A Client-driven Approach for Channel Management in Wireless LANs

Arunesh Mishra[1] Vladimir Brik[2] Suman Banerjee[3] Aravind Srinivasan[4] William Arbaugh[5]

[145]CS Dept, University of Maryland, College Park, USA.{arunesh,srin,waa}@cs.umd.edu.
[23]CS Dept, University of Wisconsin, Madison, USA.{vladimir,suman}@cs.wisc.edu.

***Abstract*— We propose an efficient client-based approach for channel management (channel assignment and load balancing) in 802.11-based WLANs that lead to better usage of the wireless spectrum. This approach is based on a "conflict set coloring" formulation that jointly performs load balancing along with channel assignment. Such a formulation has a number of advantages. First, it explicitly captures interference effects at clients. Next, it intrinsically exposes opportunities for better channel re-use. Finally, algorithms based on this formulation do not depend on specific physical RF models and hence can be applied efficiently to a wide-range of in-building as well as outdoor scenarios.**

We have performed extensive packet-level simulations and measurements on a deployed wireless testbed of 70 APs to validate the performance of our proposed algorithms. We show that in addition to single network scenarios, the conflict set coloring formulation is well suited for channel assignment where multiple wireless networks share and contend for spectrum in the same physical space. Our results over a wide range of both simulated topologies and in-building testbed experiments indicate that our approach improves application level performance at the clients by upto three times (and atleast 50%) in comparison to current best-known techniques.

I. Introduction

Wireless LANs have seen explosive growth in recent years as a last-hop connectivity solution. They operate in the 2.4 and 5 Ghz bands where unlicensed spectrum is very limited. Due to such growth, network administrators are faced with an emerging challenge of efficiently managing bandwidth resources to provide better service to clients. In this paper, we focus on the specific problem of channel assignment to improve application throughput on a per-user basis and for the network as a whole.

Channel assignment in other domains such as cellular networks has been modeled traditionally as a vertex coloring problem. However, the irregular coverage topologies present in WLANs due to the vagaries of the indoor RF environment make the channel assignment algorithms in cellular networks inefficient when applied to WLAN scenarios[1], [2], [3].

Existing approaches to assigning channels:

Each AP operates on a *single* administrator-specified channel. The mobile client scans the wireless medium to *associate* with an AP that has a strong signal. All communication between an AP and its associated clients (which form a Basic Service Set (BSS)) occur in the channel assigned to the AP. As a basic design rule, APs within range of each other are set to different "non-overlapping" channels. Network administrators typically use multiple techniques to assign channels to APs to reduce interference between them. First they conduct detailed Radio Frequency (RF) site surveys, often using spectrum analyzers, prior to setting up APs within the building and use this information to *manually* assign specific channels to them [4]. Beyond such initial assignment, each AP continuously monitors its assigned channel for data transmissions by other APs and their clients. If the volume of traffic in that channel (from other APs or clients of other APs) is greater than a threshold, the first AP moves to a less congested channel. We call this technique for channel assignment, *Least Congested Channel Search (LCCS)*.

Consider the WLAN topology shown in Figure 1. Three APs and respective clients (client-AP associations) are indicated using directed arrows. Since the Region X shown in the figure is devoid of clients, APs AP_1 and AP_2 can be assigned the same channel to maximize spectrum re-use. However, the same optimization cannot be done with AP_2 and AP_3 since Region Y has clients interfering with each other. This distinction between the two scenarios is critical to leverage such channel re-use opportunities. Note that by capturing congestion information at the APs *alone*, it is very hard to distinguish between these two scenarios in an algorithmic manner.

Approaches such as LCCS are *AP-centric* in nature, that is, they capture interference at the APs but do not involve client participation. For the setup shown in Figure 1, LCCS will be unable to distinguish between Region X versus Region Y – precisely because the interference present in Region Y is *hidden* from the respective APs. We call this the *Hidden Interference Problem*, which is discussed in detail in Section II. In this work, we show that AP-centric approaches lack the ability to detect various similar interference scenarios which can cause serious inefficiencies in the channel utilization. Such observations provide the motivation to innovate client-centric models and techniques for channel assignment in the context of WLANs.

The end goal of this work is to improve application performance. While client-based channel assignment solves a part of the problem, load balancing of clients among APs is also needed for a complete solution. Prior work by Bejerano et. al. [5] provides a provably good centralized load balancing method which assumes that the channel assignment is performed independently. It is natural to expect that by jointly

[3] Supported in part by NSF Award CNS-0520152.

[4] Affiliated to CS and UMIACS. Supported in part by NSF Awards CCR-0208005 and CNS-0426683.

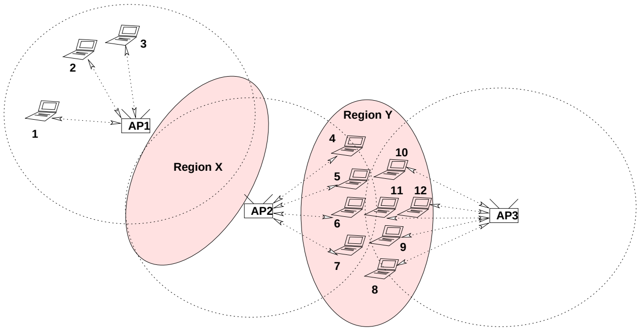


Fig. 1. Channel assignment should be based on user performance. The unshaded circles indicate the interference radius of each AP.

considering both channel assignment and the load balancing problem, significant improvements can be achieved. In this work, we provide a centralized solution to this joint problem of channel assignment and load balancing to improve application performance. Through application level metrics we show that such a joint solution has significant advantages compared to addressing the two problems independently. We refer to this problem of channel assignment with load balancing as *channel management*.

In this paper, we propose a novel client-centric model of capturing the interference constraints in a WLAN. Based on this model, we develop a centralized technique for addressing the problem of channel management. Such centralized approaches are applicable to *managed* networks in organizational settings such as airports, hotels, business offices, and centrally managed hotspots. We also extend our approach to deployments where multiple networks managed by different entities share the same physical spectrum [6].

Our Conflict Set Coloring Model

We capture the hidden interference scenarios (similar to Figure 1) by constructing a set theoretic model called *conflict set coloring*. We use the term *conflict* to denote scenarios where any two stations (APs or clients) belonging to different BSS interfere with each other by the virtue of sharing the same channel.

Let (X, C) denote a wireless LAN, with X as the set of APs, and C as the set of clients. Each client $c \in C$ is represented as a tuple $\langle r_c, i_c \rangle$ consisting of two sets: (i) a *range* set r_c, which consists of all APs in communication range, i.e., all APs to which a client can associate to and obtain service; and (ii) an *interference* set i_c, which consists of all APs within one-hop range of the *AP-client link* : that is all APs that are within range of any station (AP or client) that is in direct range of the AP or client under consideration. Let $T = \{t_c = \langle r_c, i_c \rangle | \forall \text{ clients } c\}$ be the set of tuples for all clients. We call (X, T) a *Conflict (CF) Set System*.

Two entities that seek to communicate data wirelessly can suffer interference at either points[7]. In WLANs, one of these entities is an AP and the other a client. Figure 2 presents the concepts of range and interference sets pictorially. The data link of interest is $AP_0 - C_0$. There are two possible cases of interference each at AP_0 and C_0 as shown. When an AP interferes with a client (such as AP_2 interfering with C_0), that AP becomes a part of the range set of the client. This is because the client could potentially obtain network service from the interfering AP. Thus, in Figure 2, the range set of C_0 contains AP_0 and AP_2. For all other cases, the AP AP_x representing the *link* $AP_x - C_x$ that interferes with $AP_0 - C_0$ becomes a part of the interference set of C_0. For example, the link $AP_1 - C_1$ interferes with $AP_0 - C_0$ at C_0. AP_1 is inserted into the interference set of C_0. Thus, the interference set of C_0 comprises of $\{AP_1, AP_3, AP_4\}$.

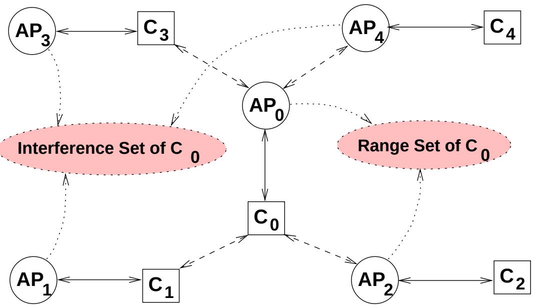


Fig. 2. Four interference situations and how the *range* and *interference* set constructs capture them.

The goal of channel management based on conflict set coloring is to assign channels/colors in such a way that each client is assigned to APs (chosen from the range set) which suffer from minimum conflict (or are conflict *free* if possible). We propose a centralized algorithm called *CFAssign-RaC* (stands for conflict set color assignment using Randomized Compaction) which addresses the joint problem of channel management. We describe some of the key advantages of using such an approach for channel management in WLANs:

Client-driven approach: We call a channel management algorithm client-driven if it aims to minimize interference or conflicts *at wireless clients* apart from the APs. Our proposed *CFAssign-RaC* algorithm implicitly models the location and distribution of wireless clients with respect to the APs while making channel assignment and load balancing decisions in order to meet the minimization objective. We demonstrate that such a client-driven approach leads to more efficient channel management at APs that reduce interference for *wireless clients.* We elaborate more on this model in Section IV. Existing approaches (LCCS) and other potential alternative approaches (using the vertex coloring model) capture the interference at APs instead of the interference at clients and hence perform poorly (see Section II and III). Client-driven approaches have been used for various optimizations in WLANs [8], [9]. However, this the first such approach to address the joint problem of channel management.

Joint channel assignment and load balancing: In order to achieve the best application performance, the problem of channel assignment should not be studied in isolation from the problem of client-AP association (load balancing). In current WLAN systems, these two problems are addressed independent of each other as follows. First, APs are assigned to different channels based on techniques like RF site survey and

LCCS. Subsequently, each client *independently* identifies an AP in its vicinity with good signal strength and associates with it. Or, existing load balancing techniques such as [5] can be used to distribute client load among the APs. We show in this paper that more efficient use of wireless channels is possible when we consider the channel assignment problem in tandem with the problem of load balancing client-AP associations, i.e. performing channel management. The conflict set coloring approach of the *CFAssign-RaC* algorithm implicitly couples and simultaneously solves both these problems for reduced interference at wireless clients.

Dynamic channel re-use and discovery of hidden-APs: Efficient channel re-use is an important requirement of all potential algorithms. Our proposed solutions dynamically identify opportunities for channel reuse when overlap regions among APs (such as among AP_1 and AP_2 in Figure 1) are devoid of mobile clients, but quickly revert back to a re-assignment of channels as the overlap regions become populated with clients. (We ignore small timescale, transient migrations of user populations.)

Independence from RF propagation models: The conflict set coloring model meets this objective of propagation model independence by using empirical samples of the clients' experience of interference to make decisions on channel assignment (instead of using properties of radio propagation to "infer" interference). Such client participation exposes a more accurate view of client interference and enables the algorithms to make better channel assignment decisions.

We also present a detailed evaluation of the *CFAssign-RaC* algorithm based on conflict set coloring, using both extensive *packet-level simulations* that evaluate the impact on application level metrics *as well as measurements* performed on a large testbed consisting of 70 APs distributed over four floors of an office building. Our results indicate that the proposed techniques lead to significantly lower interference at wireless clients. Our simulations show that for various topologies (dense and sparse) our approach results in upto three times (and atleast 50%) improved throughputs at the application-level. Based on measurements performed over an in-building wireless network, we observed that the *CFAssign-RaC* algorithm always found the optimal solution in practice and thus brought about significant reduction in client interference over the LCCS approach.

Roadmap: The rest of the paper is structured as follows: We first discuss limitations of existing AP-centric approaches such as LCCS (Section II) and vertex coloring when applied to WLANs (Section III). We then present our conflict set coloring model (Section IV) and discuss the *CFAssign-RaC* algorithm in detail (Section V). Next, we incorporate load balancing into the *CFAssign-RaC* algorithm (Section VI). We evaluate our approach through extensive simulations (Section VII) and measurements over a deployed in-building wireless network (Section VIII). We discuss related work in Section IX and summarize in Section X.

II. Existing Approaches and Limitations

The state-of-the-art method [4] for channel assignment is to perform a Least Congested Channel Search (LCCS). There exist approaches for load balancing after a channel assignment has been computed [5]. We first analytically model LCCS and discuss its limitations. Next, we discuss some of the inherent shortcomings of using any existing load balancing technique *after* performing a channel assignment.

Limitations of LCCS - The Hidden Interference Problem

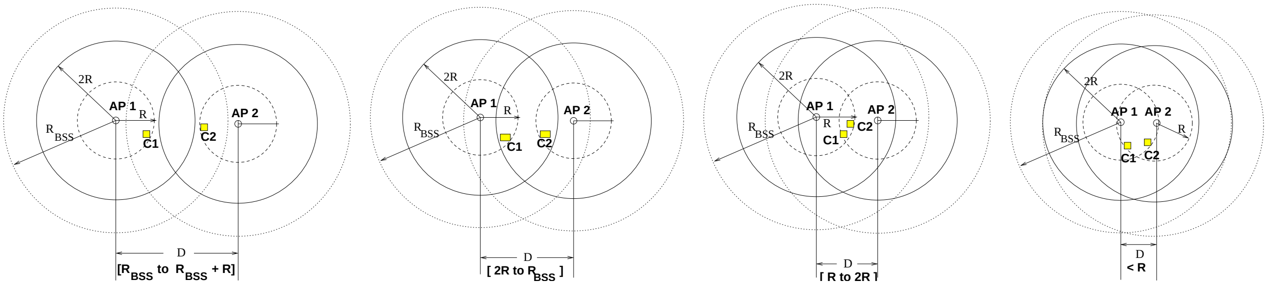
In LCCS, an AP, on detecting interference from other APs or clients associated to other APs, searches for a "less-congested" channel of operation on a periodic basis. We now show that there are many potential scenarios in which an AP using the LCCS algorithm is unable detect certain scenarios of interference with neighboring APs and clients.

Figure 3 shows four different *conflict* scenarios that can arise between two neighboring APs that operate on the same channel. We define the term *conflict* to denote scenarios where any two stations (APs or clients) belonging to different BSS interfere with each other by the virtue of sharing the same channel. In Figure 3, the innermost circle around an AP indicates its communication radius. The circle of radius 2R around an AP indicates the region where transmission from clients associated to this AP could potentially interfere with any station.

The outermost circle denoted by R_{BSS} ($\geq 2R$) indicates the size of an AP's BSS. This is maximum distance from the AP upto which transmission from the AP or clients associated to this AP would trigger carrier sense at a receiver – however, the signal would not be strong enough for data packet to be received without errors. Optimizations such as [8], [6] can mitigate the effect due to carrier-sense based interference and can be used in conjunction with the techniques discussed in this paper. Henceforth, the term *interference refers to data transmissions unintended for a recipient*. Thus, a transmission from a node X intended for Y is labeled as interference to all other nodes that receive the packet.

In Figure 3 assume clients C_1 and C_2 are associated with APs AP_1 and AP_2 respectively. We distinguish between the following scenarios:

- **No conflict:** In this scenario (not shown for brevity) the two APs are separated by a distance greater than $R_{BSS}+R$. This is the minimum distance that allows two APs to use the same channel without causing interference *independent of the client distribution*.
- **Type-1 conflict:** In this case (Figure 3(a)) the two APs are separated by a distance between R_{BSS} and $R_{BSS}+R$. This implies that transmissions of clients C_1 and C_2 will cause elevated signal levels to trigger carrier sense at each other. However, they will be unable to receive the transmissions correctly to determine the source. Techniques outlined in [8], [6] can be used in conjunction with the model and algorithms presented in this paper to mitigate the interference due to such scenarios.



(a) Type-1 Conflict (b) Type-2 Conflict (c) Type-3 Conflict (d) Type-4 Conflict

Fig. 3. Figure shows four different *conflict* situations that can arise with two APs. Shown for each AP is its communication radius (R) and a region of radius 2R where transmissions from associated clients would interfere.

- **Type-2 conflict:** In this case (Figure 3(b)) the distance between the two APs is such that $2R < D \leq R_{BSS}$. In this case, the clients C_1 and C_2 interfere with each other. This interference will appear as increased signal level at the APs. However, they will be unable to determine the source of such interference. Only the clients themselves can deterministically capture such scenarios of interference which an AP-centric technique such as LCCS will fail to recognize.
- **Type-3 conflict:** In this case (Figure 3(c)) the distance between the two APs is between R and $2R$. Depending on the position of C_2, AP_1 might or might not be able to detect this interference. This applies to C_1 and AP_2 as well.
- **Type-4 conflict:** In this case (Figure 3(d)) the distance between the two APs is less than R. Such interference can be detected at the APs in a deterministic manner regardless of client positions. This is because each AP can receive data packets transmitted by the other and infer this interference.

Among these four scenarios, the two APs should be allocated the same channel only in the No-conflict case and possibly the Type-1 Conflict case. In the other three cases, the two APs should be assigned to different non-overlapping channels, if possible. Note that AP-centric approaches such as LCCS can only detect Type-4 conflicts deterministically. Also the frequency of Type-2 and Type-3 conflicts is expected to be quite high. Both simple geometric analysis (72% of all conflicts assuming a uniform distribution of APs and homogeneous transmission radii) as well as real-life measurements taken on our operational WLAN testbed (in Section VIII) demonstrate this.

The interference present in Type-2 and Type-3 conflicts outlined above is *hidden* from the APs. The APs are cognizant of the presence of clients, however, without client participation they cannot infer the presence of such interference. Hence, in parallel with the *Hidden Terminal Problem* of wireless data networks, we term such a scenario as the *Hidden Interference Problem* of WLANs. Thus client feedback is seen to be vital in making channel decisions.

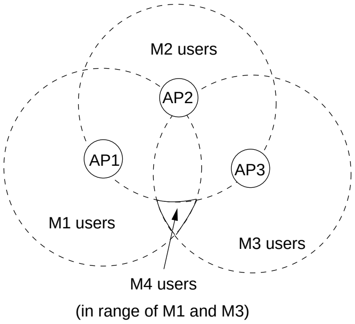


Fig. 4. Figure shows an example topology where LCCS with optimal load balancing produces an unfair solution compared to *CFAssign-RaC*.

Load Balancing after LCCS

Once a channel assignment is known, it is possible to improve user performance by carefully balancing load across APs. Given a channel assignment, one of the best known techniques for load balancing is given by Bejerano et. al. in [5]. It provides a Linear Programming (LP) based centralized scheme for fairness and load balancing in a WLAN. The approach is well suited for scenarios where the channel assignment strategy is fixed and static. In our work, we address WLAN scenarios where the channel assignment can be controlled either in a centralized manner. It can be seen by constructing some examples that a joint optimization of the channel assignment and load balancing will perform better with regard to end user throughput. We present a detailed discussion in a technical report [10].

III. A Vertex Coloring based Approach and its Limitations

Based on the discussion in the previous section, it may appear natural to model the channel assignment problem in WLANs as a vertex coloring problem cognizant of client interference. Vertex coloring approaches have been successfully applied to frequency assignment in cellular networks [11], [12], [13].

In our prior work [14], we have explored specific implementations of this vertex coloring based channel assignment in WLANs [10]. Due to space constraints we do not report on specific details of this approach. Instead we focus on specific shortcomings of such an approach which form the intuition behind the conflict set formulation that we propose. We augment this discussion with a simulation based evaluation of the vertex coloring approach in Section VII.

Consider Figure 5(a) which shows four APs: $AP_{\{1...4\}}$. Assume that none of these APs interfere with each other. Clients C_1, C_2, C_3, C_4 are in direct communication range of APs AP_1, AP_2, AP_3, AP_4 respectively and no other APs. Therefore, C_1 associates with AP_1, C_2 associates with AP_2, and so on. Client C_5 is in direct communication range of all the four APs. Without loss of generality, let us assume that C_5 associates with AP_1. Figure 5(b) shows the corresponding conflict graph, which is a 4-clique. Note that this graph is cognizant of interference suffered at the clients. There is an edge between each pair APs since client C_5 is in communication range of both APs. Therefore, based on the vertex coloring approach, each AP will be assigned a distinct color (channel) and we will need four different non-overlapping channels to guarantee conflict freedom.

However, in reality only two channels are sufficient to guarantee conflict freedom in this example. For example, we can assign AP_1 (and its clients C_1, C_5) to one channel and all other APs to the same second channel. Note that AP_2, AP_3, AP_4 and their clients do not interfere with each other.

Such a solution is not realizable in the vertex coloring approach because the graph model fundamentally lacks client representation. We next describe our conflict set coloring formulation where such an assignment can be achieved.

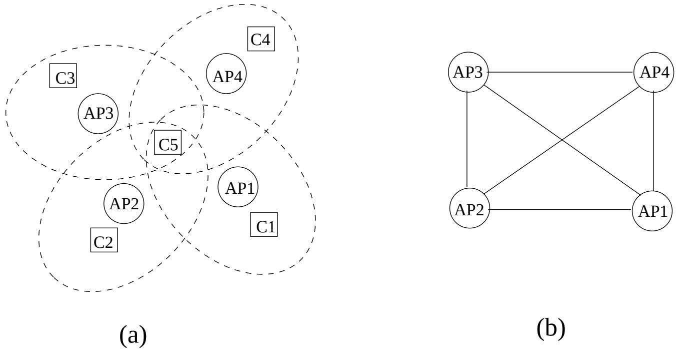


Fig. 5. (a) A WLAN example and (b) the graph created by the vertex coloring formulation.

IV. Conflict Set Coloring Formulation

We formulate the problem of channel management in wireless LANs as a *conflict (CF) set coloring problem*. Consider the example shown in Figure 5. Clients $C_1 \dots C_4$ are assigned to $AP_1 \dots AP_4$ respectively. Client C_5 is in range of all four APs: $\{AP_1, AP_2, AP_3, AP_4\}$. Given this set, we need a coloring of the APs such that this set has at least one AP with a *unique* color, i.e., that color is not used by another AP within the set. The client hence, associates to the AP with the unique color. This can be trivially achieved if we select exactly one AP and assign it a particular color (associate client C_5 to it), and then assign different colors to the rest. For example, AP_1 could be assigned color 1 while AP_2, AP_3, AP_4 could be assigned color 2.

Figure 2 shows the four logical ways in which an AP-client link can suffer conflicts at either ends. A client is said to be *conflict-free* if its association with an AP on the assigned channel eliminates conflicts at both the AP and the client. If there does not exist such an AP, the client then associates to the AP such that the AP-client link has *minimum conflict* – where conflict on a particular channel can be measured as the number of APs that share the channel. In Section VI we discuss the exact method of measuring the total conflict on a channel. The goal of channel management over this conflict set system is to assign channels to APs in such a way that it minimizes the conflict for each client. This solution also yields an association mapping of clients to APs, where a client associates to the AP that has the minimum conflict.

We formalize this notion of conflict set coloring as follows: Each client is represented as a tuple of two sets $\langle r, i\rangle$: a *range* set r and an *interference* set i.

Range set of a client: For each client we define its range set as the set of all APs such that the client lies within the communication range of each such AP, regardless of the current channel of operation of the APs. In Figure 5, the range set for C_5 is $\{AP_1, AP_2, AP_3, AP_4\}$. Note that a client can compute its range set empirically by monitoring APs in its vicinity and has to associate to one of such APs to obtain network service.

Interference set of a client: The range set of a client captures some of the interference experienced by the client, but does not capture the total interference observed by the client-AP link. A client can suffer additional interference from clients of other APs, if the client is within the transmission range of such clients. Note that if the client is within the transmission range of such an AP, then the AP will be in the range set. On the other hand, if the client is outside the transmission range of such an AP, the latter becomes a part of the interference set. Also if two APs are within transmission range of each other, clients associated to one such AP would suffer interference from the other. Such APs are also a part of the interference set.

An AP a is a member of the *interference set* of a client c if (i) a is not a member of the range set of c and, (ii) a is within communication range of some station x (AP or client) and x is either the client c or an AP within range of c. That is, the AP a is within one-hop range of the client-AP link. Figure 2 illustrates these concepts pictorially.

Scenarios of Interference: Figure 6 shows the various conflict scenarios. An edge indicates that the corresponding stations are within communication range of each other. We discuss how the conflict set system captures each of these scenarios:

(i) In Figure 6(A), C_1 is within range of AP_1 and AP_2. This is captured by the notion of range sets which consists of all APs that are in communication range of a client. Channel assignment algorithms based on this conflict set system will hence assign different channels to AP_1 and AP_2.

(ii) In Figure 6(B), AP_1 and AP_2 interfere with each other.

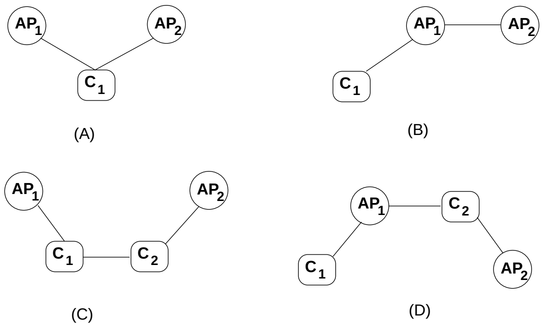


Fig. 6. Various interference scenarios.

Such interference can be detected by the APs themselves. In our conflict set model, this inter-AP interference is represented as interference suffered by clients as this affects the client-throughput. For example, the interference between AP_1 and AP_2 in Figure 6(B) effects the throughput for client C_1 and hence, AP_2 is a member of the interference set of client C_1. Note that such a client-based representation of inter-AP interference is beneficial as (A) it assigns different channels to the interfering APs (if possible) to reduce the effect of such interference on clients associated to such APs, and (B) in the degenerate case (not shown in figure) that both APs are not serving *any* clients, they are assigned the same channel (to maximize channel re-use), thus permitting improved channel assignments to other neighboring APs.

(iii) In Figure 6(C), the APs are out of range of each other. However, the clients could interfere with each other if they are associated to the respective APs as shown. This is handled by making AP_1 a member of the interference set of C_2 and vice versa. In order to minimize the conflicts, our formulation would thus assign different channels to these APs, if possible.

(iv) In Figure 6(D), AP_1 and AP_2 are in the range set of C_2. Also AP_2 is in the interference set of C_1. As before, our formulation would assign different channels to AP_1 and AP_2 if possible.

Some advantages of conflict set coloring formulation: There are multiple advantages of using the conflict set coloring formulation to obtain an efficient channel assignment in WLANs. First, this formulation directly captures the effect of interference at the clients using the range and the interference sets. Thus, algorithms that provide solutions to this problem aim to directly reduce the metric of interest — conflict at the wireless clients. Therefore we call this a *client-driven* approach. This is a key difference from the vertex coloring approach (Section III) where the vertices to be colored represent the APs and the edges indirectly account for the interference on clients due to the assignment of same color to two neighboring APs.

Next, the conflict set coloring formulation intrinsically captures opportunities for channel re-use. In the example of Figure 5, we would need 2 channels for the minimum conflict assignment (which made all clients conflict-free). Now if client C_5 were to become inactive (or moved to a completely different location), all APs could use the same channel. This allows other neighboring APs to use the second channel for improved throughputs. By keeping the model updated on a periodic yet *coarse-grained* basis, we will be able to neglect fine-grained user migrations, and capture medium and large-scale variations of client distributions.

Finally, the conflict set coloring formulation captures effects of interference through sets (range and interference sets) instead of physical RF models, e.g., two-ray pathloss model. Therefore algorithms that provide solutions to this conflict set formulation will be efficient *irrespective of the underlying physical RF properties of the wireless environment*. This advantage is possible because we *sample* the interference constraints directly at the clients rather than *infer* such constraints using properties of radio propagation. Thus, such an approach is applicable to indoor environments which are challenging to model from an RF perspective.

Conflict Minimization Objective: A client in conflict can suffer from drastic reduction in throughput — the reduction factor can be non-linear in the total number of stations (clients or APs) in conflict with this client [15]. The objective of the conflict set coloring problem is to minimize the conflict suffered by each client. As a special case, such an objective function also maximizes the number of clients that are conflict-free. When considering application throughput as the end goal, it is important to optimize an objective function which is aware of fairness and load balancing issues while performing channel assignment as discussed earlier in Section VI.

For simplicity of presentation, we first consider the objective of *maximizing* the number of clients that are conflict free. Below we discuss a formal representation of the conflict set coloring problem based on conflict-freedom maximization. We present the coloring algorithm based on this objective function in Section V. Later in Section VI, we propose a conflict minimization objective function which incorporates load balancing and fairness issues and adapt the *CFAssign-RaC* algorithm to incorporate this objective function.

Notations and Definitions

Let (X, C) denote a wireless network, where X is the set of all APs and C is the set of all clients. For each client $c \in C$, we associate a tuple $t_c = \langle r_c, i_c \rangle$ where $r_c \in 2^X$ (2^X denotes the power set of X) is the range set for c and $i_c \in 2^X$ is the interference set for c. Let $T = \{t_c = \langle r_c, i_c \rangle | \forall \text{ clients } c\}$. We call (X, T) constructed in the above manner, a conflict set system (or simply, a set system) for the network (X, C). Let $\theta : X \rightarrow \{1 \ldots k\}$ be the color/channel assignment using k colors, for a set system (X, T).

Property of Conflict-freedom: For a client represented by $t_c = \langle r_c, i_c \rangle \in T$, define $z_j = \{x \in r_c \cup i_c : \theta(x) = j\}$, then $\exists j \in \{1 \ldots k\}$, such that (i) $|z_j| = 1$ and (ii) let $z_j = \{x_j\}$, then $x_j \in r_c$. In other words:
Assignment of colors (θ) to the APs is in such a way that for the client (c) there is at least one AP (a) in the range set of c which is assigned a color, j, and no other AP in the range set or interference set of c has been assigned to this same color,

j. We shall refer to this property as the conflict-free coloring property, and we say the client c is conflict-free.

A solution to conflict set coloring also implicitly defines an association mapping for the client, i.e., the client will associate to the AP which holds the conflict-free color in its range set. Thus, the solution provides the following:

1) θ gives the channel assignment for the APs.
2) For $t_c \in T$, define $\gamma(c) = x$ such that color of x is conflict free in t_c. (x is the AP that leads to conflict-freedom for this client.) For clients that are not conflict-free, x is the AP that suffers from minimum conflict out of all APs in the range set of c. $\gamma(c)$ provides the association mapping for all clients.

It has been shown that, in general, the conflict set coloring problem is not easier than vertex coloring, and is hard to approximate [16]. In the next section, we define efficient strategies that maximize the number of conflict-free clients in the set system.

V. Randomized Compaction

We describe a randomized algorithm for conflict set coloring with the conflict-freedom as the objective function. The algorithm, called *CFAssign-RaC* (*CFAssign* using "randomized compaction") works in a centralized manner and is particularly suited for centrally managed wireless networks with multiple APs, as is typical in most organizations, airports, hotels, etc. By using the ability to detect and capture different types of conflicts (Section II) and by taking advantage of the conflict set coloring formulation which captures opportunities for channel re-use, the centralized *CFAssign-RaC* algorithm performs better than LCCS and other AP-centric approaches.

Algorithm 1 *CFAssign-RaC* (X, T, k)

```
X = set of access points,
T = set of ⟨range,interference⟩ tuples for each client,
$k$ = number of colors
$\theta : X \rightarrow \{1 \dots k\}$ is the returned channel assignment

 1: $X'$ be a random permutation of $X$.
 2: Let $X' = \{x_1, x_2, \dots, x_i\}$.
 3: Set $\forall x \in X, \theta(x) = -1$ /* indicates an unassigned AP */
 4: while true do
 5:     $ncf \leftarrow$ Num_Conflict_Free$(T, \theta)$
 6:     for $i = 1 \dots |X|$ do
 7:         $\theta(x_i) \leftarrow$ Compaction_Step $(x_i, \theta, T, k)$
 8:     end for
 9:     if Num_Conflict_Free$(T, \theta) = ncf$ then
10:         stop
11:     end if
12: end while
```

Conceptually the *CFAssign-RaC* algorithm (described in Algorithm 1) progressively choses the 'best' color (channel) for an AP that maximizes the number of clients that are conflict-free. We first describe a *compaction* step which can be applied to any existing color assignment to increase conflict-freedom among clients, and later describe the algorithm using this.

Compaction Step: Consider an AP $ap \in X$. Keeping all other color assignments the same, the compaction step assigns a color to ap which maximizes the number of conflict-free clients overall for the set system (X, T). Such a color is chosen as the new assignment for ap. Note that the number of conflict-free clients can be easily computed using the interference sets available at the central entity. Step 7 shown in Algorithm 1 corresponds to this compaction.

In Step 6, we calculate the value of this objective function, which is the number of conflict free clients. The channel assignment for AP ap is changed only if it improves the objective function value. In Step 7, if the objective function value stays the same after applying the compaction step (Step 6), the algorithm terminates.

The *CFAssign-RaC* algorithm operates by repeatedly invoking the compaction step for each AP in succession. The order of invocation is randomized by using a random permutation of the APs (Step 2). The entire compaction process (Steps 6-8) is repeated till the objective function (number of conflict-free clients) stops improving. Note that the objective function is a discrete value, and is lower bounded (by zero). Thus, after a single executing of the compaction process (Steps 6-8), the algorithm either improves the objective function or terminates (Step 9). Thus, the algorithm will provably terminate.

Because of the hardness of this problem [16], we invoke the *CFAssign-RaC* algorithm multiple times with different random permutations and obtained the best solution across these runs. Once an AP-permutation is fixed, *CFAssign-RaC* uses the compaction step to iteratively refine the solution based on the objective function. By invoking this algorithm multiple times, we perform a randomized search with iterative refinement over the solution space. This increases the chances of converging to a better optima and possibly the global optimum over multiple executions.

Implementation Issues

The *CFAssign-RaC* algorithm needs an accurately constructed conflict set coloring formulation (X, T). APs can find out the range and interference sets of their clients by requesting the latter to conduct a *site-report* as specified in IEEE 802.11K drafts [17]. In a site-report, a client scans all channels and reports all the APs within its range on the different channels. Such scans can be requested periodically or dynamically based on mobility. A scan for IEEE 802.11b can be completed in around *150 ms* [18], which is negligible compared to duration of a channel re-assignment.

Channel re-assignment can be done either periodically or dynamically based on feedback. The feedback based technique triggers a re-assignment if the quality of the current assignment as measured by the objective function degrades below a relative threshold. We discuss such extensions in our simulations with mobility in Section VII.

Changing the channel for an AP/client is a relatively low-cost operation ($1-2$ ms) which can be implemented mostly as a driver update [9]. The actual operation of changing the channel can be synchronized with an AP's beacon. The IEEE 802.11K draft specifies MAC level primitives to achieve this goal.

VI. Load Balancing

The goal of a channel assignment scheme is to improve user perceived throughput and network utilization. Apart from suffering interference from other APs and clients associated to other APs, a client shares the medium with clients associated to its own AP . The *CFAssign-RaC* algorithm makes clients associate to APs that are conflict-free i.e., free from inter-AP interference. However, if many clients are already associated to an AP, such clients would experience throughput reduction due to considerable *intra-AP* load. Thus, the channel assignment solution should associate clients to APs that minimize a combination of both intra-AP load and inter-AP interference.

Prior work presented in [5] seeks to provide a min-max fair assignment of clients to APs. However, as discussed earlier in Section II, a combined solution to the problem of channel assignment and load balancing is essential. Here we augment objective function of our conflict set coloring formulation to capture load balancing constraints.

Given a wireless network (X, C) and a client $c \in C$. Let the tuple $t_c = \langle r_c, i_c \rangle$ denote the range and interference sets for client c. Let $\theta : X \to \{1 \ldots k\}$ be a channel/color assignment and let $\gamma : C \to X$ be the association mapping function (i.e., the AP to which any client is associated to). Let $\eta(x)$, where $x \in X$, denote the number of clients that are associated to AP x. Now say, the client c is associated to an AP $x \in X$. This client would suffer *conflict* from all APs and clients on the same channel as the client c. Given an AP y on the same channel as x, $\eta(y) + 1$ stations (the AP y and all clients associated to AP y) share the medium with client c. The sum $\sum(\eta(y) + 1)$, for all $y \in (r_c \cup \ i_c)$ such that $\theta(y) = \theta(x)$ captures the total *conflict* (intra-AP and inter-AP) that would be suffered by a client associating to AP x. We denote this by the quantity cf_c. That is, let

$$cf_c = \sum_{\forall x \in (r_c \cup \ i_c) | \theta(x) = \theta(\gamma(c))} (\eta(x) + 1) \qquad (1)$$

The quantity cf_c captures the total *load* suffered by c or more closely, the number of stations that contend with c for the medium. The expected throughput over a unit timescale can be represented as $1/cf_c$ (ignoring short term unfairness inherent in 802.11 MAC).

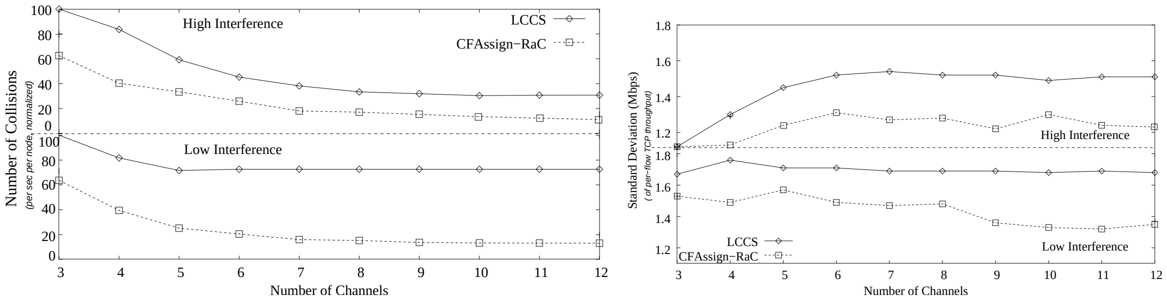
Intuitively, the objective for our channel assignment scheme is to minimize the total conflict in the system, i.e., minimize $\sum_{\forall c \in C} cf_c$. However, this objective function can cause unfairness or imbalance in expected throughput between the clients and thus, we use a min-max conflict optimization function [19]. Let $\vec{CF} = \{cf_1, \ldots, cf_{|C|}\}$ denote the *conflict vector*, i.e., the total conflict experienced by each client, arranged in non-increasing order of value. A channel assignment θ and the corresponding association mapping γ, is said to be a min-max conflict assignment if its corresponding conflict vector $\vec{CF} = \{cf_1, \ldots, cf_n\}$, has the same or lower lexicographical value than any other channel assignment. Given two $n-$tuples of numbers $C = \{cf_1, cf_2, \ldots, cf'_n\}$ and $C' = \{cf'_1, cf'_2, \ldots, cf'_n\}$, each in non-increasing order, we say that C *lexicographically dominates* C' if $C = C'$, or there is some index j for which $cf_j > cf'_j$ and $cf_i = cf'_i$ for all $i \leq j$ [20]. Let $C' \leq C$ denote that C lexicographically dominates C'. We say that C and C' are equivalent if both $C \leq C'$ and $C' \leq C$. This relation defines a total order on the equivalence classes of conflict vectors or the corresponding channel assignments and association mappings. Also, the conflict vectors in the unique minimal equivalence class (under $\leq$) correspond to the fairest channel assignments and association mappings. We denote the lexicographical value of this conflict vector (arranged in non-increasing order of conflict value), the objective function τ. The goal is to minimize the value of this objective function.

To incorporate load balancing, we modify the *CFAssign-RaC* algorithm in the following manner. Step 6 of *CFAssign-RaC* in Algorithm 1 uses the number of conflict-free clients as the objective function. We replace this with the τ objective function discussed above. Note that a client decides to associate to an AP that offers minimum total conflict. This in turn affects the value of the conflict function τ. Thus, because of this feedback, *CFAssign-RaC* requires more rounds to converge to a solution where any further changes to the coloring would only worsen the value of τ (Step 6). The *CFAssign-RaC* algorithm converges provably even with the new objective function (in practice 6 rounds are sufficient). Details are available in a technical report [10]. The key insight is the fact that a lower value of τ is a fairer solution, and is thus bounded (by the fairest solution).

Note that the *CFAssign-RaC* algorithm (modified to be cognizant of client load) jointly solves both the channel assignment and the load balancing problems as follows: *CFAssign-RaC* directly outputs the channel assignment for each AP. By using the load-aware objective function to address conflict set coloring, the *CFAssign-RaC* algorithm implicitly decides the association between the clients and APs (each client is associated to the AP from its range set which has the minimum conflict). This association is a solution to the load balancing problem as well.

VII. Simulations

We have evaluated the performance of our proposed conflict set formulation and the *CFAssign-RaC* algorithm through extensive packet level simulations using the NS2 simulator for a wide range of scenarios including different densities of APs and clients, varying number of available wireless channels, and varying degree of interference. We study the effect of the algorithms on various metrics such as application level throughput for both UDP and TCP flows, the average packet delays, the MAC level collisions and the fairness in terms of the standard deviations of per-flow throughputs. Through two different scenarios of client mobility, we also study the



(a) MAC level collisions.

(b) Standard deviation of per-flow TCP throughput.

Fig. 10. Comparison of MAC level collisions and standard deviation of per-flow TCP throughput for various topologies.

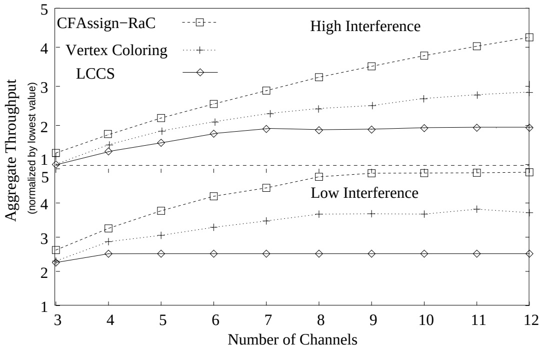


Fig. 7. Comparison of application throughput for CBR/UDP flows for the three algorithms.

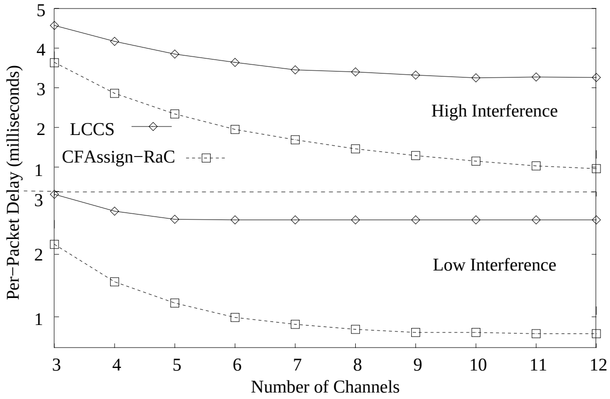


Fig. 9. Comparison of per-packet delay for CBR/UDP flows.

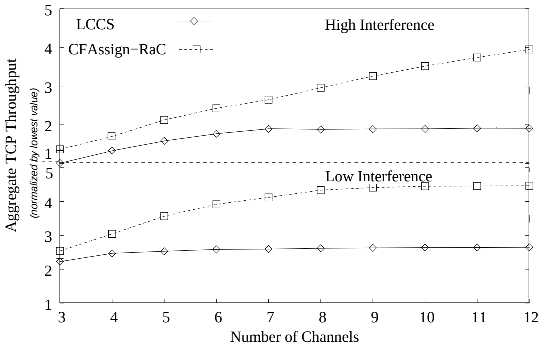


Fig. 8. Comparison of application throughput for FTP/TCP flows for the three algorithms.

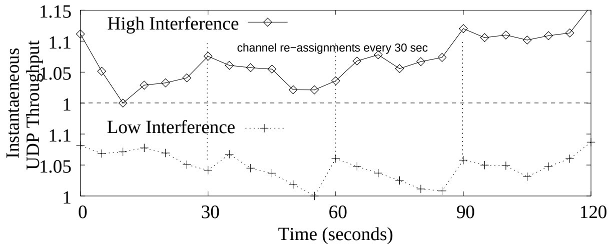


Fig. 11. Instantaneous UDP throughput under mobility Scenario One.

performance and overhead of the *CFAssign-RaC* algorithm augmented with mechanisms for feedback based channel reassignment.

Simulation Parameters: The network topologies consist of 50 APs and 200 clients distributed in a specified region of coverage. We generated two sets of scenarios: high and low interference. For each scenario, we generated 15 different network topologies. The interference was controlled by setting the transmit power of the clients and APs and the receiver sensitivity thresholds. The mean size of the range sets (i.e the set of APs within range) of clients were 4 and 8 for the low and high interference topologies respectively.

Two sets of experiments were performed, one with TCP flows and the other with UDP. FTP download applications over large file sizes were used to create the TCP flows. The UDP flows were generated using a constant bit rate (CBR) traffic generator with rates high enough to saturate the medium. The packet size for all traffic was set at 1024 bytes and the bit rate for the medium was set at 11Mbps.

Various metrics were measured to study the effect of our channel management algorithm on different layers of the network stack. First, we measured the application level throughput for both FTP/TCP and CBR/UDP flows. Second, we measured the per-packet delay encountered by the CBR/UDP flows at the application layer. This delay includes the queues at transmitting stations, and the MAC level delay (because of collisions and backoffs). This metric is useful in studying the effect on voice applications where a deadline oriented delivery of packets is more important than reliability. Third, we studied the utilization of the spectrum as measured by the number of collisions observed on a per node per second basis.

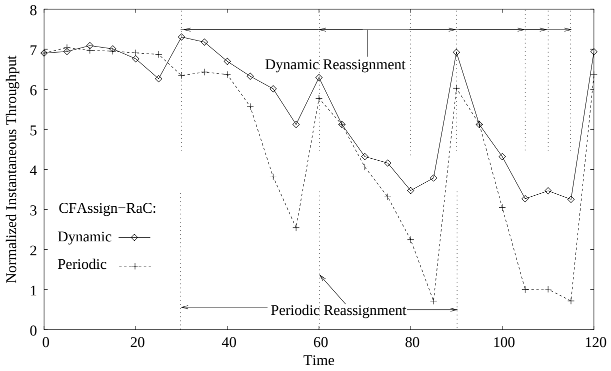


Fig. 12. Comparison of instantaneous throughput of UDP/CBR flows for *RaC-Dynamic* and *RaC-Periodic* under mobility Scenario Two.

This indicates how much of the spectrum was wasted due to contention and how the algorithms affect this. Finally, we study the fairness properties of the algorithms with respect to the throughput achieved on a per-flow basis by observing the standard deviation of the various throughputs. We discuss our key results in detail below:

Throughput: Figures 7 and 8 show the aggregate application level throughput for FTP/TCP and CBR/UDP flows normalized by the lowest value in each plot. Each plot shows two subplots: the top/bottom subplot shows the results for the high/low interference topologies respectively. The lowest value in each subplot is used to normalize all datapoints for that subplot. The number of channels is varied from 3 (802.11b) to 12 (802.11a). The *CFAssign-RaC* algorithm improves the network throughput significantly than LCCS with the performance gap increasing with greater number of channels. Increasing the number of available channels allows the *CFAssign-RaC* algorithm to assign different channels to a proportionately increasing number of Type-2 and Type-3 conflicts (see Section II) which cannot be detected by LCCS, thus widening the performance gap. Increasing amount of interference (as measured by the sizes of the range and interference sets) also increases the number of such conflicts.

Notably, the UDP flows utilize the network better than TCP(absence of backoffs) and hence the aggregate UDP throughput is higher than the TCP counterpart for the same network parameters. Also, we have observed that the TCP flows suffer less collisions than UDP.

Figure 7 also show the throughput achieved by using vertex coloring based approaches [14] (Section III). The vertex coloring based algorithms perform better than LCCS. However, the conflict set system captures channel re-use opportunities better and hence results in improved application throughput.

Per-packet delay: Figure 9 shows two subplots for the per-packet delay for the CBR/UDP flows for high and low interference topologies. The CBR traffic models the voice over IP traffic patterns. Bounded delay jitter is an important quality metric for such synchronous multimedia applications. Figure 9 shows that better channel assignment reduces the average application level delay. The reduction in delay improves as the number of channels increase for the same reason as discussed for the throughput case above.

MAC-level Collisions: Figure 10(a) plots the average number of collisions encountered per second at the MAC level on a per client basis normalized by the highest value in the plot. The rate of MAC level collisions captures the amount of interference on the channel. Thus, this metric directly reflects the quality of a channel assignment scheme in terms of reduction in interference or conflict. Figure 10(a) shows that the *CFAssign-RaC* algorithm reduces the rate of collisions drastically over LCCS. Also the amount of reduction increases with the number of channels because of the lack of ability of LCCS to detect certain conflicts as discussed earlier.

Better Fairness: Figure 10(b) plots the standard deviation of the per-flow throughputs. The *CFAssign-RaC* algorithms improve the fairness as can be seen from the consistently lower values of the standard deviation when compared to LCCS. This is primarily because of the modified τ (Section VI) objective function which incorporates fairness and load balancing constraints on a per-user basis.

Mobility Simulations: Earlier studies [21], [22] have shown that at any given time a small percentage of the users are mobile. Here we evaluate strategies for channel re-assignment based on client mobility. First, we study how periodic re-assignment for the *CFAssign-RaC* algorithm performs in a typical in-building WLAN scenario. Second, we study a feedback based dynamic channel re-assignment strategy and show how the re-assignment frequency adapts to client mobility.

Scenario One: Consider an in-building wireless LAN, with 200 clients and 50 APs with a randomly generated topology. We assume that at a given time atmost 20% of the clients are mobile. This fraction is based on the earlier measurement studies [21], [22]. The clients select a location at random and move with a certain speed. As the clients move, the structure of their range and interference sets gets altered as they perform handoffs. The *CFAssign-RaC* algorithm performs channel re-assignment on a periodic basis. Figure 11 plots the instantaneous client throughput against time. The figure shows two subplots, one each for low and high interference topologies. Each subplot is normalized by the lowest data value of that plot. The plots thus show the relative throughput improvements that occur with channel re-assignments. The simulation runs for a period of 120 seconds and the channel re-assignments occur every 30 seconds. Although in reality, client movements could be negligible for a 120 second period, the client speeds in our simulations were adjusted to match the duration of the simulation. From time $T = 0$ (or just after channel re-assignment), as the nodes move, the throughput begins to drop until the next re-assignment. Figure 11 also shows some opportunistic improvements which happen due to favorable changes in client distributions.

Scenario Two: We study a feedback based re-assignment strategy for the *CFAssign-RaC* algorithm to dynamically trigger channel re-assignments based on client mobility. Specifically, we study two approaches - *RaC-Periodic* as the variant of *CFAssign-RaC* with periodic re-assignments and *RaC-Dynamic* which triggers re-assignment based on the value of

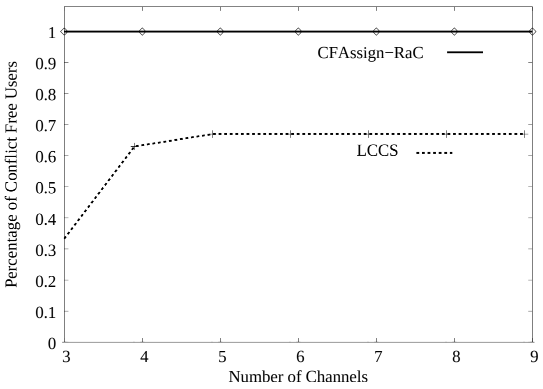


Fig. 13. Comparison of the LCCS and *CFAssign-RaC* algorithm over the in-building testbed network.

the τ objective function discussed in section VI. We consider the same wireless network topology discussed in the previous scenario. Clients use the following mobility model : (i) The clients are made to move with linearly increasing speed with time during the simulation. (ii) The number of clients that are mobile increases linearly with time. Such a mobility model increases the overall mobility in the system with time, thus allowing us to evaluate the performance of the two approaches under varying levels of client mobility.

RaC-Dynamic uses the following re-assignment strategy: Let τ_{LAST} denote the value of the τ objective function (section VI) for the last channel re-assignment. Based on current handoffs reported by APs (using protocols such as IEEE Inter-AP Protocol 802.11f), the central server calculates the value of τ_{NOW}. If $\tau_{NOW} - \tau_{LAST} \geq \tau_{THRESH}$, the algorithm triggers a re-assignment. Here τ_{THRESH} is an empirical constant which can be adjusted to tune the triggering mechanism.

Figure 12 compares the instantaneous throughput achieved by *RaC-Periodic* and *RaC-Dynamic*. The vertical dotted lines at the bottom of the plot indicate channel re-assignments for *RaC-Periodic* which happens every 30 seconds. The vertical lines at the top of the plot indicate the dynamic re-assignments for *RaC-Dynamic*. Initially, since fewer clients are mobile, both approaches trigger re-assignments at T =30,60 sec. However, as more clients become mobile, *RaC-Dynamic* triggers re-assignment more frequently at T =80, 90, 105, 110, 115 sec. This results in improved throughput as shown in the figure. Also if very few clients are mobile, the threshold condition delays the trigger of a re-assignment thus reducing the overhead when unnecessary.

Some of our other results show that the approach improves the throughput by upto 40% even if 10% of the clients and APs implement the *CFAssign-RaC* algorithm thus motivating incremental deployment. Also we observed that the algorithms degrade gracefully in face of increasing interference. Details of these results are available in a technical report [10].

VIII. Experiments

In this section, we discuss results from an operational wireless network with 70 APs spanning four floors of an office building. We observe that in practice, *CFAssign-RaC* always gave the optimal solution (all clients were conflict-free) with just 3 channels while LCCS was unable to resolve certain conflicts even with greater number of channels.

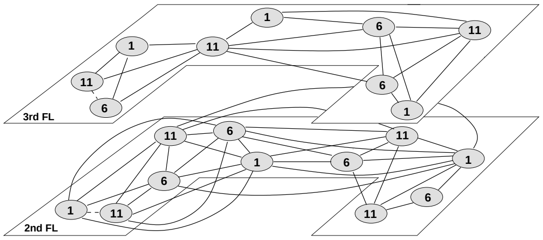


Fig. 14. Visualization of the testbed over two (out of the four) floors. LCCS based channel assignment shown in circles representing AP positions. Edges indicate overlap in coverage.

Topology of the Wireless Testbed: The wireless testbed network consists of 70 IEEE 802.11b APs distributed over four floors of an office building. Half of the APs are *Soekris* boards (see www.soekris.com) with a IEEE 802.11b *Prism II* wireless card configured as a host-based AP while the others are based on Cisco Aironet 340 (AP-340). The *Soekris* based APs operate at 100mW transmit power, while the Cisco AP-340 operates at 30mW transmit power. The channel assignment in use on the network was done by searching for the least congested channel. Figure 14 shows a visualization of the topology of the testbed network over two floors (out of four, for clarity), edges have been placed showing overlap in coverage.

Measurements to compute range and interference sets: The range and interference sets of clients were constructed by the following data-collection process: 40 usage-points were identified on each floor. Two test clients based on IBM Thinkpad T23 laptops with *Prism II* 802.11b wireless cards were used. To compute the range set (i.e. set of APs in range), a single test client was placed at each of these usage-points and a scan of all channels was performed. The interference set (set of APs whose clients are in range) was identified by using two clients placed at near-by locations and performing the necessary scan operations. This gave us the range and interference sets from a total of 160 usage-points.

Experiment Results: We studied the distribution of the sizes of the range and interference sets. 70% of the range sets were of size 6; while 25% were of size 4 and size 8. 80% of the interference sets were of size 2; the rest were of size 4 and 6. Figure 13 shows the performance of *CFAssign-RaC* versus the current channel assignment based on LCCS. The figure shows that *CFAssign-RaC* obtained a conflict-free assignment of channels by just using 3 channels. LCCS was unable to find a conflict-free assignment regardless of the number of available channels. As discussed in Section III, this is because LCCS cannot capture certain types of interference (Section III).

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 management methods discussed in this paper.

IX. Related Work

We discuss some of the existing approaches in Sections II and III. Apart from these, related work broadly falls into three areas:

Channel Assignment in Cellular Networks: Channel assignment in cellular networks is a well studied problem [2]. However, the cells in a cellular network are well planned. They have very regular structures (hexagon is a good approximation) and coverage areas unlike indoor environments. Because of such differences, channel assignment methods for cellular networks such as [23], [24] cannot be applied to WLANs.

Vertex Coloring: Vertex coloring for general graphs is NP-hard. In fact, it is NP-hard to even find a constant approximation. There has been prior work on fully distributed, scalable and light-weight vertex coloring protocols. In [25], Heterniemi and Jacobs present a distributed fault tolerant algorithm to $(\Delta + 1)$ vertex color a general graph. Their algorithm uses local information and is scalable. In [26], Ko and Rubenstein present an algorithm for vertex coloring that can potentially apply to channel assignment in wireless ad-hoc networks. Although such vertex coloring based approaches model the interference constraints well in ad-hoc networks, they suffer from the inaccuracies as described earlier in Section III. In [27], 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. This and similar approaches [3] are AP-centric in nature, hence suffer from the inefficiencies pointed in Section II.

CF coloring for regular structures: Prior work in [16] provides centralized approximation algorithms for conflict free coloring but with a different objective function than ours. Further, their approach assumes regular structures for the transmission ranges (such as axis parallel rectangles, and unit disks) and well-defined properties on the conflict sets which are unrealistic in WLAN environments.

X. Summary

We proposed a client-based model called conflict set coloring that captures interference at the clients to efficiently utilize spectrum in a wireless LAN. We evaluate a centralized algorithm called *CFAssign-RaC* based on conflict set coloring which jointly performs channel assignment and load balancing, otherwise called channel management. Through extensive simulations and measurements from deployed testbeds we show the practical usefulness of such an approach to centrally managed networks. We believe that such client-centric approaches are the key to improved application performance in WLANs and can find wider applicability to newer wireless technologies.

References

[1] I. Katzela and M. Naghsineh, "Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey," *IEEE Personal Communications*, June 1996.

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

[3] K. K. Leung and B.J. Kim, "Frequency assignment for multi-cell ieee 802.11 wireless networks," in *IEEE VTC*, 2003.

[4] J. Geier, "Assigning 802.11b access point channels," *Wi-Fi Planet*, 2004.

[5] Y. Bejerano, S.J. Han, and L. Li, "Fairness and load balancing in wireless lans using association control," in *ACM Mobicom*, 2004.

[6] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *ACM Mobicom*, 2005.

[7] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *ACM MobiCom*, September 2003.

[8] A. Vasan, R. Ramjee, and T. Woo, "Echos: Enhanced capacity 802.11 hotspots," in *IEEE Infocom*, 2005.

[9] I. Ramani and S. Savage, "Syncscan:practical fast handoff for 802.11 infrastructure networks," in *IEEE Infocom*, 2005.

[10] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "Efficient strategies for channel management in wireless lans," *UMD, CS Tech. Rep. CS-TR-4729*, 2005.

[11] R. Leese and S. Hurley, Eds., *Methods and Algorithms for Radio Channel Assignment*, Oxford University Press, 2002.

[12] L. Narayanan, Ed., *Channel assignment and graph multicoloring*, John Wiley & Sons, 2002.

[13] C. McDiarmid and B. Reed, "Channel assignment and weighted coloring," 2000.

[14] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for wlans," *ACM MC2R*, vol. 9, no. 3, 2005.

[15] S. Khurana, A. Kahol, and A. P. Jayasumana, "Effect of hidden terminals of the performance of the ieee 802.11 mac protocol," *IEEE LCN*, 1998.

[16] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky, "Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks," in *IEEE FOCS*, 2002.

[17] IEEE, "Ieee wlan mac and phy: Specification for radio resource measurement," *IEEE Draft 802.11K*, 2005.

[18] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the ieee 802.11 mac layer handoff process," in *ACM CCR*, 2003.

[19] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

[20] J. Kleinberg, Y. Rabani, and E. Tardos, "Fairness in routing and load balancing," in *IEEE FOCS*, 1999.

[21] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *ACM Mobicom*, 2004.

[22] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," in *ACM Mobisys*, May 2003.

[23] B. Krishnamachari, S. Wicker, R. Bejar, and C. Fernandez, "On the complexity of distributed self-configuration in wireless networks," *Journal of Telecommunication Systems*, 2003.

[24] F. Mazzini, G. Mateus, and J. M. Smith, "Lagrangean based methods for solving large-scale cellular network design problems," *Journal of Wireless Networks*, 2003.

[25] S. T. Hedetniemi and D. P. Jacobs, "Fault tolerant distributed coloring algorithms that stabilize in linear time," in *Proceedings of the IEEE IPDPS*, 2002.

[26] B.J. Ko and D. Rubenstein, "Distributed, self-stabilizing placement of replicated resources in emerging networks," in *IEEE ICNP*, 2003.

[27] Y. Lee, K. Kim, and Y. Choi, "Optimization of ap placement and channel assignment in wireless lans," in *IEEE LCN*, 2002.