Load Aware Channel-Width Assignments in Wireless LANs

[1]Paramvir Bahl, [1]Ranveer Chandra, [1]Thomas Moscibroda, [1]Yunnan Wu, and [2]Yuan Yuan
[1]Microsoft Research, [2]University of Maryland
{bahl, ranveer, moscitho, yunnanwu}@microsoft.com, yuanyuan@cs.umd.edu

July 2007
Technical Report
MSR-TR-2007-79

Measurements studies show that there exist large spatial and temporal fluctuations in the traffic load handled by different access points in Wireless LANs. In order to alleviate this problem, researchers have proposed various load-balancing techniques based for instance on channel assignment, power control, or client allocation. Fundamentally, however, assigning each AP the same amount of bandwidth (one channel) can inevitably lead to inefficient usage of the spectrum. In this work, we address the problem by adaptively tuning a radio parameter that has so far been largely untouched in Wireless LAN networks: the channel-width. Particularly, we show that a significant improvement in network capacity and per-client fairness can be achieved if the channel-widths at different APs are made a function of the traffic load. We propose the use of dynamic-width channels, where every AP adjusts its center-frequency and channel width to match its current traffic load. Our techniques are made possible by recent advances in radio hardware design and do not require changes in current hardware. We demonstrate the effectiveness of our scheme through analysis and simulations using real-world scenarios.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

I. INTRODUCTION

One of the core design principles of IEEE 802.11 networks is the use of a simple, fixed channelization structure. The entire available spectrum is divided into smaller channels of equal bandwidth, and the network is operated as a cell network with one channel allocated per cell. For example, the 2.4GHz ISM band has 3 non-overlapping channels, and each Access Point (AP) operates on a particular channel. A Wi-Fi client can only communicate with an AP on one IEEE 802.11 channel at any given instant in time. We argue that this fixed channelization structure severely constrains the total capacity and fairness of IEEE networks. Here's why: typically, clients are distributed across the network unevenly, certain APs become hotspots while others remain unused. Having a-priori channels of fixed width does not account for this scenario and such *spatial disparity* of traffic distribution [21], [14], [15] reduces the overall achievable capacity of the network. To make it more concrete, consider for example the case of a single client in a Wireless Local Area Network (WLAN) with multiple APs. Current IEEE 802.11 will allow the client to only utilize the bandwidth of one channel.

A second reason for inefficiency is un-fairness. The fact that some APs are heavily loaded while others are not, creates a location-induced fairness problem. For example, an AP near a conference room might serve multiple clients on a single channel, which hurts the performance of all clients associated to this AP, while an AP in the corner of a building serves very few clients.

We take a fresh look at the concept of channelization in IEEE 802.11 networks. Our work is inspired by recent advances in hardware technology that allow wireless devices to dynamically change their operating frequency and channel-width [3], [6], [28] with very little overhead. Based on these developments, we propose and evaluate a radically new WLAN architecture that breaks the conventional channelization paradigm – a centralized controller dynamically allocates variable size channel-widths and center-frequencies to every AP. The width of an AP's channel is determined as a function of the traffic demand and the number of interfering APs in its vicinity.

By dynamically allocating variable-length bands to each AP, the network is able to cope with both temporal and spatial disparity of user traffic, which significantly increases the overall network capacity. If there are few clients in the system, the centralized controller assigns a channel with larger width to each AP (subject to practical limitations), enabling the clients to communicate at a higher speed. In addition, this approach is better in terms of fairness than IEEE 802.11, because heavily-loaded APs get larger bandwidth and are thus able to balance the per-client throughput across the network.

We also provide a careful exploration of the theoretical problem of allocating channels of variable width to APs. Whereas the problem of channel assignment in the conventional channelization framework can be modelled as graph coloring, this approach does not model the practical constraint that, due to hardware limitations, each AP can only use a contiguous spectrum band. The presence of this contiguity constraint introduces important new algorithmic challenges. We present a compact integer linear program (ILP) that finds the optimal solution. Being computationally inefficient, the practical applicability of this ILP is limited to small scale networks. Since the problem of allocating dynamic-width channels is NP-hard, we present a efficient approximation algorithm that succeeds in avoiding fragmentation and can be shown to be within a small constant factor of the optimum in terms of both throughput and fairness. Finally, we also propose three computationally efficient heuristic approaches. Our algorithms take into account the practical constraint that often, only a discrete set of channel-width options is available; and they achieve close to optimal performance while significantly outperforming IEEE 802.11's fixed channelization approach.

To summarize, we make three primary contributions:

- We revisit channelization, which is a fundamental, yet largely unexplored, aspect in the design of WLANs. We show that among other parameters, such as transmission power and data rate, WLAN designers should consider the channel-width as a configurable parameter in the design of efficient WLANs.
- We expose and quantify the vast potential increase in both capacity and fairness that results from abandoning today's fixed channelization concept, and we propose a system that is capable of tapping this potential.
- Based on a formal definition of the resulting optimization problem, we devise and evaluate novel algorithms that dynamically and flexibly allocate channels of variable width to different APs. These algorithms are computationally efficient and directly applicable in practice as they operate under the constraints implied by today's available hardware platforms. All our algorithms achieve significantly better results than the state-of-the-art solutions based on fixed channels.

Our results are obtained using extensive simulation in QualNet that are triggered by real data traces from large enterprise/campus WLAN deployments, as well as a network with user mobility. Additionally, since the simulations closely capture the capabilities and constraints of existing hardware, our results show that our approach can be used to significantly improve the per-client capacity and fairness of IEEE 802.11 networks.

II. ADAPTIVE CHANNEL BANDWIDTH

Recent advances in hardware technology allow wireless devices to dynamically change their center frequency and bandwidth to a range of values. For example, WiMAX allows clients to use bandwidths that are multiples of 1.25 MHz, 1.5 MHz and 1.75 MHz [6]. Atheros 802.11 chipsets [2] forms a 40 MHz channel by bonding two continuous 20 MHz channels. Furthermore, we have modified an Atheros chipset to use 5 MHz channel width and any central frequency in steps of 1 MHz in the 2.4 GHz ISM band. These technical advances enable us to reconsider some of the previous design

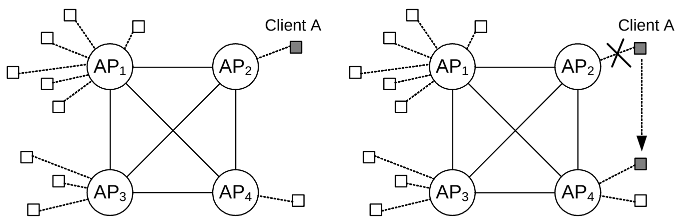

Fig. 1. A network with four mutually interfering APs. With fixed channel bandwidths, both throughput and fairness is suboptimal.

TABLE I
BANDWIDTH RECEIVED BY EACH CLIENT (NORMALIZED BY 20MHZ)

Scenario	AP_1	AP_2	AP_3	AP_4	B	FI
Case 1:C	1/6	1	1/3	1	4	0.58
Case 1:A	2/6	1/2	1/3	1/2	4	**0.97**
Case 2:C	1/6	X	1/3	1/2	3	0.82
Case 2:A	2/6	X	1/3	1/2	**4**	**0.97**

decisions in wireless networks that were made due to practical limitations of the time.

IEEE 802.11 divides the spectrum into a fixed number of channels with equal channel width, which is 22 MHz wide in IEEE 802.11b/g and 20 MHz wide in IEEE 802.11a. Under the assumption of uniform traffic distribution across the network, channelization increases capacity and reduces interference. However, in dynamc conditions, the adherence to fixed-width channels can be problematic and suboptimal. When the number of APs is fewer than the number of available channels, the spectrum is not fully utilized since each AP uses only one channel. On the other hand, if the number of APs is large, two or more neighboring APs are inevitably assigned the same channel, which creates interference [9]. Recent measurements have shown that spatial and temporal disparity in client distributions [11], [21], [14], [15] in large-scale WLANs exist. For example, a recent study of IBM's WLAN consisting of 177 APs [11] shows that 40% of the APs never have more than 10 active clients, while a few APs in auditoriums and cafeterias have 30 simultaneously associated users. The study also shows that the set of heavily loaded APs changes over time, but the current practice of assigning fixed-width channels in IEEE 802.11 networks does not take into account such spatial and temporal disparity in client distributions.

Figure 1 illustrates the scenario with four APs all within mutual interference distance of one another. In case 1, AP_1 has 6 clients, AP_3 has 3 clients, while the remaining two APs have one client each. In case 2, client A moves away from AP_2 and associates to AP_4. We compare the performance of using the fixed-width channels (C) with adaptive-width channels (A). In the fixed-width channel case, the spectrum is divided into 4 channels of 20 MHz each. In the adaptive-width channel case, channels may be 10, 20, or 40 MHz. Table I lists the bandwidth received *per client* at each AP. Also included is the total bandwidth used (B), and Jain's fairness index (FI). The index is calculated using $(\sum c_i)^2/n\sum c_i^2$, where c_i is the bandwidth obtained by client i, and n is the total number of clients.

In case 1, the fixed-width channelization leads to severe unfairness among different clients. The client in the crowded location (AP_1) receives 1/6 of bandwidth compared to the client associated with AP_2 and AP_4. In contrast, with an allocation of 40 MHz to AP_1, 20 MHz to AP_2 and 10 MHz to the remaining APs, fairness improves significantly to 0.97.

Flexible and adaptive channelization is not only important for fairness, but also for system capacity. For instance, in case 2 if client A moves from AP_2 to AP_4, an adaptive approach can reallocate the 10 MHz spectrum formerly used by AP_2 to AP_4 (thus giving AP_4 a total of 20 MHz).

Our study of real-world traces shows that in a large corportate and University wireless networks fairness and capacity problems illustrated in Figure 1 occur frequently.

These examples motivate the need for adaptive-width channel allocation in IEEE 802.11 networks. In Section 6, we show that a wireless network that implements our algorithms to assign different channel-widths to different APs, achieves higher capacity and better per-client fairness than a IEEE 802.11 network.

III. RELATED WORK

AP load balancing in WLANs attempts to evenly distribute the number of clients across all APs in a region. One way to solve this problem is to use Cell Breathing [10]. In this approach the APs in a region adjust their transmission power to force some of its' associated clients to handoff to neighboring APs. Similarly, APs might also increase their transmission power to induce clients to associate to them. This technique is very useful in hotspot and flash crowd scenarios, where many users associate to the same AP, even when the neighboring APs are lightly loaded. Although this scheme is useful is balancing the load across APs, it can potentially worsen the performance if clients associate to far away APs and send the packets at a lower data rate. An alternate approach is client-based, where Wi-Fi devices take smart decisions and associate to the more lightly loaded AP [27]. However, this scheme does not completely solve the user unfairness problem. For example, many clients close to an AP might be contending for resources on a channel of fixed bandwidth, while fewer clients on a neighboring AP might be contending on the same bandwidth.

Another approach to solving the user unfairness problem is to assign more APs to the WLAN [8], [23]. Each user will have a dedicated AP in most scenarios, and so every user gets around the same throughput. However, the benefits of these schemes are limited because of fixed bandwidth channels. First, in extremely dense hotspots, the number of clients might outnumber the number of APs and user unfairness might be unavoidable. Second, when there are very few clients in the network, this technique will waste a large amount of bandwidth.

We overcome the shortcomings of the above schemes by attacking the fundamental problem of fixed-width channels.

We allocate variable size contiguous spectrum to the APs as a function of its load. Our previous work on context of cognitive radio networks ([28], [29]) exploits tunable bandwidths, but we are not aware of load-aware bandwidth assignment in infrastructure-WLAN networks

There are several schemes that are complementary to ours and can be integrated with our proposed approach to further enhance the performance of the WLAN. For example, each AP may allocate a different spectrum slice to every client that is associated to it. This will minimize interference from nearby transmitters and give better throughput.

IV. Design and Algorithm

As pointed out, we envision a network architecture in which the bandwidth of different APs can be adapted according to their respective traffic intensity. Hotspots with many clients will get wider channels at the cost of neighboring APs with little load, which will receive less bandwidth. We begin the section with an overview of our assumptions regarding our system and architecture. Based on this, we formulate a theoretical model which allows us to go on and formulate our algorithms in Sections IV-C through IV-E.

A. System Assumptions

We primarily consider enterprise networks in which all APs are connected via a backbone network. Each access point is capable of obtaining some measure that represents its current *load*. A simple measure could be the number of clients currently associated with this AP, but more sophisticated and accurate measures that take into account the traffic demands of each client may be preferable. At any rate, each AP periodically reports its load to a centralized server that is attached to the network's backbone network and maintains a view of the traffic distribution across the network in a local database.[1] Periodically, the centralized server—based on information stored in its database—runs an algorithm that computes an optimal or near-optimal allocation of channel-widths and center-frequencies to APs. Once computed, it sends the allocations to the respective APs which, along with their associated clients, switch to the new center-frequency and channel-width.

Besides the flexibility to assign more bandwidth to certain APs, bandwidth allocation must also be adaptive in a temporal sense. That is, in order to react to mobility and the dynamic nature of user demand at different APs, bandwidth allocation should not be static in time, as it is in the standard IEEE 802.11 architecture. The centralized server therefore reassigns new bandwidths and center-frequencies to APs periodically, say in intervals of 10 minutes. Alternatively, the spectrum allocation may be updated whenever a *threshold of suboptimality* is surpassed. That is, APs switch to a new bandwidth assignment only when the efficiency of the currently used assignment degrades below a certain point in comparison to the optimal reassignment.

Efficiently setting up and managing a Wireless LAN network poses challenging and complex problems. Several degrees of freedom may be tuned to optimize the network's throughput and/or fairness, including transmission ranges (cell breathing [10]), data rates, load balancing schemes, modulation schemes, density of deployment, and even the locations of the APs. In the sequel, we assume these variables to be fixed (e.g., we assume that each AP has configured the transmission power to obtain the uniform transmission range in different bandwidth settings.), which allows us to more closely study the impact of flexible and dynamic bandwidth allocation on WLAN efficiency. Doing so keeps our results clean from complex inter-dependencies. On the other hand, it is clear that by simultaneously optimizing over multiple tuning parameters (e.g., by combining our adaptive-bandwidth allocation with cell-breathing), even better results are achievable.

Further assumptions we make is that the achievable data rate is linear to the available bandwidth [13]. Also, we make the conservative assumption that overlapping bandwidths always interfere. That is, we seek to assign non-overlapping frequency interval to any two interfering access points.

Clearly, numerous problems of utmost practical importance remain. For instance, since bandwidths of different APs are variable and dynamic in time, there needs to be an efficient method for clients to discover the bandwidth and center frequency of its neighboring access points. Also, the process of an AP (along with its associated clients) switching to a different center-frequency and bandwidth must be smooth and seamless. A more detailed discussion of these and other important practical issues (including the issue of legacy clients) follows in Section V.

B. Problem Formulation and Notation

The main algorithmic problem involved in the system architecture sketched in the previous section is the selection of appropriate channel-widths and center-frequencies. We study a simple network model that allows us to characterize the potential gain of our novel channelization approach. It also allows us to analyze and understand the respective merits of different allocation algorithms. The model makes several simplifying assumptions, but manages to capture those characteristics that govern the design of appropriate algorithms for our bandwidth selection problem. As we focus on the impact of having different channel-widths at APs, we assume each AP to have a fixed (but not necessarily uniform) transmission power P_i and fixed location.

Let the network consist of n access points $AP_1, \ldots, AP_n$. Given the fixed locations and transmission powers, we can determine a *conflict-graph* $G = (V, E)$ of the wireless network as follows [19], [26]: Every AP is represented by a node $i \in V$ and there is an edge between two APs if they have significantly overlapping coverage regions and should therefore avoid transmitting on the same frequency.

[1] Alternatively, using more decentralized, distributed solutions are also possible and an interesting direction for future research. Since the main focus of this work is to identify and quantify the potential gain when abandoning fixed bandwidth channels in IEEE 802.11, we focus on the conceptually simpler centralized solution.

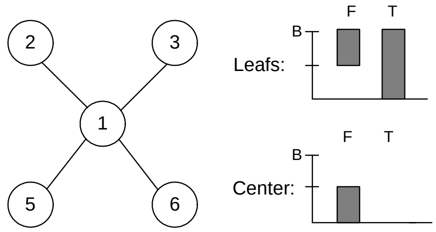

Fig. 2. Network in which a throughput-optimal solution is unfair. T and F denote the allocations in a throughput-optimal and fair solution, respectively.

Practically, we model an edge $(i,j) \in E$ if simultaneous transmission of both AP_i and AP_j could result in harmful interference at some client in the network. Clearly, this binary model of interference is a tremendous simplification of physical reality [25]. In the context of our work, however, it is justified as it is conservative and ignores additional optimizations that could further enhance our system.

In our practical system, the interference relationship between neighboring access points can be determined in an ad hoc fashion (e.g., by APs using beacon messages to probe their proximity to other APs, or by client feedback) as proposed for instance in [26]; or it may be statically provided as part of the network planning. In any case, the conflict graph is static and needs to be updated only rarely, therefore posing no serious practical problem on our system design. For an AP_i, we denote by $N(i)$ the set of all neighboring APs that are potentially in conflict with AP_i, $N(i) = \{AP_j \mid (i,j) \in E\}$.

Let the total *demand* of clients that are associated to AP_i be denoted by D_i bit/s. This demand, along with the interference graph, forms the input to the spectrum assignment algorithm running in the centralized server. The *load* that an AP can serve depends on its clients' demand and, crucially, on its *channel-width*. Let B_i be the channel-width allocated to AP_i and let B_{tot} be the total system bandwidth available. As pointed out, it can be modeled as

$$L_i = \min\{\chi B_i, D_i\}, \quad (1)$$

where χ is a constant that captures how efficiently the available frequency spectrum can be utilized [13]. With standard modulation techniques, this constant is roughly $\chi \approx 1$.[2]

Dynamic-Width Channel-Assignment Problem: The dynamic-width channel-assignment problem in infrastructure-based wireless networks asks for a *non-interfering assignment* of a start frequency S_i and a bandwidth B_i to each access point AP_i. The access point AP_i uses the frequency band $\mathcal{I}_i = [S_i, S_i + B_i]$ for serving its clients and satisfies a load of $L_i = \min\{\chi B_i, D_i\}$. The assignment is *non-interfering* if the assigned intervals $\mathcal{I}_i$ and $\mathcal{I}_j$ of any two neighboring APs i and j, $(i,j) \in E$, is non-overlapping.

A practical algorithm for the dynamic-width channel-assignment problem should achieve two goals: *high throughput* and *fairness*. The former is achieved by maximizing system throughput $L_{Sys} = \sum_{i \in V} L_i$. For fairness, various definitions can be considered and the optimization criterion can be defined appropriately. The difficulty is that in many cases, achieving high system throughput and fairness are contradicting aims. Consider the star graph with uniform demands shown in Figure 2. An allocation maximizing system throughput assigns each leaf AP the entire spectrum, while giving no bandwidth to the center AP. While achieving maximum throughput, such a solution starves clients associated to the AP in the center. A completely fair solution, on the other hand, consists of assigning each AP a channel-width spanning half of the totally available spectrum. In this paper, we address this fairness vs. throughput trade-off by a simple practical solution: We fix a lower bound on the degree of fairness that must be maintained between different APs and strive to optimize the system throughput under this condition.

C. Optimal Solution

The *dynamic-width channel-assignment problem* is fundamentally different from *coloring problems* or *multicoloring problems* that have been extensively studied in the networking community. The reason is that, unlike in (multi)coloring problems, the interval assigned to each AP must consist of a *contiguous* chunk of spectrum of various sizes. This contiguity constraint that does not exist in coloring problems can lead to *fragmentation* of the spectrum. When spectrum becomes fragmented, APs may be unable to reserve a large contiguous part of the spectrum even though the totality of unused spectrum would be sufficiently high. Besides its practical importance, the problem is thus of great theoretical importance as it combines the complexity of both coloring and packing problems.

It is possible to characterize the optimal solution of a problem instance by means of an integer linear program (ILP). Let b_i and s_i be variables that denote the bandwidth and start frequency allocated to AP_i. Further, for each pair of APs i and j with $(i,j) \in E$, we use two binary indicator variables f_{ij} and f_{ji}. The following ILP determines the optimal system throughput achievable in a network with arbitrary channel-width options.

$$\begin{array}{rcll}
\max & \sum_{AP_i \in V} b_i & & \\
s_i + b_i - s_j - f_{ij} \cdot B & < & 0 & , \forall (i,j) \in E \\
s_j + b_j - s_i - f_{ji} \cdot B & < & 0 & , \forall (i,j) \in E \\
f_{ij} + f_{ji} & \leq & 1 & , \forall (i,j) \in E \\
s_i + b_i & \leq & F_{top} & , \forall\ AP_i \in V \\
s_i & \geq & F_{bottom} & , \forall\ AP_i \in V \\
\chi \cdot b_i & \leq & D_i & , \forall\ AP_i \in V \\
f_{ij}, f_{ji} & \in & \{0,1\} & , \forall (i,j) \in E
\end{array}$$

The first two constraints force the auxiliary variables f_{ij} and f_{ji} to behave as follows. The variable f_{ij} is 1 if and only if the top-frequency $s_i + b_i$ of AP_i's spectrum interval is "above" (higher frequency) than the lower end s_j of AP_j's interval. Conversely, $f_{ji} = 1$ if and only if $s_j + b_j > s_i$. Considering

[2] Formula 1 abstracts away the fact that different frequency bands have different signal propagation characteristics. Within the spectrum and bandwidth range studied in this paper, however, the formula is a reasonable approximation.

two intervals $[s_i, s_i+b_i]$ and $[s_j, s_j+b_j]$, it is easy to observe that these intervals overlap if and only if $s_i+b_i > s_j$ and $s_j+b_j > s_i$, i.e., if the top-frequency of both intervals are higher than the start-frequencies of the respective other interval. The third constraint therefore guarantees that no two neighboring intervals in the graph overlap, i.e., the resulting channel assignment is non-overlapping. The remaining constraints are straightforward. The first two ensure that the assigned interval is located within the available spectrum $[F_{bottom}, F_{top}]$. And finally, the sixth one expresses that raising the bandwidth above the demand does not increase throughput.

The important aspect missing in this ILP formulation is *fairness*. However, fairness conditions can easily be integrated into our ILP by adding additional constraints. In our evaluation section, for instance, we consider a fairness condition in which every AP is guaranteed to receive at least its fair share of bandwidth in its neighborhood. In particular, we define $\phi(i) = D_i/(D_i+\sum_{j\in N(i)} D_j)$ as the *minimum fair spectrum-share* that AP_i should receive. We can then enforce this notion of fairness by adding the following constraint to the ILP: $b_i \geq \alpha\phi(i)\cdot B_{tot},\ \forall\ AP_i \in V$. The constant α characterizes the trade-off between fairness and throughput. The smaller α, the more flexibility the ILP solver has to sacrifice fairness in order to improve throughput. Other notions of fairness can similarly be included into our ILP formulation.

The ILP formulation assumes start-frequencies and channel-widths to be arbitrarily tunable. This is in contradiction to existing hardware platforms which typically have a small limited number of *bandwidth options*, a set of available channel-widths to which the transceiver can be tuned. Discrete sets of bandwidths can easily be incorporated in our ILP formulation by restricting the variables b_i to belong to a corresponding set of integers. In Section VI, we examine the impact of this discrete set of bandwidth options.

While the ILP formulation describes the theoretical optimum of any problem instance, it is computationally practicable only in small networks. Specifically, the dynamic-width channel-assignment problem is NP-hard and hence, unless $P = NP$, there exists no efficient solution for its ILP formulation. For the sake of simplicity, we present a simplified version of the theorem that proves hardness only for $\alpha > 2/3$.

Theorem 4.1: The dynamic-width channel-assignment problem problem is NP-hard for any fairness parameter $\alpha > 2/3$. This holds even in restricted geometric graph models such as the unit disk graph.

Proof: The proof is by reduction to the 3-coloring problem of a graph, which is known to be NP-complete even in unit disk graphs [17]. Given an instance $G = (V, E)$ of the 3-coloring problem, construct an instance $G' = (V', E')$ of the dynamic-width channel-assignment problem as follows. For each $v_i \in V$, create 7 APs $AP_i^1, \ldots, AP_i^7$ and connect them to build three triangles as (AP_i^1, AP_i^2), (AP_i^1, AP_i^3), (AP_i^2, AP_i^4), (AP_i^2, AP_i^5), (AP_i^3, AP_i^6), and finally, (AP_i^3, AP_i^7) (cf Figure 3). Further, assume that for each i, AP_i^2 and AP_i^3 have $1/(\alpha - 2/3)$ backlogged clients, and all other APs have one client. When scaling, $D_i^2 = D_i^3 = \alpha - 2/3$ and all other demands are 1. Finally, for each $(v_i, v_j) \in E$, add a link (AP_i^1, AP_j^1) to E'. Observe that due to the fairness condition, every feasible solution must assign APs AP_i^2 and AP_i^3 a spectrum block of width at least

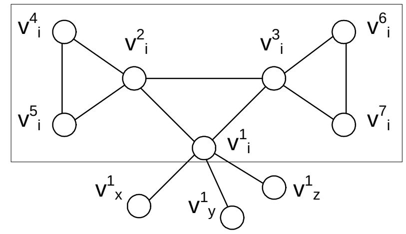

Fig. 3. The gadget representing a node in G.

$$B_i^2 \;=\; B_i^3 \;\overset{!}{\geq}\; \alpha\phi(i)B_{tot} \;\geq\; \frac{\alpha/(\alpha-2/3)B_{tot}}{2/(\alpha-2/3)+3} = \frac{B_{tot}}{3}.$$

We first show that if G is 3-colorable (yes-instance), the total system throughput is at least $T_{Sys}^{yes} \geq 7|V|B_{tot}/3$. Since G is 3-colorable, the graph induced by the APs AP_i^1 can also be colored using three colors. Since each gadget itself can also be colored using three colors (regardless of the specific color assigned to its connector AP AP_i^1), it follows that the entire graph G' is 3-colorable. The lower bound on T_{Sys}^{yes} is now easily obtained by assigning each AP with colors 1, 2, and 3 the spectrum $[F_{bottom}, F_{bottom}+B_{tot}/3]$, $[F_{bottom}+B_{tot}/3, F_{bottom}+2B_{tot}/3]$, and $[F_{bottom}+2B_{tot}/3, F_{bottom}+B_{tot}]$, respectively.

Next, we show that if G is not 3-colorable (no-instance), the total system throughput T_{Sys}^{no} is strictly less than $7|V|/3$. Since the subgraph induced by APs AP_i^1 is not 3-colorable, there must exist at least one AP, say AP_x^1, that is assigned a channel-width of at most $B_{tot}/4$. The total throughput achieve by APs $AP_x^1, \ldots, AP_x^7$ is then at most $(2+1/4)B_{tot}$. Also, because all APs AP_i^2 and AP_i^3 have a bandwidth of at least $B_{tot}/3$, no AP AP_i^1 in G' can have a higher bandwidth than $B_{tot}/3$. Hence, the total throughput is at most $|V-1|B_{tot}\cdot 7/3 + (2+1/4)B_{tot} < T_{Sys}^{yes}$. This concludes the proof. □

While the ILP formulation can thus be used to compute optimal assignments in small-scale networks, this approach does not scale. Therefore, we now investigate computationally efficient approximate solutions.

D. LP-Based Approximation

As mentioned earlier, whereas the problem of channel assignment in the conventional channelization framework can be modelled as graph coloring, a key new flavor in our problem is the need for avoiding *fragmentation*. Specifically, we need to assign one interval to each node, which does not overlap with the intervals assigned to its conflicting nodes (neighbors in the conflict graph). We have degrees of freedom in deciding how long the intervals should be and in deciding where to put them.

1) A Packing Algorithm that avoids Fragmentation: We start by first studying the packing problem in isolation. Assume that the widths of the bandwidth interval allocated to each AP was already determined. How should we efficiently place these intervals? Intuitively, adhering to the following rules of thumb may help:

R1. Pack large items first.
R2. Try to fill up from one end.

Besides being a packing problem, our channel-bandwidth assignment problem also has the flavor of a complex (interval) *coloring* problem. In greedy coloring algorithms, nodes are visited one-by-one, and each node tries to reuse some existing color if possible selecting a new color only if necessary. Clearly, this procedure colors any graph using at most $\Delta(U)+1$ colors, where $\Delta(U)$ is the maximum node-degree. Similarly, if we were not constrained to assign a *contiguous interval* to each AP, we could assure that all required bandwidth can be packed in a total bandwidth of

$$\delta(\boldsymbol{b}) \triangleq \max_{u\in V} \Big(b_u + \sum_{v\in N(u)} b_v\Big), \tag{2}$$

which is essentially the continuous counterpart of the $\Delta(U)+1$ upper-bound. That is, without the contiguity constraint, the greedy coloring algorithm assures that the total bandwidth requirement is $\delta(\boldsymbol{b})$.

We now present an approximation algorithm that combines both the packing and coloring aspects of the problem. Assume that the sizes of all bandwidth intervals followed a power series, i.e., each interval has length 2^k for some integer k. Applying rule of thumb #1, we sort the items in decreasing order of their sizes and try to pack them one by one into the real axis $[0,+\infty]$. Applying rule of thumb #2, when packing each item, we always try to fill up from one end, closer to the origin. When packing in this way, it can be proven by induction that whenever an interval of size 2^k is packed, all available intervals (the spectrum gaps still available) are of size at least 2^k (in fact, they are an integer multiple of 2^k). Hence, in this case, we do not suffer from fragmentation and as pointed out before, the total bandwidth required to pack all intervals is at most $\delta(\boldsymbol{b})$. Therefore, this method achieves for the joint packing and coloring problem the same performance that one can achieve for coloring.

If the bandwidth intervals to be packed do not follow a power series, we can round them accordingly. Suppose the given interval lengths are $b_0 \geq b_1 \ldots \geq b_N$. Then we round each b_i to $\tilde{b}_i = \lceil b_i/b_0 \rceil * b_0$, where the $\lceil x \rceil = 2^{-k}$, for some integer k. Consequently, all intervals can be packed within a maximum length of

$$\max_{u\in V} \ \Big(\tilde{b}_u + \sum_{v\in N(u)} \tilde{b}_v\Big) \ \leq \ 2\delta(\boldsymbol{b}). \tag{3}$$

Finally, we can linearly map the assigned frequencies in $[0, 2\delta(\boldsymbol{b})]$ to the entire available spectrum interval $[F_{bottom}, F_{top}]$. Doing so, we have packed demands $\boldsymbol{b}$ in a maximum interval of $2\delta(\boldsymbol{b})$, which is at most by a factor of 2 (due to the rounding) worse than applying the greedy coloring algorithm to a relaxed problem where each node can make use of non-contiguous bands.

2) Optimizing the interval lengths: The packing algorithm presented in the previous subsection is effective in assuring the performance for the worst AP (with maximum demand in its neighborhood). While this is good from the fairness perspective, it may harm throughput in scenarios in which some parts of the graph are dense, and others are sparse. (Consider for instance a dense clique and a line-network attached to it. Due to the linear scaling at the end of the packing procedure, APs on the line will not utilize the available spectrum efficiently). In this section we present a method for enhancing the overall throughput without sacrificing fairness. We use the packing algorithm as a building block that packs any demand vector $\boldsymbol{b}$ into an spectrum of width $[0, 2\delta(\boldsymbol{b})]$. The idea is to employ linear programs to search for a demand vector with good worst-case performance $\delta(\boldsymbol{b})$ and good overall throughput. We then run the packing algorithm over the resulting demand vector $\boldsymbol{b}$ to pack it into $[0, 2\delta(\boldsymbol{b})]$.

Consider the following linear program:

$$B_{\text{total}}(\alpha) \triangleq \ \max_{\boldsymbol{b}} \ \sum_u b_u, \quad \text{subject to:} \tag{4}$$

$$b_u \geq \alpha\phi_u \cdot B_{tot}, \quad \forall u \tag{5}$$

$$b_u + \sum_{v\in N(u)} b_v \leq B, \quad \forall u. \tag{6}$$

Constraint (6) ensures that the computed vector $\boldsymbol{b}$ results in a feasible solution with a greedy coloring algorithm. Constraint (5) maintains fairness by guaranteeing node u a resource share of αb_u. By varying the constant scaling parameter α from 0 to some maximum value α^*, different tradeoffs between fairness and throughput efficiency can be achieved. Using the maximum value α^* maximizes the worst node's performance; this value can be determined using the following LP:

$$\alpha^* \ = \ \max_{\boldsymbol{b},\alpha} \ \ \alpha, \quad \text{subject to:} \quad (5)(6) \tag{7}$$

Practical Deployment: Our LP-based algorithm leaves open various parameters for tuning the involved fairness vs. throughput trade-off. A simple way of employing it in practice is the following: First, determine the optimal fairness parameter $\alpha = \alpha^*$ using LP 7. Then, using this α, use the first LP to compute $B_{\text{total}}(\alpha)$. This amounts to a conservative approach that maximizes the sum throughput (by "flattening" the demands at the nodes) while assuring the maximum level of fairness at the worst node. The LPs can either be solved directly using an LP solver, or we can apply efficient approximation algorithms for so-called packing LPs [16].

3) Greedy Tuning Step for Discrete Bandwidth Options: The LPs and the packing algorithm together present a method for allocating frequency intervals while avoiding fragmentation. It is designed from the outset for the case where the intervals can be arbitrarily placed. As hardware advances, eventually the hardware may achieve full flexibility in adjusting the center frequency and bandwidth. If instead only a discrete set of bandwidth options are available (as is the case in most currently available hardware), we can round the resulting assignment to comply with the available bandwidth options. In our implementation, we use an additional simple greedy tuning step in order to increase bandwidth wherever possible. The tuning step considers all the APs one by one. If for an AP there exists a wider band that is available, use it; if there is a band with lower-start position, switch to it (recall rule of thumb #2). Repeat thus iterating over all APs until no more improvement are possible.

For any specific α, the performance achieved by the LP-based algorithm can be shown to be within a small constant factor of the optimal algorithm. Due to lack of space, we present the claim without proof.

Claim 4.2: When modeling the wireless network as a disk graph, it holds that for any fairness parameter α, the LP-based algorithm achieves a system throughput that is within a constant factor of the optimal solution. The constant depends on the underlying network model.

E. GreedyRaising: Simple Greedy Heuristics

The LP-based approximation algorithm presented in the previous chapter provides provable performance guarantees with regard to both fairness and system throughput. In this section, we propose three simpler heuristic solutions that is both easier to deploy (it does not require solving a linear program) and, as we show in Section VI still manages to achieve an excellent performance.

All three algorithms are based on the greedy-packing subroutine shown in Algorithm 1. This greedy packing routine takes as its input an ordering of the APs (for example, from heaviest to lightest load) and a bandwidth requirement for each AP. It then proceeds in order of the given ordering and, when considering AP_i, greedily attempts to pack a non-overlapping frequency interval of channel-width B_i into the spectrum. As in the packing scheme of Section IV-D, intervals are packed at the lowest possible frequency at which the interval is non-overlapping with any previously assigned interval at a neighbor.

Depending on the given ordering and bandwidth input, the greedy-packing scheme may not succeed. If the desired channel-widths are too wide, it becomes theoretically impossible to correctly pack. However, even if it *is* theoretically possible to achieve a valid assignment of bandwidth intervals to APs, the greedy allocation may make suboptimal decisions and get stuck in the process. In this case, the subroutine returns false, thereby indicating the the caller should retry using narrower channel-widths.

The basic idea of our so-called GreedyRaising heuristics is the following. Starting from a feasible initial assignment, the heuristics "probes" APs one-by-one and checks whether greedy-packing remains successful if the AP's channel-width is raised. More specifically, GreedyRaising considers all APs in a given sequence $\mathcal{O}$. When considering an AP, its channel-width is increased to the next higher bandwidth option, and the greedy-packing subroutine is called in order to see whether it still succeeds. If it does, the higher bandwidth is adopted; if not, its channel-width is reset to its original value.

The only thing that remains to be defined is the ordering $\mathcal{O}$ in which the access points are considered in both the greedy packing subroutine and the main algorithm. In our studies, we

Algorithm 1 $GreedyPack(B_1, \ldots, B_N, \mathcal{O})$ Routine

Input: Bandwidths $B_1, \ldots, B_N$ and an ordering $\mathcal{O}$ of APs
Output: If possible, a non-overlapping packing of bandwidths into the available spectrum. Return false if no packing is found.

In the order of $\mathcal{O}$: **for each** $AP_i \in V$ **do**
  pack an interval of channel-width B_i in the lowest possible non-overlapping frequency.
end for
if the interval of all APs was successfully packed within the total bandwidth $[F_{bottom}, F_{top}]$ **then**
  return for each $AP_i \in V$ its starting frequency S_i in the successful packing.
else return false
end if

Algorithm 2 GreedyRaising Algorithm

Input: An ordering $\mathcal{O}$ of APs
Output: A non-overlapping packing of bandwidth intervals in the available spectrum.

Set parameter $\theta := 1$ and let $successful := FALSE$;
while not $successful$ **do**
  Let $\phi'_i := \theta \cdot D_i/(D_i + \sum_{j \in N(i)} D_j)$ for each $AP_i \in V$.
  Let B_i be the largest bandwidth option s.t. $B_i \leq \phi'_i \cdot B$
  $successful := GreedyPack(B_1, \ldots, B_N, \mathcal{O})$.
  $\theta := \theta/2$;
end for
In the order of $\mathcal{O}$: **for each** $AP_i \in V$ **do**
  Let $\widehat{B}_i$ be the next higher bandwidth option of B_i.
  $successful := GreedyPack(B_1, .., \widehat{B}_i, .., B_N, \mathcal{O})$.
  if $successful = TRUE$ **then** $B_i := \widehat{B}_i$.
end for

distinguish three possible orderings and evaluate their relative merits. The three orderings are:

- **Most-Congested-First**: In this ordering, APs are sorted in decreasing order of their load.
- **Random**: In this ordering, APs are ordered randomly.[3]
- **Smallest-Last**: Consider an ordering $\mathcal{O}$ and let τ_i be the number of APs that are neighbors of AP_i and that appear *before* AP_i in $\mathcal{O}$. The smallest-last ordering is an ordering which minimizes the maximum τ_i over all APs in the network [22]. This ordering has been studied in the context of coloring problems and is based on the following observation. When considering AP_i in the greedy-packing routine, τ_i reflects the number of potentially interfering intervals that have already been packed in AP_i's neighborhood. Intuitively, the fewer such intervals, the easier it is pack AP_i's allocated bandwidth chunk. Considering

[3]When using this ordering, we slightly adapt our heuristic in the following way. Instead of initially computing a single ordering $\mathcal{O}$ that is used throughout the procedure's execution, we generate a new random ordering $\mathcal{O}$ whenever the greedy packing subroutine is called. This reduces the risk of being stuck with a bad ordering.

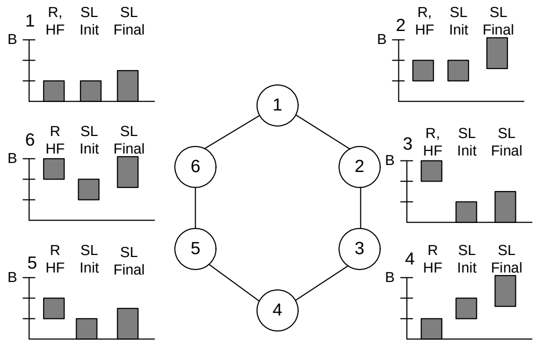

Fig. 4. Ring network with bandwidth options $B/2$ and $B/3$ and uniform load. The smallest-last (SL) packing heuristic performs better ($L_{Sys} = 3B$) than the heavy-first (HF) and random (R) heuristics ($L_{Sys} = 2B$). In the example, the ordering of HF and R is $\mathcal{O} = (1, 4, 2, 3, 5, 6)$.

the APs in smallest-last order minimizes the maximum obstruction that any AP faces when its bandwidth interval is packed. It has been shown in [22] that the smallest-last ordering can be computed efficiently in a single pass:

1. $j := N$; $H := G$;
2. Let AP_j be a minimum degree AP in H;
3. Remove AP_j from H and set $j := j - 1$;
4. Return to step 2 until H is empty;
5. Output $\mathcal{O} = (AP_1, \ldots, AP_N)$.

As our evaluations in Section VI will show, all three GreedyRaising heuristics have the potential of significantly outperforming the scheme based on fixed channels currently employed in IEEE 802.11. The evaluations further indicate that of the three heuristics, the one based on smallest-last orderings consistently achieves the best results.

The tendency of smallest-last to perform better than other orderings can be illustrated using simple scenarios. Consider for instance a network whose APs have (close to) uniform load and are deployed such that the resulting interference graph forms a ring (a line would yield the same results) as shown in Figure 4. In such a network, an optimal allocation would be to assign half of the total bandwidth to each AP, alternating between the upper and lower half. Assume that the ordering of the heavy-first and random orderings are $\mathcal{O} = (1, 4, 2, 3, 5, 6)$ (in the case of heavy-first, this can be the case if the loads are slightly different among APs, or simply by random tie-breaking). After the initial packing (Line 7 of Algorithm 2), all APs are assigned a bandwidth of $B/3$. When attempting to greedily increase some these bandwidths in the second phase, however, no further progress is possible. In particular, regardless of which interval is increased, the packing gets stuck in the process. With the smallest-last ordering, however, the optimal allocation will be reached. Assume for instance that AP_3 is the first AP to be selected (possibly using a random tie-breaking rule). The next AP is one of the two having the least number of neighbors in $G \setminus \{AP_3\}$, i.e., either AP_2 or AP_4. Whichever the algorithm selects, the next AP to be selected must be one that has just one neighbor left in the graph. All possible resulting orderings therefore have the characteristics that the ring is considered "in sequence". In the initial allocation of the smallest-last ordering, every AP is allocated a bandwidth of $B/3$ as in the other heuristics (SL-Init). But, due to the efficient packing, the channel-width of all APs can be raised to the next higher bandwidth option, $B/2$.

F. Discussion

One of the assumptions made in our theoretical modeling is that the frequency bands assigned to neighboring APs should never overlap, which may be overly conservative in many cases [24]. However, both our model and all our algorithms can easily be adapted to incorporate co-channel interference. Particularly, if it is known how much spectrum overlap between neighboring APs is tolerable, our algorithms can be adjusted as follows. For OPT, the first two conditions of the ILP have to be adapted. In the LP-based algorithm packing algorithm it suffices to round up to a power of less than 2, and finally, the packing scheme of all our heuristic approaches will be able to pack the bandwidths more tightly. Finally, notice that both the LP-based algorithm and the GreedyRaising heuristics are computationally efficient and quickly converge to a solution even in large-scale networks.

V. Practical Considerations

Adaptively changing the center frequency and bandwidth allocated to an AP poses several interesting systems challenges. We need to design a new scanning mechanism for clients to discover the APs, since it might be infeasible for them to explore all possible values of center frequencies and bandwidths. Our design should be backward compatible, and the APs should also work with legacy (unmodified) Wi-Fi network cards. In this section, we present some initial thoughts on how these problems can be addressed in a real deployment.

We propose adding an extra radio to each AP, similar to a few commercially available two-radio APs [1], [4]. One radio will operate on the first channel of the band, for example channel 1 for IEEE 802.11b/g networks, or channel 36 for IEEE 802.11a networks. The other radio will adaptively adjust its center frequency and bandwidth to operate in the frequency spectrum that is not occupied by the first radio. Each AP will use the first radio to broadcast beacons and provide service to legacy clients. The beacons will also contain information about the center frequency and bandwidth of the second radio. Clients can then discover the center frequency and bandwidth of the APs by listening to beacons on the channel of the first radio. Even legacy clients will eventually go to channel 1 or 36 as part of the normal scanning process, and discover the APs.

The above architecture has multiple benefits beyond discovery and backward compatibility. For example, it enables fast handoff among clients by allowing a client to quickly discover the nearby APs, by switching to the first channel, and discovering the operating frequency and bandwidth of nearby APs (using Probe Requests and Responses).

Another practical concern is the feasibility of dynamically changing the bandwidth and central frequency of wireless cards. We are currently implementing a proof-of-concept on a wireless card based on the USB and MiniPCI Atheros ar5523 chipset [2]. We have modified the firmware to tune the bandwidth of the wireless card to 5 MHz and 20 MHz, and change the frequency to any value in the 2.4 GHz band. To change the bandwidth, we reduced the speed of the crystal clock by tuning a register value of the Phase Locked Loop (PLL) in the firmware. Consequently, our approach requires the card to go through a firmware reset, which takes a few milliseconds. However, we strongly believe that a firmware reset is unnecessary given the evidence that the same chipset can change the bandwidth to 40 MHz using Turbo mode [2] without a firmware reset.

VI. Performance Evaluation

In this section, we quantify the benefits of dynamic-width channels using simulations QualNet [5]. We compare our schemes, including ILP, LP, and GreedyRaising, against a recently proposed channel assignment algorithm based on fixed channels, called RaC [9]. We analyze the performance using two metrics: aggregate throughput of all clients in the WLAN and per-client fairness. The fairness metric reflects the uniformiy of throughput achieved by all clients, and we define it using Jain's fairness index: $(\sum C_i)^2/n\sum C_i^2$, where C_i is the throughput obtained by client i, and n is the total number of clients.

We first confirm the assumption that the bandwidth, and in turn throughput, achieved by an AP is proportional to the bandwidth allocated to it. We tested this assumption for two bandwidth values: 5 MHz and 20 MHz, on a Netgear AWG132 USB wireless card, which has the Atheros ar5523 chipset, with our modified firmware. In the 5 MHz case, we confirm that the data rate of the packets when the client and the AP were close to each other was 54/4 = 13.5 Mbps. The UDP throughput when using 5 MHz bandwidth was 5.9 Mbps, which is slightly less than 1/4th the throughput when using 20 MHz bandwidth (25.7 Mbps at 54 Mbps data rate). A more accurate reference clock and better frequency alignment mechanisms are required to further improve the effective throughput for smaller bandwidths. We believe that the advances in current radio devices, such as software defined radios [3], will greatly improve these throughput numbers.

A. Simulation Settings

We simulate three real-world usage scenarios: a small-scale enterprise WLAN, a large enterprise/campus WLAN deployment, and a network with user mobility. For a small scale enterprise WLAN, we use the wireless usage data from [12]. This dataset contains monitoring information of 6 APs on the floor of an office building. The floorplan and location of APs is illustrated in Figure 5. The dataset includes the location of all the clients and their wireless usage over a 5-day work week from 8 AM to 8 PM everyday. For our simulations, we feed the coordinates of the APs and the clients in QualNet, and use our algorithms to decide the center frequency and bandwidth of each AP.

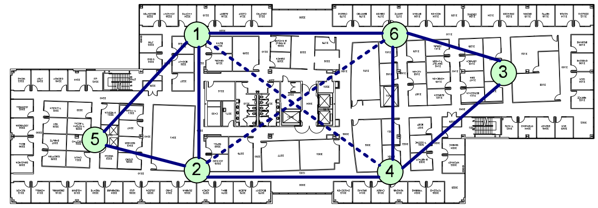

Fig. 5. Floor plan and AP locations on the floor of an office building. The solid lines represent two interfering APs, and dashed lines indicate that the APs interfere at one of the clients.

In the second set of simulations, we consider a larger enterprise network of 20 to 50 APs. We use the data from [11] that analyzed a network across three buildings comprising 177 APs to determine the number of clients associated to each AP. Since we did not have information about the clients' location, we simulate scenarios in which the associated clients are randomly placed within the transmission range of the AP.

Finally, we consider the impact of user mobility on our AP bandwidth allocation scheme. We use the model, called Model T [20], which is based on traces collected across 2 years from the large WLAN deployment in Dartmouth College, and incorporate it with the Random Waypoint Model, to model the mobility pattern of each client.

In our simulations, we study two sets of bandwidth possibilities to show the impact of bandwidth settings on our proposed approach. The first set of bandwidths includes 5, 10, 20 and 40 MHz. The second set includes a wider range of bandwidths: 3, 5, 6, 7, 10, 12, 14, 20, 24, 28 and 40 MHz. We assume that each 1 MHz spectrum delivers 1.2 Mbps data rate [18]. The overall available spectrum is 86 MHz, i.e. the size of 2.4 ISM band. When using channels of 20 MHz, we have 4 non-overlapped channels. Without loss of generality, we neglect the overhead of guide band between two adjacent channels. In our proposed schemes, the clients always associate to the nearest AP and the weight of APs in our algorithms is measured by the number of clients served by the AP. In addition, to stress test the system, we set each client to have at least one backlogged CBR flow to the associated AP. The MAC layer we use is IEEE 802.11 [7]. We use the two-ray propagation model to model path loss. Furthermore, to isolate the impact of varying channel width, we assume no rate or power control.

B. Small WLAN Deployment

We first study the effect of our scheme on a small, but real, WLAN deployment. The floorplan of the office building is illustrated in Figure 5. We extract the user activities from the dataset of [12]. Figure 6 shows the maximum number of clients that are simultaneously associated to each AP during every hour from 8 AM to 8 PM on Monday and Tuesday of

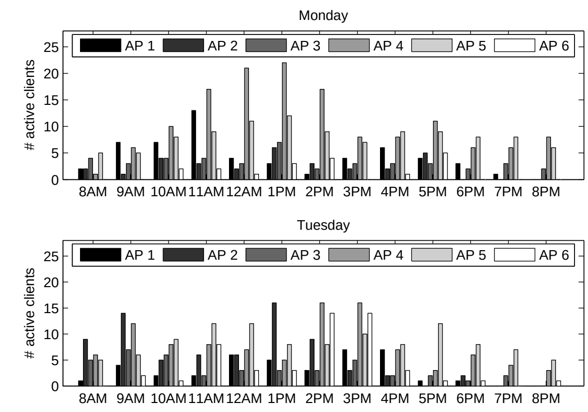

Fig. 6. # active clients at different time of day

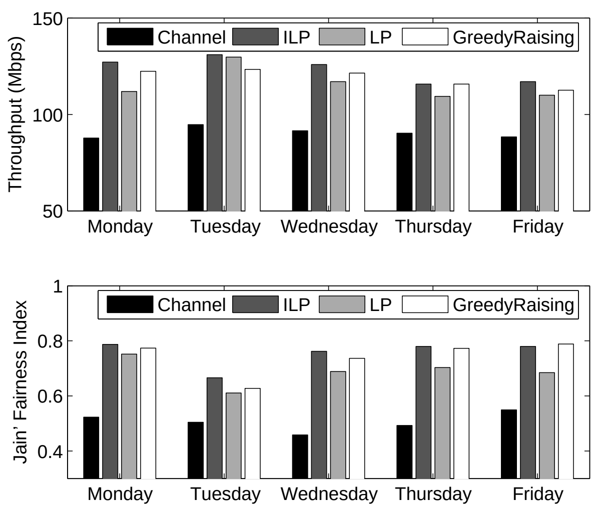

Fig. 7. Throughput and fairness index of different allocation schemes

a work week. [4] Clearly, there is a spatial and time disparity in network usage across different APs. At any given time, APs at some locations serve a significantly larger number of clients then the others. For example, from 11 AM to 2 PM on Monday, AP 4 had up to 22 clients during the peak period since it is located close to several conference rooms. Furthermore, the client populations at the APs varies significantly over time. The set of heavily-loaded APs also changes at various times of the day across different days.

Using this trace, we studied the performance of four schemes: ILP, LP, GreedyRaising (using smallest-last order), and RaC using 4 bandwidth options. Figure 7 depicts the throughput and the fairness index of each AP across 5 days. In all cases, ILP achieves the highest performance, up to 45% higher throughput than RaC, which is based on the fixed channels. The fairness index achieved by ILP is about 0.8, while RaC's fairness index is less than 0.5. This result shows that adaptively assigning the bandwidth to each AP not only improves the capacity of the WLAN, but also ensures more uniform service to all associated clients. On the other hand, when using 4 fixed channels, RaC uses coloring on the AP conflict graph, such that no two interfering APs use the same channel. However, the service received by each client is heavily biased based on their location. The clients associated to a crowded AP suffers from degraded performance, which is reflected as a suboptimal fairness index.

[4]The plots for the other 3 days are omitted due to the space limitations, but they all show a similar trend.

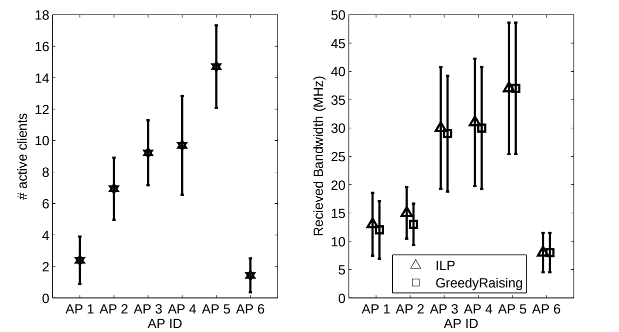

Fig. 8. Average number of clients associated to each AP and the corresponding bandwidth allocated by our scheme.

Compared with ILP, GreedyRaising obtains comparable performance since it emulates the operation of ILP. Based on a certain order, it attempts to raise the bandwidth for each AP starting from the initial feasible assignment. The advantage of GreedyRaising is that it is fast as it benefits from a small set of available bandwidth possibilities. The worst case complexity of the GreedyRaising algorithm is $O(n^3)$, where n is the number of APs. These properties make GreedyRaising a practical solution. LP reduces the throughput by up to 14% since it evaluates all contiguous bandwidth possibilites. Consequently, it loses some throughput as it rounds the bandwidth to the nearest permissible value.

Figure 8 illustrates the number of clients associated to each AP and the corresponding throughput achieved by each AP. The graph shows the average and standard deviation for these values, which demonstrate that dynamic-width channels give more bandwidth to the AP that serves more clients, and the assigned bandwidth varies depending on the variance of the number of associated clients. RaC uses fixed channels, and therefore the amount of bandwidth allocated to each AP does not depend on the number of clients associated to it.

We also studied our algorithms with a larger set of bandwidth options. We observe that in this simple scenario, adding more bandwidth options does not noticeably improve the performance. We also varied the packing schemes and compared their performance. Among them, the smallest-last scheme consistently achieve 5 –10% throughput gain compared to the other two schemes. The gain can be explained by the intuition that assigning the least congested APs last has a higher chance to fit all APs in the available spectrum.

C. Large Wireless Networks

We now study the performance of dynamic-width channel allocation in large campus WLAN deployments. We use observation of the number of clients associated to each AP

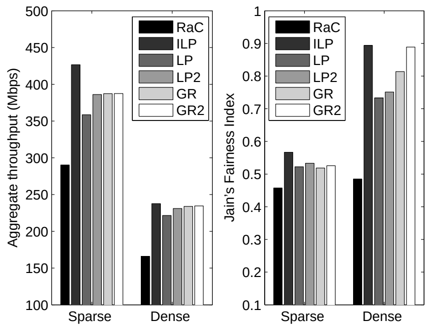

Fig. 9. Throughput and fairness in a WLAN of 20 APs

from a previous study [11]. In this trace, 50% of APs serve less than 5 users, while 10% of APs serve over 15 users. The average number of clients served by each AP is 8. Since the traces provide no information about the location of APs and clients, we randomly place the APs in a flat area of 1000 x 1000 meters. For each AP, we randomly place the client within the transmission range of the AP. The clients are assumed to be static during the experiment. We study our bandwidth allocation scheme for two different scenarios: a 20 AP WLAN and a 50 AP WLAN. For each scenario, we varied the interference among APs by changing the transmission power from -1.6 dbm to 4.2 dbm. All our results are averaged over 20 simulation runs.

Figure 9 illustrates the throughput and fairness index of all clients in sparse and dense deployments when using 20 APs. We emulate a sparse deployment by changing the transmit power of each AP to -1.6 dbm, such that each AP has 2 to 3 neighboring APs. In this scenario, ILP achieves 47% more throughput than RaC. This can be explained by ILP's attempt to allocate all the available bandwidth to the APs. In contrast, RaC is unable to utilize all the channels, as each AP might not have sufficient interfering neighbors. Furthermore, ILP allocates bandwidth to APs proportional to the number of clients associated to it, which further improves system throughput. In fact, it assigns each AP with 40 MHz of bandwidth, as there is little contention among the APs. We note that the fairness index of ILP in the sparse deployment is less than 0.6 since even APs with fewer clients are allocated the maximum of 40 MHz. This appears to be the right behavior as it maximizes spectrum utilization.

We also analyzed a dense AP deployment by setting the transmission power of each AP to be 4.2 dbm (each AP has 5 to 6 interfering APs on average). In this scenario, ILP achieves 53% more throughput than RaC, and improves the fairness index to about 0.9. However, the total throughput of the system is much lesser due to increased interference. ILP allocates separate bandwidths to interfering APs, and therefore it is able to obtain better spectrum utilization. Further, since there is more contention in the system, the lightly loaded APs do not get allocated a 40 MHz bandwidth, and hence the fairness index for ILP is much higher. We note that LP and GreedyRaising obtain near optimal throughput.

We now compare the GreedyRaising algorithm with RaC

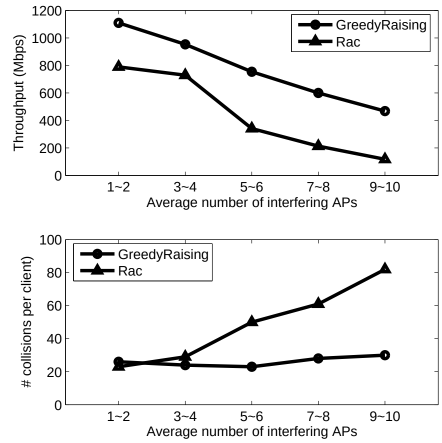

Fig. 10. System throughput and per-client collisions in a WLAN of 50 APs

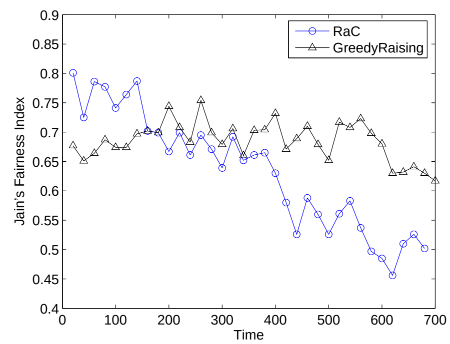

Fig. 11. Fairness Index of 160 clients in a 25 AP WLAN when aggregated over 20 second intervals over 700 seconds.

in a larger WLAN of 50 APs. As we see in Figure 10, the system throughput achieved by GreedyRaising and RaC decreases with increased interference among APs. However, GreedyRaising gets much higher throughput. This can be explained by the second graph, which plots the number of collisions per client with an increase in the number of interfering APs. GreedyRaising allocates separate chunks of the spectrum to interfering APs, and hence the number of per-client collisions stays the same. However, there are not enough non-overlapping channels available to RaC, and hence increased interference among APs increases the number of collisions at each client.

D. Handling User Mobility

Given the recent growth of mobile applications, such as VoIP, over WLANs, we study the effectiveness of our approach in handling user mobility. We stress test our system by having 40% of wireless clients mobile. We combine the registration and mobility pattern defined in Model T [20] with the Random Waypoint Model. Model T captures the popular APs towards which most of the client movements are directed. Each node selects an AP using Model T, and moves towards it with a speed chosen randomly from an interval, (V_{min}, V_{max}]. Upon reaching its destination, the node moves to a new destination after it pauses for a random period between

0 and 10 seconds. We set V_{min} at 0.01 m/s and vary V_{max} from 0.2 to 1.2 m/s.

We consider a WLAN with 25 APs deployed uniformly in a 500 m x 500 m area. The transmission power of each AP set to -1.6 dbm, which gives an approximation transmission range of 100 meters. The number of interfering APs varies from 3 to 8. Initially, clients are uniformally distributed across each AP. At the start of the simulation, clients begin to move towards the APs defined by model T. RaC reassigns the channel every 50 seconds. GreedyRaising adjusts the bandwidth allocation if a new assignment improves the fairness index or the system throughput by more than 10%.

Figure 11 shows the fairness index for the WLAN over time. Each point in the graph is an aggregate fairness value computed over a 20 second interval. Initially, GreedyRaising has a worse fairness index than RaC. This is because GreedyRaising assigns more bandwidth to APs on the edge, and lesser bandwidth to APs in the middle to create enough channels. As clients begin to move, their distribution across all the APs gets skewed as the popular APs serve a much larger number of clients. GreedyRaising captures this change and dynamically adjusts the channel widths, therefore, achieving consistent fairness over time. On the other hand, RaC is based on the fixed channels, and consequently it is unable to handle skewed client AP distributions.

We also measured the overall system throughput aggregated over the entire 700 second interval. Our approach delivers a total throughput of 273 Mbps while RaC delivers 195 Mbps throughput. The reason for the difference is similar to observations in the previous subsections.

VII. Conclusions

The fixed-width channelization technique in IEEE 802.11 networks is inherently incapable of efficiently coping with the spatially non-uniform and temporally dynamic user demand that is prevalent in most infrastructure networks deployed today. In this paper, we have argued that by moving beyond these pre-determined channels of fixed width, a significant increase of both per-client fairness and system capacity can be achieved. Made feasible by recent advances in hardware technology, we propose a system and a set of algorithms that efficiently and dynamically allocate center-frequencies and channel widths to APs as a function of their traffic load.

References

[1] Aruba Networks, http://www.arubanetworks.com.

[2] Atheros Communications, http//www.atheros.com/pt/whitepapers/atheros-superg-whitepaper.pdf.

[3] GNU Software Radio, http://www.gnu.org/software/gnuradio/.

[4] Meru Networks, http://www.merunetworks.com.

[5] QualNet, http://www.qualnet.com/.

[6] Wimax forum whitepapers: http://www.wimaxforum.org/.

[7] IEEE 802.11b/D3.0, Wireless LAN Medium Access Control(MAC) and Physical (PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4 GHz Band, 1999.

[8] N. Ahmed and S. Keshav. SMARTA: A Self-Managing Architecture for Thin Access Points. In *ACM CoNext*, 2006.

[9] A. M. V. B. S. B. A.Srinivasan and W. Arbaugh. Client-driven Channel Management for Wireless LANs. In *IEEE INFOCOM*, 2006.

[10] P. Bahl, M. T. Hajiaghayi, K. Jain, V. Mirrokni, L. Qiu, and A. Seberi. Cell Breathing in Wireless LANs: Algorithms and Evaluation. *IEEE Transactions on Mobile Computing*, 2006.

[11] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *MobiSys*, 2003.

[12] R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-Based Management System for Enterprise Wireless LANs. In *Fourth Symposium on Networked Systems Design and Implementation (NSDI)*, April 2007.

[13] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[14] David Kotz and Kobby Essien. Analysis of a Campus-wide Wireless Network. *Wireless Networks*, 11:115–133, 2005.

[15] Diane Tang and Mary Baker. Analyis of a Local-Area Wireless Network. In *Proceedings of the Annual ACM International Conference on Mobile Computing (MobiCom)*, August 2000.

[16] N. Garg and J. Koenemann. Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems. In *Proc. of the 39^{th} Annual Symposium on Foundations of Computer Science (FOCS)*, 1998.

[17] A. Graf, M. Stumpf, and G. Weissenfels. On Coloring Unit Disk Graphs. *Algorithmica*, 20, 1998.

[18] J. Proakis, Digital Communications, McGraw Hill, 2001.

[19] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of Interference on Multi-hop Wireless Network Performance. In *ACM MobiCom 2003*.

[20] R. Jain, D. Lelescu, and M. Balakrishnan. Model T: An Empirical Model for User Registration Patterns in a Campus Wireless LAN. In *mobicom*, pages 170–184, 2005.

[21] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proceedings of the Annual ACM International Conference on Mobile Computing (MobiCom)*, 2002.

[22] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering in graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.

[23] V. Mhatre and K. Papagiannaki. Optimal Design of High Density 802.11 WLANs. In *ACM CoNext*, 2006.

[24] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially Overlapped Channels not Considered Harmful. *SIGMETRICS Perform. Eval. Rev.*, 34(1):63–74, 2006.

[25] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-Based Models. In *ACM HotNets*, 2006.

[26] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *In Proc. of Internet Measurement Conference (IMC)*, 2005.

[27] K. Sundaresan and K. Papagiannaki. The Need for Cross-layer Information in Access Point Selection Algorithms. In *Internet Measurment Conference*, pages 257–262, 2006.

[28] Y. Yuan, P. Bahl, R. Chandra, P. A. Chou, I. Farrel, T. Moscibroda, S. Narlanka, and Y. Wu. KNOWS: Kognitiv Networking Over White Spaces. In *Proceedings of IEEE DySpan*, 2007.

[29] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu. Allocating Dynamic Time-Spectrum Blocks in Cognitive Radio Networks. In *Proceedings of ACM MobiHoc*, 2007.