when a man named John Chapman was roaming through the countryside, altruistically planting apple trees for the next generation to enjoy. He became a folklore legend known as Johnny Appleseed. Similar in spirit and motivation, a modern incarnation of Johnny Appleseed could roam the urban landscape, altruistically improving the configuration settings of wireless routers he encounters in order to decrease overall channel interference. This is made feasible by the following key insight: that a large number of routers are left with their default configuration settings, including administrator login information. Knowing this, Johnny Appleseed could selflessly (but illegally) access such routers and change their channel settings, incrementally improving the overall wireless environment.

In this paper, we demonstrate through simulation and modeling that the problem of channel interference can be quite severe in dense urban areas, but that relatively simple solutions can lead to practical improvements for the mobile user. In particular, even a solution based on limited knowledge and with a limited ability to modify router configuration settings could provide a tangible benefit to the overall environment by decreasing channel interference. This paper describes the details of our experiment and outlines our results.

2. THE JOHNNY APPLESEED APPROACH

Our solution is based on the assumption that a large enough number of wireless routers are left with their default configuration settings. This is a double-edged sword: the default channel settings bear much of the blame for creating the interference problem in the first place, but the default administrator login information allows an altruistic person to change them. In fact, most routers have default administrator usernames and passwords that are well known and widely available to the public [15]. This metaphorical Johnny Appleseed could drive through an urban area with a laptop or PDA in search of wireless networks, a practice known as *wardriving*, and look for routers left in their default state. By using local information about other nearby wireless routers, such as their transmitting channel and distance or signal strength, it is possible to find a locally optimal channel for the current network and assign it. He could then move on to the next router that fits these criteria, and continue to make iterative improvements throughout the area.

This approach raises many very serious legal and ethical questions [16], and this paper does not recommend or encourage actually implementing this idea. Instead, it is presented as an interesting and novel thought model for examining how simple solutions could improve the state of many chaotic wireless deployments.

Much research has already been done on the general problem of wireless channel assignment, often modeling it as a graph-coloring problem [1], as in this paper. The first advantage of this approach over previous work is that it can be efficacious without any changes to the existing WLAN infrastructures: no hardware or software upgrades, router communication or cooperation, and no a priori planning by the network owners or users. The second advantage is that it uses only locally available information, lending itself directly to wardriving and enabling it to scale linearly with the number of wireless routers in an area.

In theory, incremental improvements could be made to the overall interference problem by assigning a more optimal channel to each vulnerable wireless router that is found. How well does this theory translate to real world deployments, which can easily have on the order of tens of thousands of wireless routers? Could this approach reduce the overall channel interference in a chaotic wireless deployment?

3. EXPERIMENT DESIGN

3.1 Key 802.11 Characteristics

IEEE 802.11b and 802.11g both operate in the unlicensed 2.4 GHz frequency band and divide it into 14 channels, of which 1 through 11 are available in the U.S. The major disadvantage of this frequency division is that the channels are not actually independent, and instead partially overlap other nearby channels. As a result, 802.11b/g only offers three non-overlapping channels: 1, 6, and 11. Only these three channels can be used within transmitting proximity of each other without interference.

Two access points (APs) on the same channel in close proximity to each other are said to produce *co-channel interference*, whereas two APs on different channels whose center frequencies are less than 15 MHz apart produce *adjacent channel interference.*

Carrier Sense Multiple Access (CSMA) with collision avoidance is used at the MAC layer to help reduce actual channel interference, but it does so at the expense of increased latency and decreased network throughput [17]. As a result, even if no actual co-channel interference occurs between two neighboring APs, the use of CSMA to prevent that interference can itself have a detrimental effect on the network. To simplify discussions, we implicitly include the effects of CMSA's interference avoidance when referring to the negative impact of channel interference.

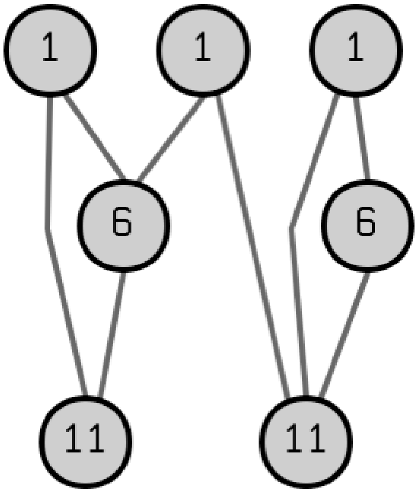
3.2 Evaluation Methodology

To evaluate the Johnny Appleseed solution, wardriving data was collected for four different locations across the United States. The data was used in simulation to describe both the current state of chaotic wireless deployments and how our solution affects it.

Channel assignment is typically modeled as a graph-coloring problem. The graph represents a wireless deployment, nodes of the graph represent wireless APs, edges represent channel interference between APs, and node colors represent the three non-overlapping channels. We extended this basic model to include all 11 channels that are available in the United States, as well as weighted edges. An edge's weight represents the most likely worst-case signal-to-noise ratio between two APs, scaled by the predicted signal strength, which decreases proportionally to the square of their distance between the APs (i.e., the length of the edge). Thus, the edge weighting allows channel interference to be taken into account, while also decreasing it realistically with distance.

The graph of a small, simplified wireless environment is shown in Figure 1. In this case, each adjacent AP is on a non-overlapping channel and thus each edge has zero weight, making this an ideal coloring for the graph.

Figure 1. Channel assignment as a graph-coloring problem.



We compared six graph-coloring algorithms in this experiment, two of which were first described in [1], and four of which were developed as alternatives and improvements. We also used several link- and node-centric metrics to help evaluate and compare the algorithms. Each algorithm had its own objective function that it was trying to minimize, such as the sum weight of every edge in the graph.

3.3 Data Sources

In order to provide a fair and realistic evaluation of our approach and the underlying channel assignment algorithms, we needed data that specified where wireless APs were located in a given area and what channels they were broadcasting on. We used the Wireless Geographic Logging Engine (WiGLE) [9], an online database of wireless AP data gathered through wardriving. The data was originally uploaded by registered users who ran commonly available wardriving tools such as Netstumbler [10] and Kismet [11]. From that data, we were able to extract the key information that was needed: router names, MAC addresses, channel settings, encryption usage, and most importantly, GPS coordinates. The raw data was filtered for duplicate MAC addresses, converted into Keyhole Markup Language (KML) [12], and visually rendered in Google Earth [13].

Other sources of wardriving data are available, but none matched the convenience, volume, and schema uniformity of WiGLE. Previous research had used data obtained from WifiMaps.com [14], but the website had been closed for months at the time of this experiment, so we were unable to compare with their results.

Edge distance was calculating using the GPS coordinates and Vincenty's formula [8], an extremely accurate geodesic calculation technique. Each pair of nodes in the graph had their distance calculated; if it fell below a certain threshold, the corresponding edge was added to the graph. For our purposes, the threshold was arbitrarily set at 100 feet. In the channel assignment algorithms, the edges are further weighted based on their distance and the channel settings of their node pairs.

3.4 Parameters

Four different parameters were used in simulation to test the Johnny Appleseed approach under a variety of circumstances.

3.4.1 Geographical Data Set

Each data set had wireless AP information for about five squares miles within the geographic location it represented. The four locations used in this experiment were: Westwood, CA; Downtown Los Angeles, CA; Cambridge, MA; and Manhattan, NY. The data sets represent a small, more-or-less random section of each location.

3.4.2 Node Set

It would be naive to assume that every wireless AP could be hacked and have its channel changed. Instead, we consider different subsets of the nodes in the graph, where an algorithm is only allowed to modify the specific subset it is given. All other nodes are left in their original channel state.

The node sets considered were: all unencrypted nodes, 100% of the nodes, 20% of the nodes, and 10% of the nodes. The observation behind using the unencrypted node set was that if a router's transmissions are not being encrypted, that router is more likely to still be on its default installation and could potentially be easily hacked. This observation was certainly too optimistic, but made for an interesting scenario to look at. The 100% node set represented the best-case scenario where every router is hackable. On the other hand, the 20% and 10% node sets were likely the most realistic, though we do not have any hard data regarding what percentage of routers in real world deployments are left on their default passwords. In order to create the 20% node set, the interference graphs were all randomized in case the WiGLE data had some sort of ordering, and then they were parsed. Every fifth node was added to the node set. Likewise, every tenth node was added to create the 10% node set.

3.4.3 Channel Assignment Algorithm

We tested several algorithms: Hrandom, Hminmax, Hminmax_adjacent, Hsum, Hsum_adjacent, and Hsum_simple. These algorithms are described in Section 4.

3.4.4 Number of Iterations

Every algorithm operated by taking global optimization steps towards minimizing a given heuristic. In each of these iterations, every node in the available node set was examined and a better channel chosen, if one existed. Algorithms could take multiple iterations to reach a stable global channel assignment, but they do, in fact, terminate. However, allowing multiple iterations does not match the constraints of the Johnny Appleseed strategy, where each router is only visited once. Thus, we looked at both single iteration and multiple iteration scenarios, to capture results from the realistic case as well as the best case.

3.5 Metrics

Three objective functions were used in [1] to evaluate channel assignment algorithms: *Lmax*, *Lavg*, and *Lsum*. The "L" prefix here stands for "link," since these metrics measure link characteristics in the graph built to analyze the wireless deployments. *Lmax* is the maximum interference of any edge in the graph, *Lavg* is the average interference over all edges in the graph, and *Lsum* is the sum of all interference observed over all edges in the graph. In the real-world deployments that we studied, there were significant numbers of edges that displayed no interference (by our definition). Since the point of the study is to determine how much we can reduce interference, including these links in the "before and after" average comparisons distorts the picture of the improvements possible, so we defined a fourth link metric, *Lavg2*, which is the average interference only for links that show any interference. Note that after any form of channel reassignment for a given set of access points, the number of links that exhibit some interference might increase or decrease, so "before" and "after" values of *Lavg2* do not necessarily incorporate exactly the same set of links.

We similarly defined four metrics that calculate interference observed by nodes, instead of the links between them. The interference on a node was calculated as the sum of all interference between that node's channel and each neighbor's channel. The metrics are mirrors of the link metrics: *Nmax*, *Navg*, *Navg2*, and *Nsum*, where the "N" stands for "node" and the meaning of the metric is the obvious node analogy for the matching link metrics.

Some of these metrics proved more interesting and useful than others, and some of them produced essentially identical results for the tests. We will, in the interests of space, only present results on the metrics that offer useful and unique insights: *Lavg2*, *Navg2*, and *Lsum*. The maximum metrics, *Lmax* and *Nmax*, never decreased as a result of channel reassignments. In the relatively heavy deployments that we studied, the maximum link interference was never decreased by any of our channel reassignments. When there is a sufficiently dense cluster of access points in a small area, no possible channel assignment will improve the interference characteristics of the worst case.

While the values of *Lavg* and *Navg* are not exactly the same as *Lavg2* and *Navg2*, the trends are similar, and they provide little additional insight. Therefore, we only report the *Lavg2* and *Navg2*.

3.6 Predicting Interference

We worked only in simulation, and did not simulate at the physical level. Thus, we did not simulate the detailed effects of changing channel assignments on interference. Instead, we used a simple model developed in [1] to predict the interference that would be experienced between two access points. While simple, this model was based on actual experiments run by those researchers. In the experiment, an 802.11b wireless transmission was made on channel

6 and received by another station on channels 1 through 11. The signal-to-noise ratio was measured for each channel, and normalized to a scale of 0 to 1, with results as shown in Table 1.

Table 1. Original SNRs, normalized to a scale of 0 to 1.

Chan	1	2	3	4	5	6	7	8	9	10	11
SNR	0	.22	.6	.72	.77	1	.96	.77	.66	.39	0

The independent nature of channels 1, 6, and 11 are evident in the results. Channels 1 and 11 receive no measurable amounts of the transmission that was made on channel 6, while every other channel receives some to varying degrees. While the table from [1] was developed based only on data gathered relative to transmissions on channel 6, we assume that the same table can be used to calculate adjacent channel interference for any channel.

One unfortunate characteristic of these results is that variations did not match on both sides of channel 6. Channel 5 has a normalized SNR of 0.77 while channel 7 has 0.96. To generalize this table and maintain the ease of verifying our own experimental results, a simplification was made, as shown in Table 2.

Table 2. Modified SNRs, normalized to a scale of 0 to 1.

Chan	1	2	3	4	5	6	7	8	9	10	11
SNR	0	.39	.66	.77	.96	1	.96	.77	.66	.39	0

In this modified SNR table, channels 2, 3, 4, and 5 now have new values, mirroring those of channels 10, 9, 8, and 7, respectively. Essentially, we chose the higher of the SNR values and applied them uniformly to both sides of channel 6. As a result, there was no longer concern for the direction of a transmission in terms of the sending and receiving channels. For example, a transmission from channel 4 to 3 would have the same SNR as one from channel 3 to 4; the direction no longer mattered. We will discuss the implications of this approximation in section 6.

The new table allowed us to predict how much of a neighbor's signal will be heard by a particular AP. While normally a high SNR is desired, because it is expressed in terms of an AP and a mobile client that wants to receive the signal, in this experiment a low SNR is instead better, because it is in terms of two APs that do not want to receive each other's interfering signal. Thus, interference occurs when one AP's signal is heard by another AP, because of their physical and channel proximity.

Lastly, recall that in order to calculate an edge's total weight, its interference is scaled by the theoretical signal strength between the two nodes. This complicated the interference calculation but added realism to the experiment.

3.7 Simplifying Assumptions

Many simplifying assumptions were made, implicitly and explicitly, in this experiment. Here we state each assumption explicitly, along with its justification.

First, we assumed that wireless interference ends after 100 feet in distance. It is well known that actual wireless interference varies widely in largely unpredictable ways that depend on many factors. We could obtain no information about any of these factors from the databases we used for the access point data. Some assumption on effective range for interference was necessary to keep the connected components of the interference graph at reasonable sizes. Since the indoor range of 802.11b/g has been quoted at 100 feet, this assumption seemed reasonable and as good as any other.

Second, there is no external interference from structures and other electronic devices, such as cordless phones and microwaves. Given the data schema available from the wardriving databases, it would be impossible to model such phenomena realistically.

Third, the metrics used are meaningful indicators of the mobile user's experience. Three of the metrics were used in [1], and results in [2] show a direct connection between interference and wireless throughput. The nature of the correlation between these metrics and throughput was not explored in this project, however.

Fourth, as in [1] and [3], we assumed that the modified SNR table is a reasonable, if crude, estimator of the real world interference that would occur. [3] showed that the theoretical and measured signal-to-noise ratios match each other very closely. This assumption is necessary because of the limited depth of information available through the wardriving data sets. Were more detailed information available, it might be feasible to run a richer simulation of the dynamic environment each AP was in; in lieu of such details, though, this assumption allows us to work with the actual (and voluminous) data available. Results from [1] involving a small wireless test bed also support this assumption.

Fifth, also as in [1] and [3], we assume that the SNR table can be generalized. The original experiment that produced the SNR table made transmissions on channel 6, with channels 1 through 11 receiving. We assume that these results can be generalized to any of the 11 channels transmitting, and that the main contributor to differing SNR values is not the absolute frequency of a given channel, but the frequency distance between it and every other channel. While external interference can have a dramatic impact upon the SNR values, such interference is usually caused by other household or offices devices that use the same 2.4GHz band and, in aggregate, are likely to negatively impact all of the wireless channels equally.

Sixth, it was assumed that the wardriving data represents a reasonable portrayal of current wireless deployments. One of the difficulties of working with wardriving data is determining when it is too stale to use. In this experiment, we used data from up to two years ago. The age of the data might somewhat limit the current applicability of the results, but it was the most recent relevant data available at the time of the experiment.

Seventh, we assume that the wardriver's GPS coordinates are reasonably close to the actual wireless AP's coordinates. When wardriving, the receiver's GPS coordinates are recorded, rather than those of the actual AP. This results in a misleading concentration of access points along the streets the wardriver travels, and reduces the accuracy of the measurements of distance between access points. Unfortunately, the wardriving databases do not contain information that would directly allow a more accurate mapping of real AP locations. Deriving the actual AP coordinates from the wardriver's coordinates would be an entire experiment unto itself, so we regard this simplifying assumption as both necessary and acceptable.

4. CHANNEL ASSIGNMENT ALGORITHMS

We examined six channel assignment algorithms in this experiment. Two of them, *Hminmax* and *Hsum*, were the original algorithms from [1] and served as foundations to build upon. The “H” prefix stands for “heuristic,” because each algorithm uses a particular objective function as a heuristic when making optimization steps (i.e., channel assignment).

Each algorithm actually has two phases of execution: initialization, which happens once; and optimization, which can happen several times.

In the initialization phase, each node in the graph is given an initial channel. This could be the original channel or a uniform number, such as channel 1. In the spirit of the Johnny Appleseed idea, though, this initialization step maintains the original channel that each node had. It would be unrealistic to expect Johnny Appleseed to make an initial pass through each neighborhood in order to set all channels to 1.

In the optimization phase, each node in the graph is considered individually. A particular heuristic is used to determine a better channel setting for a node, if there is one, using only local information that is available to that node. The one exception to this, *Hsum*, will be discussed later in this section. It is important to note that after a node makes its optimization step, the local information might change; in particular, a node's neighbors may also make optimization steps and end up on different channels than before. For this reason, it usually takes several optimization steps before arriving at a global steady state. Each optimization step is also referred to as an iteration of the algorithm.

In the following sections, each algorithm is described in more detail, along with motivations for investigating the algorithm.

4.1 Hrandom

This algorithm serves as a baseline, along with the original channel state, for gauging the effectiveness of the other algorithms. If none of the other algorithms could do better than random channel assignments, then it would not be worth spending time and effort on designing and improving them.

For each node in the graph, *Hrandom* simply assigns it a random channel between 1 and 11.

4.2 Hminmax

Hminmax was one of the original algorithms from [1]. It aims to minimize interference by minimizing an access point's local *Lmax*, which is the maximum weight of any of its edges. The algorithm picks any channel that would yield a minimum local *Lmax*.

The major flaw in this algorithm is that it fails to take into account the effects of overlapping channel interference. Consider a scenario where three nodes all have edges with each other, and two of them are already on channels 1 and 6. The optimal channel assignment for the third node is 11 because all three channels would be non-overlapping, producing no interference. However, *Hminmax* will see that channels 1 and 6 have some non-zero weight, while channels 2 - 5 and 7 - 11 are weightless (because only direct interference is considered) and seemingly good choices. In this implementation, *Hminmax* would choose channel 2, which is clearly a poor choice but stays true to the algorithm's heuristic. For this reason, connected components with channels assigned by *Hminmax* tend to use the lower half of the channel spectrum and have an unreasonable amount of interference.

The heuristic for this algorithm, H(N, C), is the maximum weight of any edge to node N whose neighbor is on channel C. For each node in the graph, *Hminmax* assigns it to the channel that minimizes H(N, C).

4.3 Hminmax_adjacent

Hminmax_adjacent is the natural extension of *Hminmax*. The only difference is that it takes channel frequency overlap and adjacent channel interference into account in its heuristic, instead of focusing exclusively on co-channel interference. In the example from Section 4.2, where three nodes all have edges with each other and channels 1 and 6 are already taken, this new algorithm would see that channels 2 through 5 and 7 through 10 have modest, non-zero interference from channel overlap, while 11 alone remains free of such noise. In this case, it would give the third node its optimal channel assignment of 11, correcting the flaw in *Hminmax*.

The heuristic, H(N, C), is now the maximum weight of any edge to node N, *if node N were on channel C*. As before, *Hminmax_adjacent* will assign each node in the graph to the channel that minimizes H(N, C) for that node.

4.4 Hsum

Hsum, the other original algorithm from [1], is the only algorithm investigated that requires communication between a node in a graph and its neighbors. This algorithm assumes either that the nodes in each connected component are all under the same management, or that they are all willing and able to cooperate. The communication is required to learn the value of the *Lmax* metric for that particular connected component of the graph, which was used to mark channels that would do worse than *Lmax* as poor choices. The purpose was to be able minimize both the weighted edge sum and the maximum edge weight. It was unclear at first glance how much this channel marking strategy actually improves performance, as opposed to simple minimizing the weighted sum heuristic. In the Johnny Appleseed scenario that motivated this experiment, of course, it is the white hat hacker and not the APs that picks channels. As such, inter-node communication is impossible. Nonetheless, we tested this algorithm because it had been proposed in [1].

This algorithm tries to minimize the sum of direct interference that it experiences with its neighbors, limiting its choice of channel to those that are unmarked. The motivation was that while *Hminmax* may minimize the overall maximum interference, it often does so by incurring numerous minor signal interferences between neighbors. The aggregate of these minor interferences could add up quickly and have a very negative impact on the graph. Instead of exclusively trying to minimize the maximum weight of any edge, *Hsum* focuses first on minimizing the sum of weights, and then on the maximum.

Like *Hminmax*, the major flaw in this algorithm is that it did not consider the effects of overlapping channel interference.

This algorithm uses two heuristics, H(N, C) and S(N, C). H(N, C) is the maximum weight of any edge to node N whose neighbor is on channel C, while S(N, C) is the *summed* weight of all edges to node N that are on channel C. For each node in the graph, *Hsum* will mark all channels for which H(N, C) > Lmax, and pick the best unmarked channel that minimizes S(N, C). If every channel is marked, it picks whichever one simply minimizes S(N, C).

4.5 Hsum_adjacent

Like *Hminmax_adjacent*, this algorithm is a natural extension of its parent, *Hsum*. It takes channel overlap and adjacent channel interference into account in its heuristic, instead of just co-channel interference. The algorithm still uses the channel marking strategy in addition to the optimization heuristic, requiring nodes in a connected component to cooperate and communicate.

The previous heuristics are redefined slightly. H(N, C) is now the maximum weight of any edge to node N if N were on channel C, and S(N, C) is the summed weight of all edges to node N that are on channel C. Otherwise, the algorithm remains the same.

4.6 Hsum_simple

We derived an algorithm from *Hsum* that did not employ the channel marking strategy, freeing it of the requirement for communication between nodes and allowing it to stay true to the spirit of the Johnny Appleseed context. This helped us understand whether the additional overhead and requirements of *Hsum* were worth the cost. This will be discussed in Section 6.

The algorithm uses only one heuristic, S(N, C), which is the summed weight of all edges to node N that are on channel C. For each node in the graph, *Hsum_simple* assigns it to the channel that minimizes S(N, C).

5. WIRELESS DATA SETS

We tested datasets representing four different locations. Our results section will concentrate on only one of these datasets, both in interest of space and because the results were characteristically similar for all four data sets. However, in Table 3 we summarize the key characteristics of each data set.

Both Westwood and Downtown LA were relatively sparse graphs compared to the incredible density of Cambridge and Manhattan, both in terms of their nodes as well as their conflict edges. Manhattan had a maximum node degree of 261, which was simply astounding.

We will examine some important characteristics of the Westwood data set in more detail, since the performance results we present will concentrate on that data set.

Table 3. Interference graph statistics for the four data sets.

Map	Nodes	Conflict Edges	Maximum Node Degree	Average Node Degree
Cambridge, MA	26,036	337,848	142	26
Downtown LA, CA	10,372	76,431	108	15
Manhattan, NY	20,257	226,376	261	22
Westwood, CA	8,758	63,585	82	15

5.1 Node Degree Distribution

We looked at the cumulative node degree distribution in detail for Westwood and found some revealing trends (Figure 2). 49% of the nodes had degrees less than 11, meaning they could at least be given channels that produced no co-channel interference, only adjacent channel interference. But the remaining 51% of the nodes were forced to choose a channel that one of their neighbors was on; there was no alternative. Interestingly, 19% of the nodes had degrees less than 3, meaning they could be assigned channels that do not overlap at all with those of their neighbors. So a fifth of the nodes can be assigned to channels that avoid any interference; half of the nodes can be assigned without direct overlap, and the other half desperately need a channel assignment algorithm that can make the best of a bad situation.

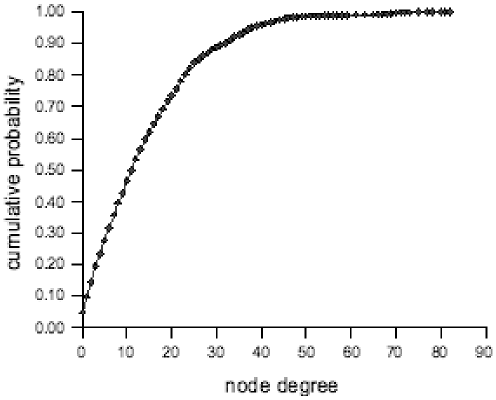


Figure 2. Westwood's node degree distribution.

5.2 Connected Component Size Distribution

Each interference graph is actually composed of many smaller connected components. Figure 3 shows the distribution of connected component sizes in the Westwood data set. Around 55% of the connected components contain a single node, which can be trivially colored. 95% of them have a size less than 14, which is still relatively small and would not be too difficult for a channel assignment algorithm to analyze. 99% of the connected components have less than 275 nodes; on the other hand, 1% of the components are very big, with the largest connected component containing 3,759 nodes; this component accounts for nearly 42% of the graph's nodes.

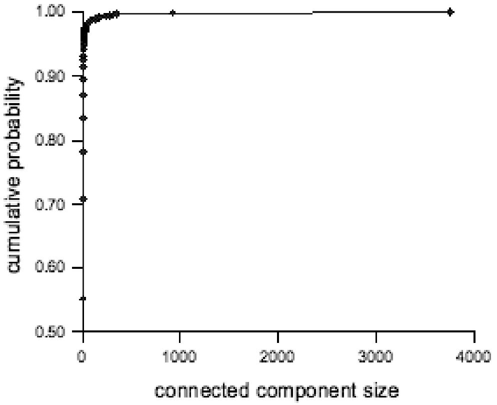


Figure 3. Westwood's connected component size distribution.

5.3 Security Issues

The underlying assumption of the Johnny Appleseed strategy is that many wireless APs still use their default settings. To gauge the extent to which this is a reasonable assumption, we looked at the security of each data set from two angles. First, how many of the APs are not using encryption? Second, how many are still using their default SSIDs? The results are shown in Table 4.

Table 4. Security statistics for four data sets tested.

Map	Unencrypted %	Default Name %
Cambridge, MA	63	22
Downtown LA, CA	49	21
Manhattan, NY	61	33
Westwood, CA	53	25

The following names were considered default router SSIDs: linksys, default, belkin54g, and netgear. We found that the majority of nodes are not using encryption, and many of them are using a default router name. Some, though not all, of the nodes whose default names and/or encryption settings have not been changed probably have also not had their default password changed, making them susceptible to Johnny Appleseed's methods. We estimate that around 10 to 20% of the nodes in these data sets probably could be hacked, though this is merely a guess.

5.4 Channel Distributions

Table 5**Error! Reference source not found.** shows the percentage of access points assigned to each channel for the four data sets tested, rounded to the nearest percent. Between 82 and 88% of the nodes are on non-overlapping channels (1, 6, and 11), but the experimental results show that the original channel assignments are far from optimal. This implies that the channel distribution is only a small part of the overall picture; more important are the details within each connected component, and how a node's channel assignment interacts with that of its neighbors.

There was extreme bias towards channel 6, the default channel for most routers on the market today, further suggesting that many nodes are likely to still be on their default installation.

Table 5. Channel distribution %s for the four data sets.

Map	1	2	3	4	5	6	7	8	9	10	11
Cambridge, MA	15	1	3	1	1	49	1	1	2	5	21
Downtown LA, CA	9	1	2	1	1	62	1	1	1	4	18
Manhattan, NY	14	1	2	1	1	53	1	1	2	8	16
Westwood, CA	10	1	2	1	1	55	1	1	4	4	21

6. PERFORMANCE RESULTS

The results presented here focus primarily on the Westwood data set, since the general trends were similar for all four data sets. We present a few summary numbers for the other three data sets at the end of this section. We present only the values for *Lavg2*, *Navg2*, and *Lsum*, as the other metrics are either similar or offer no additional insights. Results are presented for three different levels of access point alterability: all access points were alterable, only the access points with unencrypted settings were alterable (53% of all nodes, for Westwood), and only 10% of all access points were alterable. The 20% numbers are, as one would expect, somewhere between the 10% and the set of unencrypted access points, and the exact numbers do not offer information that would justify the space required to present them.

6.1 Single Iteration Results

Each algorithm was allowed to make only one optimization step in this scenario, staying true to the original Johnny Appleseed strategy where only one pass through a wireless deployment is feasible. Table 6 summarizes these results.

For each metric, the values indicate the percent of interference compared to the original situation. Since the goal is to reduce interfering signal-to-noise ratios, a percentage less than 100% indicates an improvement over the original, and greater than 100% indicates a worsening.

Table 6. Single iteration improvements for Westwood.

100%	Lavg2	Navg2	Lsum
Hrandom	74%	82%	83%
Hminmax	96%	150%	160%
Hminmax_adjacent	45%	41%	38%
Hsum	74%	80%	85%
Hsum_adjacent	46%	26%	24%
Hsum_simple	46%	26%	24%

Unencrypted	Lavg2	Navg2	Lsum
Hrandom	76%	81%	83%
Hminmax	99%	89%	90%
Hminmax_adjacent	60%	52%	50%
Hsum	78%	85%	87%
Hsum_adjacent	64%	44%	41%
Hsum_simple	64%	44%	41%

10%	Lavg2	Navg2	Lsum
Hrandom	92%	95%	95%
Hminmax	97%	87%	85%
Hminmax_adjacent	93%	87%	85%
Hsum	90%	92%	91%
Hsum_adjacent	95%	85%	83%
Hsum_simple	95%	85%	83%

Hrandom improved all three metrics in all cases, which implies that even uncomplicated heuristics or algorithms yield a small improvement using the Johnny Appleseed strategy; simply assign nodes to random channels, and most users will be better off!

Hminmax performed rather poorly. In some cases, it worsened the situation. Its poor performance can almost entirely be attributed to the algorithm's failure to consider adjacent channel interference.

Hminmax performed especially badly in the 100% node set, where it had full range to follow its flawed heuristic. Effectively, the more nodes you allow it to modify, the worse it does.

Hminmax_adjacent addressed the original algorithm's flaw with great success. By simply taking adjacent channel interference into account, this algorithm was able to consistently be among the top three.

Hsum did not do as badly as *Hminmax*, though it suffered from the same failure to consider adjacent channel interference. It performed poorly in the 10% node set, however, which is probably closer to reality than the other node sets.

Hsum_adjacent corrected the original algorithm's flaw well. The major drawback to this algorithm, as mentioned before, is that it requires nodes to communicate and cooperate in order to employ its poor channel marking scheme, which is impossible within the constraints of the Johnny Appleseed strategy.

Hsum_simple consistently matched *Hsum_adjacent*'s performance, while requiring no communication or cooperation between nodes. A close examination of the actual operation of the two algorithms revealed that *Hsum_adjacent*'s color marking strategy had only a minor influence on the channel assignment decisions, if any. The added complexity and computational overhead yielded only trivial differences.

The Johnny Appleseed approach can provide significant benefits with a single iteration, even with changes to only 10% of all access points, reducing total interference in the area by 17%. If Johnny Appleseed could break into every node, he can create an impressive 76% decrease in the total interference.

6.2 Multiple Iteration Results

A real Johnny Appleseed, traveling around to improve wireless conditions, might not be willing or able to visit a location more than once, in which case allowing the algorithms to iteratate multiple times over a node set would be unrealistic. If one assumed a different method of performing the optimization than a traveling good Samaritan, however, multiple iterations might be feasible, so it would be worthwhile to know how much benefit they give.

The algorithms we tested all stabilize after sufficient iterations. We found that the extra improvements were between minimal and non-existent. For *Hsum_simple*, the overall best strategy, there were no improvements with multiple iterations (to the nearest percentage). *Hrandom* benefited slightly from multiple iterations, but typically only reduced the metrics by one or two percentage points. From a cost/benefit standpoint, the optimal strategy for reducing interference will usually be a single iteration.

The big exception is *Hminmax*, which sometimes performs a lot better with multiple iterations. However, its performance compared to a single iteration of *Hsum_simple* is still terrible, so the difference is not of practical importance.

Table 7. Westwood channel distributions after reassignment.

1	2	3	4	5	6	7	8	9	10	11
40.6	0	0.1	0.1	0.5	28.4	0	0.2	0.1	0.1	29.8

6.3 Channel Distribution

For *Hsum_simple*, the best overall algorithm tested, the results of channel reassignment were much what one would expect. They are presented in Table 7.

Compared to the matching line (the last line) in Table 5, the assignments are much more even. Instead of 55% of all assignments being to channel 6, only 28.4% are. Few access points are assigned to any channels other than 1, 6, and 11, but some are. Channel 1 gets the most access points assigned, since the algorithm puts access points with no other nearby access points on that channel.

These results are obtained with a single iteration. With multiple iterations, the assignments even out more. For example, the assignments to channel 1 drop from 40.6% to 37.3%. Recollect, however, that this evening out had no noticeable effect on the actual interference performance characteristics that were measured.

For the other algorithms, the best performers also tended to place nodes mostly on channels 1, 6, and 11, but some of the algorithms were unaware of adjacent channel interference, and thus were less likely to avoid the intermediate channels. In particular, as would be expected, *Hrandom* scattered access points evenly across the 11 channels, with around 9% assigned to each. Thinking of it this way, it is remarkable that this clearly flawed technique still produces a noticeable performance improvement over how the problem is handled (or not handled, more precisely) today.

6.4 Summary of Data Set Results

The results for Cambridge, downtown Los Angeles, and Manhattan were characteristically similar to those for Westwood. Figure 4 compares the three metrics reported on earlier for the case of 10% changeable access points, only for the *Hsum_simple* algorithm, for the four data sets.

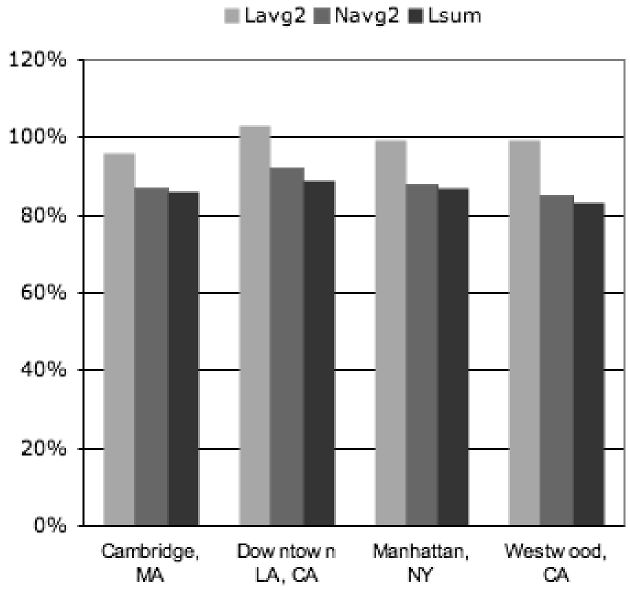


Figure 4. Key performance metrics for all four data sets.

Each location showed an average reduction in node interference of at least 8%, and an overall reduction in total interference of 11% or greater, even when only 10% of access points could be altered.

7. RELATED WORK

Channel frequency assignment has been studied extensively in the context of large cellular networks. [7] posed cellular network design as a linear mixed integer programming model that included base state location, channel assignment, and topological network design. [6] introduced as a general problem the Minimum Conflict-Free Coloring, the goal of which is to minimize the number of colors used to produce a conflict-free coloring of the graph. Adjacent channel interference was not taken into account. They developed approximation algorithms to solve this new problem through centralized means; as noted in [1], though, the algorithms were too complex to be implemented in a distribution manner.

The channel assignment problem in wireless ad hoc networks has a very different set of constraints. [5], [17], [18], [19] explored the usage of multiple channels in ad hoc networks to increase the available aggregate bandwidth. They were able to achieve dramatic increases total network throughput by giving each node multiple network interface cards and centralizing the channel assignment and bandwidth allocation algorithms. Decentralized algorithms were explored in [20], [21], [22] were similarly successful, but still relied on multiple channels per node.

Wireless LANs running in infrastructure mode began receiving more attention in [17]. It was shown that conventional frequency planning and allocation methods, previously designed for cellular networks, could not be directly applied to IEEE 802.11 wireless networks because the physical and MAC layers were coupled together. In lieu of these, the authors formulated channel assignment as an integer-programming problem and used a heuristic algorithm to solve it. Their results were very promising, but the networks they studied were comparatively small (37 and 111 nodes) and they only considered non-overlapping channel assignment. A solution based on graph coloring was first introduced in [4], where a "degree of saturation" heuristic was used on interference graphs of 10 to 20 nodes. However, no actual metrics were used to demonstrate the effectiveness of this approach, which also relied on router communication. Regardless, it was a solid stepping stone for later work such as [1], which contributed much of the ideas and innovations that this paper was built upon, such as weighted edges, channel interference signal-to-noise ratios, and specific heuristic-based algorithms and metrics.

8. LEGAL AND ETHICAL ISSUES

We have not actually performed the Johnny Appleseed approach on real wireless deployments, nor do we intend to do so. We have confined our investigations to the comfortable and safe realm of simulation, where difficult legal and ethical issues do not arise. But given that the approach we have investigated seems to provide noticeable benefits for others, should it or shouldn't it actually be tried?

From a legal point of view, in the United States, at least, it seems extremely likely that a wireless Johnny Appleseed would be behaving illegally. A number of decisions have shown that using another's wireless network without permission is illegal. Not only must Johnny Appleseed use the network to do his business, but he takes the significantly more intrusive step of actually altering configuration data on someone else's wireless access point without their permission. While the authors are not lawyers and have not consulted lawyers on this issue, it seems unlikely that such actions could be legal, regardless of the altruistic motives. Other nations might have different laws governing the use of wireless networks, but our lack of familiarity with those laws limits our ability to comment on them, other than to recommend to others to understand thoroughly the local laws before even considering actually becoming a wireless Johnny Appleseed for their environment.

Legal issues are, in many ways, easier than ethical ones. The law is a set of rules intended to be fairly rigid, with any areas of uncertainty decided by judges as needed. Ethics, on the other hand, is more personal and often situational. Would Johnny Appleseed be performing an ethical action, even if he wasn't performing a legal one?

Certainly the intent is benevolent, and even selfless. Unless Johnny Appleseed himself runs or uses a wireless network in the environs he works on, he gains no personal benefit from his actions. On the other hand, if one accepts the results of this paper, he is extremely likely to reduce wireless interference in the areas he covers, which is nearly certain to improve the experiences of the users of networks in those areas. The chances are good that the changes he makes will cause no damage to anyone, though one can never be totally certain of that.

There is an interesting parallel in the current news of network security. Researchers have discovered how to automatically remove the software that runs a large, dangerous botnet from the machines it has infected [23]. Now they are struggling with the question of whether they have the right to do so, given that removing the code from the infected machines would require them to access those machines and alter their software and configurations without permission from their owners.

In both this case and Johnny Appleseed's situation, there is a question of whether it is right to take actions for another person that will likely benefit them, but will be taken without their permission. Further, such actions might cause some unforeseen negative consequences. Merely changing the channel of a wireless access point will cause minimal (often no noticeable) interruptions or other consequences, but the general philosophy of the Johnny Appleseed approach is a fast and loose pass through an area, with no attempt to investigate conditions there deeply. Such a casual inspection could overlook important elements of the environment. On the other hand, the changes he will make are minimal and easily reversible. The general issue of the extent to which it is ethical to make even the most benevolent and targeted changes to someone else's computer and network, without their consent, deserves deeper consideration than it has received, or than it can be given here.

9. CONCLUSION

The Johnny Appleseed approach is a practical method for improving the wireless interference characteristics for an 802.11b/g wireless environment. In simulation testing, we observed an average decrease in wireless interference of 11-17%, even assuming that only 10% of the wireless access points were alterable. Remember that the methodology does not imply that 10% of the access points actually had their channel setting changed, just that only that 10% were candidates for changing. So in most cases, these performance improvements were achieved with relatively few alterations of wireless settings.

If all access points in an area could be optimized, the improvements from using fairly simple and localized algorithms to choose their channel assignments could be much more dramatic. In the Westwood simulation, the average decrease in interference was 76%, which would almost certainly result in very noticeable improvements to users' wireless experiences.

The data this study was based on came from a time when wireless access point manufacturers tended to pay little attention to their default settings, which resulted in many access points being permanently left on whatever single channel the access point manufacturer configured into all of their products. More modern access points offer automatic channel assignment, often as the default. Repeating the experiments outlined here with more up-to-date wardriving data, when such becomes available, would offer insight into whether this new technology has substantially altered the wireless channel assignment landscape and whether the Johnny Appleseed approach would still have value. One trend that would suggest that it might is that users tend to leave working access points alone, unless there is a compelling reason to either upgrade them or reconfigure them. We can thus expect a great deal of legacy hardware to persist in use for some years to come.

To conclude, we again stress that we do not recommend that any real-life wireless Johnny Appleseeds take up the methods outlined here. There are legal issues and ethical questions that have not been sufficiently well considered in the community. An exception is that organizations that have large numbers of access points deployed in their own environments could consider applying these techniques purely to their own access points, which they have complete control over. In this situation, the legal and ethical issues are not problematic, but (if no effort has been made to improve channel assignment in the organization already) relatively simple methods can yield big dividends.

In general, the lessons this paper provides are that existing wireless deployments are widely based on default channel selections; that such selections lead to unnecessarily high levels of wireless interferences; and that even very simple methods applied to a modest set of all access points can provide tangible benefits.

10. REFERENCES

[1] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted Coloring Based Channel Assignment for WLANs. In *ACM SIGMOBILE Mobile Computing and Communications Review*, July 2005.

[2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-Management in Chaotic Wireless Deployments. In *ACM MobiCom*, 2005.

[3] A. Mishra, V. Shrivastava, S. Banerjee, W. Arbaugh. Partially Overlapped Channels Not Considered Harmful. In *SIGMetrics/Performance*, 2006.

[4] P. Mahonen, J. Riihijarvi, M. Petrova. Automatic channel allocation for small wireless local area networks using graph colouring algorithm approach. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2004.

[5] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In *ACM SIGMOBILE Mobile Computing and Communications Review*, 2004.

[6] G. Even, Z. Lotker, D. Ron, and S Smorodinsky. Conflict-free colors of simple geometric regions with applications to frequency assignment in cellular networks. In *Proceedings of FOCS-2002*, 2002.

[7] F. Mazzini, G. Mateus, J. M. Smith. Lagrangean based methods for solving large-scale cellular network design problems. In *Journal of Wireless Networks*, 2002.

[8] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. In *National Geodesic Survey Review XXII*, 176, 1975.

[9] Wireless geographic logging engine (WiGLE). http://www.wigle.net.

[10] Netstumbler. http://www.netstumbler.com.

[11] Kismet. http://www.kismetwireless.net.

[12] Keyhole Markup Language (KML). http://code.google.com/apis/kml/documentation/

[13] Google Earth. http://earth.google.com.

[14] WiFiMaps. http://www.wifimaps.com.

[15] Bruce Schneier: list of default router passwords. http://www.schneier.com/blog/archives/2007/02/list_of_default.html.

[16] Memorandum from Bill Shore, FBI agent. Wireless networks: warchalking/wardriving. Available at http://www.politechbot.com/p-03884.html, 2002.

[17] K. K. Leung, B.-J. Kim. Frequency assignment for multi-cell IEEE 802.11 wireless networks. In *Proceedings of VTC*, 2003.

[18] S. Avallone, I.F. Akyildiz. A channel assignment algorithm for multi-radio wireless mesh networks. In *Proceedings of 16th International Conference on Computer Communications and Networks*, 2007.

[19] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *INFOCOM*, 2006.

[20] J. Avonts, N. Van den Wijngaert, and C. Blondia. Distributed channel allocation in multi-radio wireless mesh networks. In *International Conference on Computer Communications and Networks*, 2007.

[21] M. Shin, S. Lee, and Y.-A. Kim. Distributed channel assignment for multi-radio wireless networks. In *Mobile Adhoc and Sensor Systems*, 2006.

[22] B.-J.Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed channel assignment in multi-radio 802.11 mesh networks. In *Wireless Communications and Networking Conference*, 2007.

[23] P. Amini. Kraken botnet infiltration. http://dvlabs.tippingpoint.com/blog/2008/04/28/kraken-botnet-infiltration, April 28, 2008.