Traffic-Aware Channel Assignment in Wireless LANs

Eric Rozner⋆, Yogita Mehta⋆, Aditya Akella†, and Lili Qiu⋆
University of Texas at Austin⋆, University of Wisconsin-Madison†

***Abstract*— Campus and enterprise wireless networks are increasingly characterized by ubiquitous coverage and rising traffic demands. Efficiently assigning channels to access points (APs) in these networks can significantly affect the performance and capacity of the wireless LAN. Several research studies have tackled this issue. However, even the state-of-the-art approaches assign channels without considering prevailing traffic demands.**

The channel assignment problem has a parallel in the wireline world, where recent work has established the tremendous effectiveness of using traffic demands in network engineering decisions. Motivated by this, our paper explores whether the quality of a channel assignment can be improved by incorporating observed traffic demands at APs and clients into the assignment process. Using extensive simulations over publicly-available wireless traffic traces, as well as synthetic settings, we show that being *traffic-aware* could substantially improve the overall quality of a channel assignment. We develop and evaluate practical traffic-aware assignment algorithms that predict future demands based on historical information and use the predicted demands for assigning channels. Finally we demonstrate the effectiveness of traffic-aware assignment using testbed experiments.

I. Introduction

In the past few years, wireless networks have made significant in-roads into the common workplace. Today, most enterprises and campuses - large or small - offer near-ubiquitous wireless coverage. There is also anecdotal evidence that the traffic volumes in workplace WLANs have grown significantly in the matter of just a few months [11].

Ensuring good wireless performance in these modern settings is challenging. The broadcast nature of wireless communication implies that WLANs are plagued by severe interference issues. Growing densities of deployment together with increasing traffic volumes only exacerbate these problems.

Traditionally, careful *channel assignment* has provided some respite from this problem. In the common case, network administrators conduct detailed site surveys, and try out various configurations to manually determine the right channel and placement for each AP. The state-of-the-art research [15], [17] also offers similar static solutions. However, the ever-changing nature of the wireless frontier, with newer devices and user applications contending for the wireless medium [11], will soon render these manual, one-time approaches ineffective.

Researchers in the wireline world have faced similar issues when static routing weights were deemed insufficient for managing the resources of large ISP networks. As a solution, researchers advocated adapting the routing weights to observed traffic demands. In the past few years, several operational and research papers have shown the tremendous effectiveness of this approach. Motivated by the success of these approaches in the IP world, our paper asks the following question:

> *Does the quality of a channel assignment improve when dynamic traffic demands in the WLAN are taken into account?*

To answer this question, we develop and systematically study the notion of *traffic-aware channel assignment* for WLANs. We espouse traditional objectives for optimizing the channel assignment, and show how they can be modified to incorporate the observed traffic demands of wireless APs and clients, as well as their locations. We outline simple approaches for collecting current demand information in practice. Obtaining optimal channel assignments that satisfy these objectives is NP-Hard. Therefore, we develop a simple set of techniques for efficiently obtaining channel assignments that can track the prevailing network conditions very closely.

To evaluate the benefits of our traffic-aware approach, we use extensive simulations over both real topologies and traffic demands (available publicly at [10] and [12]), as well as over several synthetic settings. We also conduct many small-scale testbed experiments. In either case – simulations or experiments – we first evaluate a setting where *perfect information* about current and future demands is available. These baseline analyses help establish the potential benefits of traffic-aware channel assignment algorithms. Our simulations and experiments show that being traffic-aware could substantially improve the quality of a channel assignment in terms of total network throughput. We find that approaches that incorporate the traffic demands of both clients and APs are often superior than those that solely rely on AP traffic demands. The exact level of improvement from traffic-awareness depends on the deployment scenario – e.g. the density of wireless nodes, the traffic volumes, and the number of "hot-spot" APs. We carefully evaluate the operating conditions where traffic-awareness can offer the maximum benefit. We also observe that traffic-aware channel assignment offers similar fairness as existing traffic-agnostic approaches.

In practice, the assumption of perfect demand information is unrealistic. To address this, we propose several approaches for traffic *demand prediction*, and we extend our traffic-aware channel assignment algorithms to use predicted demands. We show that the performance from the resulting channel assignments is very reasonably close to the ones obtained with access to perfect information (within 5%).

The rest of the paper is organized as follows. In Section II, we survey related work. We introduce traffic-agnostic and traffic-aware performance metrics used for channel assignment in Section III, and develop assignment algorithms in Section IV. In Section V, we describe prediction algorithms to estimate traffic demands. We introduce our evaluation methodology and datasets in Section VI. We present simulation and testbed results in Section VII and Section VIII, respectively. We discuss practical issues in Section IX and conclude in Section X.

II. Related Work

We review past approaches to channel assignment applied to two different settings: enterprise/campus WLANs, and multi-hop mesh networks. We note our focus is on the first setting.

Campuses/Enterprises. Assigning channels across APs in WLANs has traditionally been a static one-time approach [13]: First, net-admins conduct an "RF site survey" of the campus and determine the location and the number of APs required for adequate coverage. Then, the admin manually configures APs with 802.11's non-overlapping channels to ensure that close-by APs operate on different channels when possible. We show in this paper that such static approaches result in poor performance in the face of shifting traffic demands.

There are several research proposals for channel assignment in campus WLANS [15], [17]. However, unlike our paper, none of them consider the benefit of tailoring the channel assignment to prevailing traffic demands. For example, Lee et. al [15] advocate identifying "expected high-demand points" in a given WLAN deployment, and assigning channels so as to maximize signal strength at the demand points. This is still a static, one-time approach. Mishra et. al [17] argue that clients have a better view of interference (since interference directly impacts their performance), and therefore channel assignment must take client-side views of interference into account. However, this approach only takes client locations into account and assumes that all wireless nodes exhibit the same level of activity at all times. In our work, we show the potential benefit of taking into account the instantaneous levels of activity of different wireless nodes. We also show how to predict future trends in activity based on historical information.

Recently, several commercial "spectrum management" products have been developed to automate channel assignment across WLANs. Some of these products perform dynamic channel selection based on the current operating conditions (e.g. AutoCell from Propagate Networks [2] and Alcatel OmniAccess AirView Software [1]). A few of these also offer interference mitigation via transmit power control, and load balancing across APs. Unfortunately, due to their proprietary nature, very little is known about the design of these products, their potential benefits, the operating conditions they work best under, and reasons for their failings (if any). In our work, we provide a thorough analysis of these issues for traffic-aware channel assignment. We believe that our observations will be crucial to the design of future commercial offerings.

Multihop mesh networks. Two classes of solutions have been proposed to improve network capacity in multihop mesh networks: The first class, proposed by Raniwala *et al.* [20], [21] advocates equipping mesh network nodes with multiple network interface cards (NICs) operating on different channels. The second, proposed by Bahl et al. [5], advocates a new link-layer mechanism called SSCH, wherein neighboring mesh nodes perform synchronized channel hops to better exploit frequency diversity. In both cases the goal is to ensure that neighboring nodes are assigned the same channels, or overlapping hopping sequences, for data to be successfully transmitted. In contrast, WLAN-settings require neighboring APs to be assigned to distinct channels to mitigate interference. Nevertheless, we believe that the core idea of traffic-aware channel assignment can be applicable to these settings as well.

Next, we briefly review IP traffic engineering approaches and discuss how they motivate our work.

Traffic Engineering in ISP Networks. Traffic demands have been shown to have tremendous utility for network provisioning and route optimization in ISP networks [3], [4], [24]. A wide range of traffic engineering approaches have been developed to incorporate traffic demands. At a high level, these approaches maintain a history of observed traffic demand matrices, and optimize routing for the representative traffic demands extracted from the observed traffic during a certain history window. They differ in how representative demands are derived. Inspired by these results from the IP wireline world, we ask whether being traffic-aware has similar benefits for managing wireless network spectrum. We also seek to develop a parallel set of approaches for deriving traffic demand information in wireless LANs.

III. Performance Metrics

First, we discuss a traditional optimization metric used in channel assignment and then present modifications to make the metric traffic-aware. A desirable optimization metric should satisfy two key conditions: (i) easy/efficient to compute, and (ii) strongly correlated with the network's performance.

The goal of channel assignment is to ensure that wireless nodes which belong to distinct Basic Service Sets (BSSs), and are within interference range, operate on distinct channels whenever possible. A BSS includes an AP and all clients associated with it. An entire BSS operates on a single channel. Also, only nodes belonging to different BSSs can interfere.

A natural way to capture this goal is to use a "channel separation" metric. At a high level, the metric aims to maximize the difference in the channels used by interfering nodes. In what follows, we first introduce two channel separation metrics that ignore traffic-demands, and then extend them to account for the prevailing traffic in the network.

***Traffic-agnostic, client-agnostic* channel separation.** Let C_i denote the channel assigned to AP i, $d(i,j)$ denote the distance between i and j, I denote the interference range, and A denote the set of all APs. Also, if $d(i,j) < I$, define $Separation(i,j) = min(|C_i - C_j|, 5)$, otherwise $Separation(i,j) = 5$. The channel separation metric or objective can then be expressed as:

$$Maximize : \sum_{i,j \in A} Separation(i,j)$$

The interference range I - which is key to the definition of separation - is computed as follows: If P is the strength in dBm of a transmitted signal, then the received signal strength at a distance d meters away is $P - (40 + 3.5 \cdot 10 \cdot log(d.))$. [22][1]

[1] We use constants that correspond to measurements reported in [14].

Then, I, is defined by the equation $P-(40+3.5\cdot 10\cdot log(I)) = T$, where T is the carrier sense threshold in dBm.

***Traffic-agnostic, client-aware* channel separation.** The above metric only considers interference among APs. In real networks, minimizing interference introduced by client transmissions is also important. Indeed, our analysis of real traffic traces shows that clients transmit a significant volume of traffic. It is equally important to reduce the interference experienced by clients due to transmissions from neighboring APs or clients (e.g. might arise in hidden terminal situations).

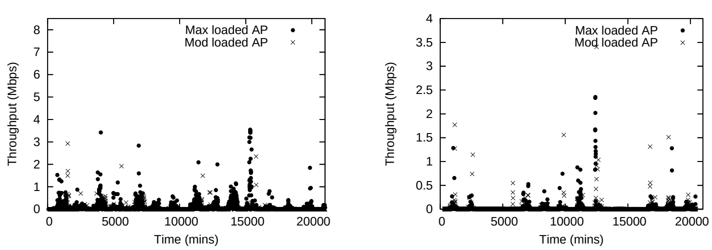
To do this, we extend the above client-agnostic channel separation to the one below, where B denotes the set of clients in the network. We assume that the client locations are known a-priori. In effect, the next metric factors in the channel separation between any two interfering APs, any two interfering clients that are associated with different APs and, an interfering AP-client pair.

$$Maximize: \sum_{i,j \in A \cup B, BSS(i) \neq BSS(j)} Separation(i,j)$$

Nodes i, j in the sum above must belong to different BSSs.

Extending channel separation to be traffic-aware. The above two metrics do not take into account the actual traffic volumes of individual clients and APs. Therefore, optimizing either of these metrics may force interfering but relatively inactive APs or clients to operate on non-overlapping channels, whereas, a smarter channel assignment would have re-used these channels to improve spatial reuse and mitigate interference at other active network locations.

To verify traffic varies across BSSs, we examine the time series of traffic demands at APs from publicly-available traces. As Figure 1 shows, traffic volumes indeed vary substantially across APs, and also across time. We observe similar variation among clients. Such variation prevents traffic-agnostic metrics from fully exploiting the capacity of the wireless channel.



(a) Dartmouth: LibBldg (Sent data) (b) IBM: MBldg (Sent Data)

Fig. 1. Time series of traffic for a heavily loaded and moderately loaded AP from LibBldg in the Dartmouth Data (a) and MBldg in the IBM data (b).

Before outlining the traffic-aware metrics, we informally define the term "demand". The "sending" demand of a node is the aggregate amount of data (excluding ACKs) that it wishes to transmit to various recipients per unit time (in the case of a client, there is a single recipient – its AP; in the case of an AP, all of its clients could be recipients). Similarly, the "receiving" demand is the aggregate amount of data (excluding ACKs) per unit time that the node wishes to receive from various transmitters. In Section V, we discuss how send and receive demand information can be collected in practice.

To incorporate traffic demands, we modify the traffic-agnostic channel separation metrics so that interfering nodes with high individual demands (specifically the BSSes containing such nodes) are first assigned to non-overlapping channels. We use the following insight: Whenever two nodes A and B are in interference range of each other, the transmissions of one node will effect not only the transmissions at the other node but also the receptions at the other node. The former effect is a manifestation of 802.11's carrier sense and back off mechanisms. The latter occurs due to packet collisions and collision drops that can arise in hidden-terminal settings.

Using this insight, we scale the channel separation between A and B with the following "weight":

$$W_{A,B} = S_A \times (S_B + R_B) + S_B \times (S_A + R_A)$$

Where S is the send demand, and R is the receive demand. The first term in the sum reflects the effect of A's transmissions on the transmission and reception of B. Intuitively, if we abuse notation and let S_A (R_A) denote the fraction of time A's transmissions (receptions) acquire the medium, the first term reflects the *probability* of A's transmissions interefering with B's transmissions or receptions. Similarly, the second term reflects the effect of B on A.

Using such weights, we can modify the first of the traffic-agnostic metrics above to the following traffic-aware channel separation metric. Note that this metric ignores interference from clients, and hence is ***traffic-aware, but client-agnostic***.

$$Maximize: \sum_{i,j \in A, j \neq i} W_{i,j} \times Separation(i,j)$$

Similarly, we can modify the second traffic-agnostic metric above to also account for traffic and interference from clients. This gives the following ***traffic-aware, client-aware*** objective:

$$Maximize: \sum_{i,j \in A \cup B, BSS(j) \neq BSS(i)} W_{i,j} \times Separation(i,j)$$

IV. Channel Assignment Algorithms

Optimizing channel assignment is known to be NP-hard [17]. The hardness properties also hold for the separation metrics above. Since finding an optimal assignment is arduous, we use simulated annealing [23] to optimize each channel assignment metric. Simulated annealing is appropriate in this context since it can iteratively improve the solution while avoiding being stuck in local optima. To achieve good performance and speed up the convergence of simulated annealing, we use an informed initialization algorithm that is inspired by the Chaitin's approach to the register allocation problem [9].

A. Initialization Algorithms

We first describe an initialization algorithm that does not consider traffic demands and treats every node equally. Then we extend the algorithm to account for differing traffic demands at each node. In either case, initialization *does not* take client locations into account, irrespective of whether the metric in question is client-aware or client-agnostic.

Figure 2 shows the algorithm for the traffic-agnostic case. The intuition of the algorithm is to defer channel assignment for those APs that have many conflicts with other APs. This is because for such APs, the choice of the channel is very important, and more restrictive, as it depends on the channels assigned to neighboring APs. Also, when an AP has few conflicts, we have a greater amount of flexibility in assigning channels. For such APs, we can even assign channels without knowing the channels chosen for the neighbors. In this algorithm, K refers to the number of non-overlapping channels.

1) Construct a conflict graph G for APs in the WLAN, where there is an edge between any two nodes if they interfere.
2) For any vertices in the conflict graph that has degree less than K, choose the one with maximum degree and delete it and its associated edges from the graph and push it onto a stack. Repeat until no vertices with degree less than K remain.
3) If the resulting graph is non-empty, choose the vertex with maximum degree and remove it from the conflict graph and push it onto the stack. Go to step 2.
4) For all the vertices on the stack, pop one vertex at a time, add it back into the graph and color it with a color that is different from all its neighbors (up to this point). If a vertex cannot be colored, mark it.
5) For the marked vertices, assign them a color that results in minimum interference, where interference is calculated as # interfering APs assigned the same color.

Fig. 2. Initialization algorithm for channel assignment.

To extend the initial assignment to be traffic-aware case, we do the following: First, we modify the degree used in step #2 and #3 by weighting it with total traffic as follows: $degree(i) = \sum_{j \in G} interfere(i,j)$, where $interfere(i,j) = 0$ if i and j are not in interference range; $interfere(i,j) = sent(j) + recv(j)$ otherwise. Note $sent(j)$ and $recv(j)$ are sent and received traffic at node j normalized by the link bandwidth. Second, in step #5 we assign marked vertices with a color that results in minimum interference, where the interference at node i from node j is defined as $interference(i,j) = 0$ if i and j are on separate channels or not in interference range; $interference(i,j) = sent(j) + recv(j)$ otherwise. We then choose the color that results in the minimum value of $interference(i,j)$ summed over all $j \in A$.

B. Further Improvement via Simulated Annealing (SA)

We apply the above initialization algorithm to get a good initial channel assignment. We further improve the channel assignment through an iterative search; we compared several options for the search, including random walk, simulated annealing (SA) with random walk and greedy search, and found that SA offers faster convergence and better assignment. Therefore we use SA in our evaluation.

SA is inspired by the metal annealing process. In each iteration, we randomly assign one of the APs (and its clients) to a different channel from the current assignment. If the new assignment is better, we update the current assignment with the new one. Otherwise, we update the current assignment with the new one with the probability $e^{(f_{new} - f_{curr})/T}$, where T is current temperature, f_{new} and f_{curr} are the values of objective functions under the new and current channel assignments. The temperature gradually decreases so that we are more likely to accept a worse solution initially and avoid being stuck at local optimal. As the temperature approaches to 0, we progressively move in the direction of improving the objective function. Our evaluation sets initial temperature to 10, and each iteration reduces temperatue to 0.999 of the current value. We use 1000 iterations and the output is the best solution (in terms of the given separation metric) over all iterations.

We note that the execution time of this approach is sufficient for practical WLAN settings (e.g., it takes less than 1 second for SA to compute the optimized metric value in the traces we study). Note that while SA is an effective search algorithm, it does not guarantee globally optimal solution within these iterations. This is not surprising due to the NP-hard nature of the channel assignment problem.

V. Traffic-Awareness in Practice

In this section, we address two issues: (1) how to collect traffic demand information in practice and (2) how to estimate traffic demands from historical information.

A. On Obtaining Demand Information

In practice, AP and client traffic demands can be gathered using SNMP [8]. Most commercial access points export an SNMP management interface that can be polled at 5 minute intervals to obtain: (1) total bytes sent by the AP (*IfOutOct*); (2) bytes received at the AP (*IfInOct*); and, (3) the number of active clients currently associated with the AP (*NumClients*).

Together these statistics provide estimates of the send and receive demands of both APs and clients at 5-minute time granularity, as follows:

$$\begin{aligned}
Send_AP_Demand[t-5,t] &= \frac{IfOutOct(t) - IfOutOct(t-5)}{\Delta(t)} \\
Recv_AP_Demand[t-5,t] &= \frac{IfInOct(t) - IfInOct(t-5)}{\Delta(t)} \\
Send_Client_Demand[t-5,t] &= \frac{IfInOct(t) - IfInOct(t-5)}{\Delta(t) \cdot NumClients(t)} \\
Recv_Client_Demand[t-5,t] &= \frac{IfOutOct(t) - IfOutOct(t-5)}{\Delta(t) \cdot NumClients(t)}
\end{aligned}$$

We assume uniform demands across all clients of an AP. We note that finer grained per-client demand information can be obtained by collecting and correlating `syslog` and `tcpdump` statistics (this approach was employed in [16]).

B. Predictability of Traffic Demands

The traffic-aware performance metrics require knowledge of traffic demands. In practice, the future traffic demands are not available, but have to be estimated based on historical demand information. The channel assignment is then based on predicted demands. This gives rise to two practical issues: (1) How to use historical data to identify trends in demands and to predict future demands reasonably accurately? (2) How to ensure that the resulting assignment is robust to mispredictions and to wild fluctuations in demands?

To answer these questions, we present a family of practical traffic-aware algorithms for channel assignment. These algorithms, each discussed below, offer varying degrees of tradeoffs between the two issues discussed above.

Exponentially-Weighted Average of Demand (EWMA). This approach predicts AP demands at time t by using a simple weighted moving average of demands observed in previous intervals. More recent demands are given greater weight: $Dem\_Pred(t) = w \cdot Dem\_Actual(t-1) + (1-w) \cdot Dem\_Pred(t-1)$. We set the weight $w = 0.9$. We use this to first estimate the AP demand estimates. We also estimate the number of active clients using EWMA. We then combine the two estimates to derive the predicted client demands.

Optimal for the Previous Interval (PREV). Here, the channel assignment for time t is simply the optimal channel assignment for the traffic demands in time $t-1$ (or the most recently sampled time interval, if there are no samples available for $t-1$). In other words, PREV is simply EWMA with $w = 1$. We note that compared to EWMA, PREV is more sensitive to short term traffic fluctuations.

Optimal Over a Time Window (PREV_N). There are several traffic patterns where PREV could be ineffective, e.g., periodic bursty traffic. Our next approach, PREV_N, tries to address this drawback by simultaneously optimizing the assignment for all traffic demands observed over a sizeable history window. In other words, given an optimization metric, PREV_N will derive a channel assignment that maximizes the *total* value of the metric for the traffic demands from the past N intervals: $Maximize : \sum_{i=1..N} Metric(Demands(t-N))$, where $Metric(Demands(t))$ denotes the value of the optimization metric under the traffic demands at t.

Peak Demand in a Window (PEAK_N). This is a variant of PREV_N: Instead of optimizing for all sets of demands in a time window, PEAK_N obtains the optimal channel assignment for the *worst-case* demand-set within the history window. This allows the channel assignment to be more responsive to sudden increases in aggregate network utilization.

We evaluate the effectiveness of these algorithms under a variety of settings in Section VII.

VI. Evaluation Methodology

To understand the benefits of traffic-awareness in different operating conditions, we evaluate the effectiveness of traffic-awareness using simulations based on both real and synthetic topologies and traffic. In Section VIII, we further evaluate its performance using testbed experiments. Below we describe the simulation methodology and datasets we use in our evaluation.

A. Simulation Methodology

We use the publicly available version 2.29 of NS-2 with support for multiple non-overlapping channels. We use either real traces or synthetic data to determine AP and client locations and their data rate. Unless otherwise stated, we use 802.11b and 11 Mbps medium bit rate with RTS/CTS enabled and transmission range set to 60 meters (with corresponding interference range = 120 m). We generate constant bit rate (CBR) traffic at a specified rate with data packet sizes of 1024 bytes. The channel assignment algorithm described in Section IV is applied to optimize the channel separation metrics. In order to evaluate the effectiveness of an assignment, we compute the *total throughput* over all connections.

B. Synthetic Scenarios

First, we use synthetic scenarios to understand when traffic-aware channel assignment is beneficial. We generate synthetic topologies and traffic traces using the evaluation methodology in [17], [18]. Specifically, we generate topologies that consist of 50 APs and 200 clients in a given area. Like [17], [18], we generate 15 random topologies, where each client has on average 4 APs in its communication range. Different from [17], [18], we generate two types of constant-bit-rate (CBR) UDP traffic to shed light on how traffic distribution affects the benefits of traffic-aware assignments. The two types of demands are (i) uniform random traffic demands and (ii) hotspot traffic demands. In uniform random traffic, each node is randomly assigned a demand from 0 to the maximum CBR throughput on a wireless link (3.6 Mbps for our NS-2 settings). In hotspot traffic demands, a specified number of “hotspots” are created; each of the hotspots is formed by randomly selecting an AP and all the other APs within its communication range. All APs in the hotspots have traffic demands uniformly distributed between 0 and 3.6 Mbps, and all other APs have traffic demands uniformly distributed between 0 and 10 Kbps.

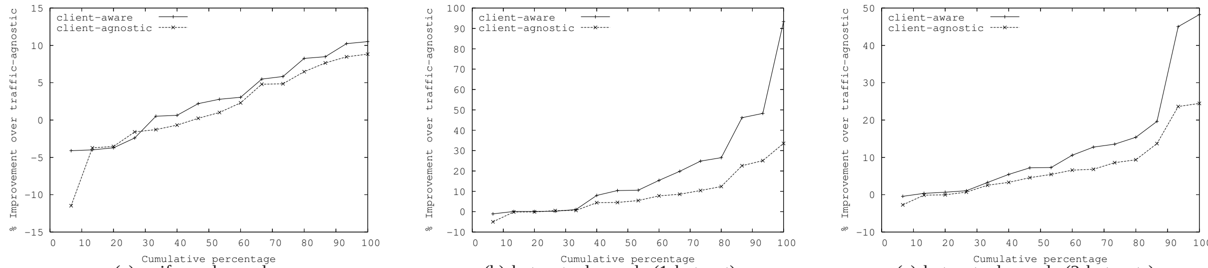
C. Trace-driven Simulation

In addition to synthetic scenarios, we also conduct trace-driven simulations over two publicly available data sets: the first was collected at Dartmouth College [10], [11] in 2004 and the second dataset was collected at the IBM T.J. Watson Research Center [12] in August 2002.

Dartmouth Traces. We analyze two weeks’ worth of Dartmouth SNMP data, collected between Feb 1st and Feb 15th, 2004. Our simulations start on day 10 of these traces and, unless otherwise noted, cover 2 full days. While the Dartmouth College traces covered several campus buildings, our evaluation and analysis focus on two specific buildings: “ResBldg94” and “LibBldg2”. These buildings contain 12 and 20 access points respectively. Other buildings of similar type (e.g. other ResBldg’s) had fewer access points.

The Dartmouth traces include *SNMP statistics* and *number of active clients per AP* sampled 5 minutes at all APs. In addition, the data contains *geographic x-y-z coordinates for the APs*. As described in Section V-A, we use the SNMP statistics and client-AP association information to derive AP and client-side demands (in Mbps) for every 5 minute interval. Also, we assume that clients associated with an AP are randomly distributed around the AP within a circle of radius 20m.

IBM Traces. Similar to the Dartmouth data, the IBM traces contained *SNMP statistics* and *number of active clients per AP*



(a) uniform demands (b) hotspots demands (1 hotspot) (c) hotspots demands (2 hotspots)

Fig. 3. Comparison of traffic-aware schemes against their traffic-agnostic counterparts in synthetic topologies.

for three different buildings: “SBldg”, “MBldg” and “LBldg”. Of the three, our study focuses on “MBldg”, which has 33 APs. Unlike the Dartmouth data, we did not have the locations of the APs. Instead, we constructed synthetic coordinates for the APs by placing them at hand-picked locations in a 5-storied building spanning a 235x100m lot. We analyze two weeks worth of SNMP data collected between Aug 1, 2002 and Aug 14, 2002. Our simulations start on day 11 of these traces and, unless otherwise noted, cover 2 full days.

Our trace-driven simulations progress in rounds, where a single round covers a given SNMP measurement interval. Within a round, we apply the channel assignment algorithm, as described in Section IV, to optimize the channel separation metrics. We quantify the effectiveness of an assignment by computing the aggregate throughput over all connections.

To study the importance of traffic-awareness, in our simulations, we focus on intervals with $\geq 50\%$ simultaneously active APs. We consider an AP to be active if the total volume of traffic it sends and receives exceeds 10Kbps. Also, in order to increase net utilization, we scale up the traffic demands in these intervals (on average, we scale 60X across all buidlings).

VII. Simulation Results

We now present our evaluation from NS-2 simulations. We first use synthetic topologies and traffic to understand when traffic-aware channel assignment is most beneficial. We quantify the effectiveness of a channel assignment by computing the total throughput achieved by all network flows under the assignment. As we will show, the benefit of traffic-awareness is larger when the load is imbalanced. Then we further compare different channel assignments using trace-driven simulations under accurate and inaccurate traffic demands.

A. Simulations on Synthetic Settings

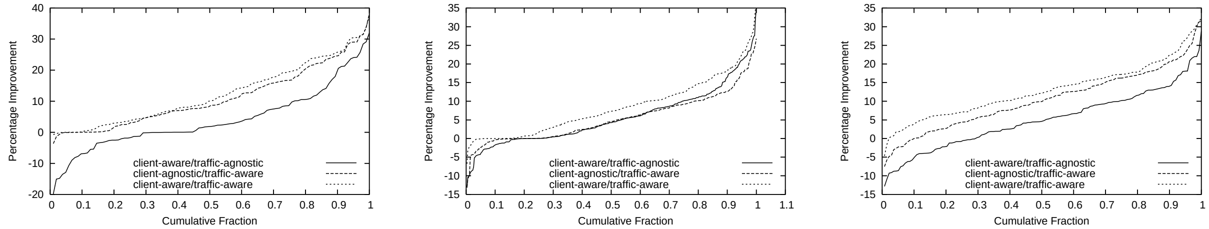
To understand the benefit of traffic-aware assignment, we create two types of traffic demands: (i) uniform, and (ii) hotspots, as described in Section VI-B. Figure 3 shows the CDF of improvement of traffic-aware channel schemes over their traffic-agnostic counterparts under each demand type. As we can see, the improvement of traffic-awareness is mostly within 10% under uniform demands, whereas the improvement under hotspots traffic is significantly higher, up to 93%. Moreover, the improvement in 1-hotspot case is higher than 2-hotspot case because the topologies are dense and with 2 hotspots (based on our generation) a large fraction of the network has high load and hence high channel utilization. Nevertheless we still observe up to 48% improvement in 2-hotspot case. These results suggest that the traffic-aware assignment is most useful for hotspots-style scenarios. The larger benefit of traffic-awareness under hotspots is realized because traffic-aware assignment aims to assign APs with high load to non-overlapping channels as much as possible; this significantly increases the throughput. Moreover, we observe the throughput (in absolute values) is highest when the channel assignment is both traffic-aware and client-aware.

Note that in Figure 3 there are a small number of cases where we observe negative throughput improvement. This is because the current channel separation metric (even after incorporating traffic and client awareness) is not perfect. For example, when two APs do not interfere but some of their clients interfere, the current metric only takes into account the interference between these clients but does not incorporate the additional effect of head-of-line blocking at APs caused by the interference experienced by their clients. We plan to further improve the metric, and we expect the benefit of traffic-awareness will be even larger under a metric that correlates more strongly with network performance.

B. Trace-Driven Simulation Results

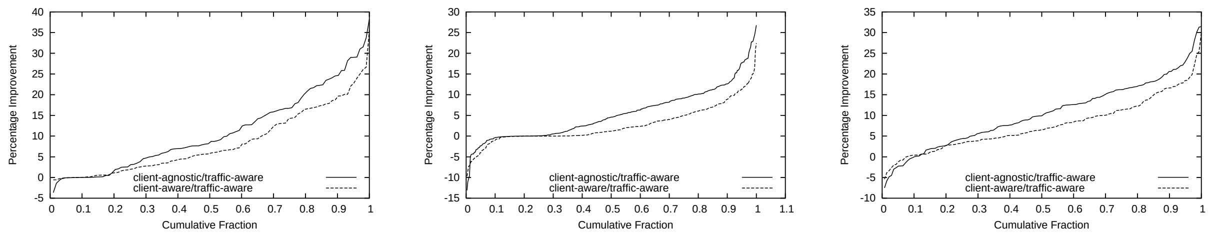
Next we compare different channel assignments using simulation based on real traffic traces, as described in Section VI.

1) Performance Benefits of Traffic-awareness: First we compare four channel separation metrics when assuming we have perfect knowledge of traffic-demands. Figure 4 shows cumulative distribution function (CDF) of performance improvement of various channel assignments against a traffic-agnostic/client-agnostic baseline. Although not shown here, we note that the average throughput improvement is 4.0%-5.7% after incorporating client-side information alone; it raises to 5.3%-11.1% by incorporating traffic-demands alone; and further to 8.3-12.4% by incorporating both traffic-demands and client-side information. As in the synthetic case, the amount of improvement is very traffic-dependent. When traffic is more evenly distributed, we see little improvement from traffic-aware assignment. When traffic is more heterogeneous, the improvement is larger, as much as 40%. Indeed, the classic Jain fairness metric computed for the demands corresponding to the maximum improvement of 40% in Figure 4(a) is almost 2X inferior compare to the fairness for median-case demands.



(a) ResBldg94 (b) LibBldg2 (c) MBldg

Fig. 4. Comparison of various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach as the baseline.



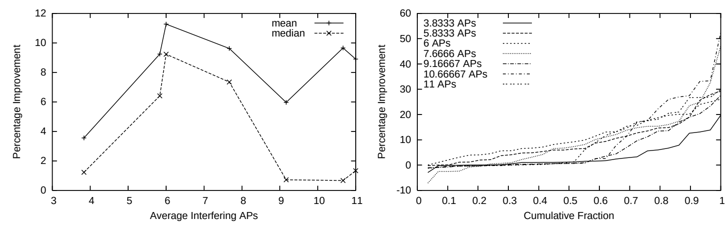
(a) ResBldg94 (b) LibBldg2 (c) MBldg

Fig. 5. Comparison of various traffic-aware schemes against their traffic-agnostic counterparts.

Approach	Fairness		
	ResBldg	LibBldg	MBldg
Traffic-agnostic	0.89	0.88	0.86
Traffic-unaware client-aware	0.90	0.90	0.87
Traffic-aware client-agnostic	0.92	0.89	0.88
Traffic-aware client-aware	0.92	0.91	0.87

TABLE I

IMPACT OF TRAFFIC-AWARENESS ON FAIRNESS



(a) Average improvement as a function of density (b) CDF of improvements

Fig. 7. Improvement in performance as a function of density for ResBldg. Figure (a) plots the average improvement in throughput performance. Figure (b) plots the CDF of the throughput improvement at different densities.

Figure 5 further compares the performance improvement of the two traffic-aware metrics against their traffic-agnostic counter-parts. The average improvement of client-agnostic traffic-aware metric over client-agnostic traffic-agnostic is between 4.6-10.1%, whereas the average improvement of client-aware/traffic-aware over client-aware/traffic-agnostic is 1.2 - 6.6%. The former improvement is larger because the baseline performance is worse. The largest improvement of traffic-awareness is over 35%, across either metric (Figure 5(a)).

2) Impact on Fairness: Next we compare different assignments in terms of their fairness, which is quantified using classic Jain's fairness index. As summarized in Table I, all the algorithms result in very similar fairness. This indicates that throughput improvement from traffic-aware assignment is not at the expense of fairness.

3) Impact of Network Density: Now we evaluate the impact of network density on channel assignments. Figure 7 shows the performance improvement when we vary transmission range, and consequently, the average number of interfering AP pairs. The improvement tends to first increase with density and then decrease. This is because when the network density is low, very few APs interfere with each other and all channel assignments yield similar throughput. When network density is high, a better channel assignment can allow more nodes to simultaneously transmit, thereby increasing total throughput. As network density increases further, all the channels are fully utilized everywhere regardless of channel assignments.

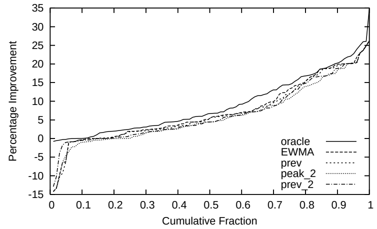
	EWMA	Previous	$Peak_2$	$Peak_4$
ResBldg	0.50	0.51	0.73	1.12
LibBldg	0.48	0.49	0.63	0.88
MBldg	0.75	0.93	1.05	1.21

TABLE II

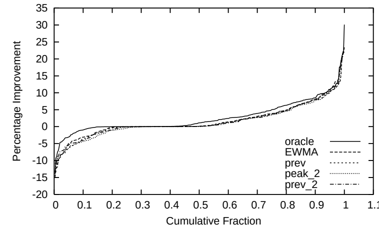
PREDICTION ERROR

4) Evaluation of Practical Traffic-aware Algorithms: In the previous evaluation, we assume that traffic-aware channel assignments have perfect knowledge of traffic-demands. In practice, such information is not known a priori, but has to be estimated based on historical information. A natural question arises: can the prediction error offset the potential gain of traffic-aware channel assignment? To answer this question, we first compute the error in predicting traffic demands using various prediction algorithms. We quantify the prediction error using mean absolute error (MAE), defined as $\frac{\sum_i |predict_i - actual_i|}{\sum_i actual_i}$. As shown in Table II, the best prediction is EWMA, which results in MAE ranging from 0.48 to 0.75. This prediction error is still quite significant. Large prediction errors are not surprising since wireless traffic at each AP has *low aggregation* and is much harder to predict than traffic in an ISP backbone. Such high variability in traffic poses challenges to traffic-aware assignment schemes.

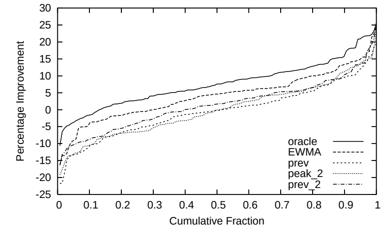
Next we evaluate the performance of channel assignment using predicted demands, and compare it with the case where the true demands are known (the "oracle"). Channel assign-



(a) ResBldg94



(b) LibBldg2



(c) MBldg

Fig. 6. Comparison of channel assignments using various prediction algorithms.

ment uses the traffic-aware, client-aware metric. As shown in Figure 6, the performance of the prediction algorithms closely tracks that of the oracle. Compared with the oracle algorithm, the degradation of predictive algorithms is mostly within 5% (e.g. see EWMA algorithm); compared with the traffic-agnostic algorithm, the improvement is still substantial, as high as 25-35%. This suggests that even though wireless traffic is hard to predict accurately, it is still possible to apply traffic-aware channel assignments, since they are reasonably robust against prediction errors. The robustness arises from the fact that traffic-aware channel assignment does not need accurate demands but rough spatial demand distribution so that it can allocate more channels to areas that need them most.

VIII. Testbed Results

In this section, we describe our testbed experiments and results. We set up a wireless testbed that consists of 12 DELL Dimensions 1100 PCs. The testbed is located on one floor of an office building (we run the experiments late at night to avoid interference with the resident wireless network). Each machine has a 2.66 GHz Intel Celeron D Processor 330 with 512 MB of memory, and runs Fedora Core 4 Linux. Each is equipped with 802.11 a/b/g NetGear WAG511 using MadWiFi. In our experiment, we use 802.11g. Half of the PCs act as APs and the other half act as clients, each AP has one client. We set the AP drivers in "Master" mode to emulate AP behavior.

As in the default configuration, the cards have RTS/CTS disabled and are set to the maximum transmission power. The data rate is fixed to 1 Mbps (autorate disabled). All the nodes are within interference range of each other.

We generate either constant-bit-rate UDP or TCP traffic from APs to clients with packet size of 1024 bytes. For both forms of traffic, we measure the throughput using nuttcp [19]. We enforce a specified demand in TCP traffic by utilizing the rate limiting function in nuttcp, which essentially places an appropriate upper-bound on TCP's congestion window. We use the same set of traffic demands for TCP and UDP and assume these demands are known a priori. For each scenario, we report the average throughput over 3 runs, where each run lasts 2 minutes and all flows start simultenously.

Table III compares throughput under traffic-unaware and traffic-aware channel assignments. We make the following observations. First, the throughput improvement is significant in many cases, with a maximum of 96.56% for UDP and up to 102.04% for TCP. Second, the throughput improvement has a strong correlation with "fairness index"; this is the Jain fairness index computed over the traffic demands. A lower index indicates more imbalance in traffic distribution, and results in larger improvement from traffic-aware channel assignment. These results are consistent with our simulation. Moreover, we observe that traffic-aware channel assignment not only benefits UDP traffic (e.g. streaming media or delay sensitive traffic), but also significantly improves TCP throughput (e.g. elastic large file downloads). Therefore traffic-awareness could benefit a wide variety of applications running over wireless links.

IX. Other Practical Issues

Infrastructure Support. To effectively incorporate traffic aware channel assignments, WLANs must deploy additional infrastructure to collect demand information, estimate client locations and mobility patterns, and to disseminate channel assignment decisions to APs in a timely manner. Common management tools, such as SNMP, coupled with recent infrastructure proposals for WLAN monitoring and management [6] could be employed for the first and third issues. For client locations, approaches such as RADAR can be employed [7]. We leave the actual implementation of a network monitoring and channel assignment infrastructure for future work.

Client-side Behavior. An important issue left unaddressed by our work is how clients respond to changes in channel assignment. Whenever an AP changes its channel, its clients will have to re-associate on the new channel. However, the 802.11 standard does not precisely define the re-association policy for clients. One approach is for the client to probe for APs using probe request packets. The APs can respond using a probe response packet (this is similar to AP's beacon packet). Alternatively, wireless stations can simply listen passively for beacons, which are transmitted every 100ms. The client associates with the AP and channel offering the highest RSSI. Although we do not quantify the impact of re-association on ongoing client transfers, we do expect that reducing the beacon interval size on APs (e.g., to 50ms) is a simple way to contain the impact on client performance, if any.

802.11a. Our analysis has focused on 802.11b and g networks which support fewer operating frequencies than technologies like 802.11a. It is conceivable that traffic aware channel assignment is less critical in 802.11a networks. However, as WLAN deployment densities grow, and as multiple independently-administered WLANs operate in close proximity of each other, we believe that static allocation of non-overlapping channels—no matter how many—is unlikely to offer good performance.

Traffic demands (AP1, AP2, AP3, AP4, AP5, AP6)	Throughput for traffic-aware assignment (Mbps)		Throughput for traffic-unaware assignment (Mbps)		Improvement over traffic-unaware	Fairness index
	Distribution	Total	Distribution	Total		
UDP Results						
(1.0, 0.33, 0.33, 0.5, 0.5, 0.33)	(0.78, 0.33, 0.33, 0.49, 0.48, 0.33)	2.75	(0.57, 0.33, 0.33, 0.50, 0.50, 0.33)	2.57	7.00%	0.82
(0.6, 0.2, 0.9, 0.6, 0.2, 0.9)	(0.54, 0.20, 0.67, 0.60, 0.20, 0.81)	3.01	(0.58, 0.20, 0.41, 0.60, 0.20, 0.56)	2.55	18.25%	0.80
(0.0, 0.0, 1.0, 1.0, 1.0, 1.0)	(0.0, 0.0, 0.83, 0.56, 0.83, 0.53)	2.75	(0.0, 0.0, 0.33, 0.47, 0.47, 0.53)	1.82	51.1%	0.67
(0.2, 0.0, 0.5, 0.2, 0.2, 1.0)	(0.20, 0.0, 0.5, 0.2, 0.2, 0.80)	1.90	(0.20, 0.0, 0.41, 0.20, 0.20, 0.42)	1.43	32.23%	0.54
(0.0, 0.0, 1.0, 0.0, 1.0, 1.0)	(0.0, 0.0, 0.81, 0.0, 0.80, 0.83)	2.44	(0.0, 0.0, 0.49, 0.0, 0.78, 0.36)	1.63	50.19%	0.50
(0.0, 0.0, 1.0, 0.0, 0.0, 1.0)	(0.0, 0.0, 0.81, 0.0, 0.0, 0.85)	1.66	(0.0, 0.0, 0.51, 0.0, 0.0, 0.3353)	0.84	96.56%	0.33
TCP Results						
(1, 0.33, 0.33, 0.5, 0.5, 0.3)	(0.76, 0.33, 0.12, 0.48, 0.38, 0.33)	2.41	(0.55, 0.33, 0.33, 0.5, 0.48, 0.33)	2.53	-4.4%	0.82
(0.6, 0.2, 0.9, 0.6, 0.2, 0.9)	(0.48, 0.2, 0.59, 0.60, 0.2, 0.74)	2.81	(0.48, 0.2, 0.08, 0.6, 0.2, 0.69)	2.25	24.43%	0.80
(0.0, 0.0, 1.0, 1.0, 1.0, 1.0)	(0.0, 0.0, 0.78, 0.62, 0.77, 0.31)	2.48	(0.0, 0.0, 0.08, 0.54, 0.29, 0.68)	1.58	56.53%	0.67
(0.2, 0.0, 0.5, 0.2, 0.2, 1.0)	(0.2, 0.0, 0.5, 0.2, 0.2, 0.77)	1.87	(0.2, 0.0, 0.38, 0.2, 0.2, 0.38)	1.36	37.61%	0.54
(0.0, 0.0, 1.0, 0.0, 1.0, 1.0)	(0.0, 0.0, 0.78, 0.0, 0.78, 0.77)	2.33	(0.0, 0.0, 0.06, 0.0, 0.78, 0.70)	1.54	50.93%	0.50
(0.0, 0.0, 1.0, 0.0, 0.0, 1.0)	(0.0, 0.0, 0.78, 0.0, 0.0, 0.77)	1.55	(0.0, 0.0, 0.07, 0.0, 0.0, 0.7)	0.76	102.04%	0.33

TABLE III

SUMMARY OF TESTBED EXPERIMENT RESULTS.

X. SUMMARY OF RESULTS AND CONCLUDING REMARKS

The centrality of channel assignment to improving the efficiency of spectrum usage in WLANs has been long recognized and well-studied. Several proposals have been made over the years, but they have all focused predominantly on static channel assignments that ignore traffic demands.

Our work explores the affect of dynamically adapting the channel assignment to prevailing traffic conditions. Using extensive simulations and live experiments, we show that *traffic-aware* channel assignment approaches could significantly improve the quality of the channel assignment in practice. We show that approaches that track both AP and client demands are clearly superior to those that ignore clients.

We perform a detailed study of the operating conditions under which traffic-awareness offers maximum benefit. We show that the benefits of the approach are tightly coupled to the deployment environment. For example, traffic-awareness is most helpful when traffic demands are concentrated at a small number of heavily-loaded APs located close to each other. The approach is of little use when traffic demands are uniform across the WLAN, or when most APs are located close to each other, or when the WLAN deployment is too sparse. Our testbed experiments shows that the benefits of traffic-awareness extend both to TCP as well as UDP traffic.

While a large portion of our study establishes the potential of traffic-awareness, we also consider several practical issues that might arise in real deployments. One such issue is that of predicting traffic demands. We discuss a variety of approaches to predict future traffic demands based on historical information, and use the predicted demands in channel assignment. One approach, EWMA, seems particularly promising, yielding performance that is within 5% of the best possible. While EWMA does not predicted demands with too great an accuracy, it can predict trends in the demands reasonably well.

Our paper establishes the importance of traffic-awareness to the management of wireless LANs. Although our focus has been on campus and enterprise networks, we believe that the central idea of this paper – traffic-awareness – is widely applicable to other scenarios such as multi-hop mesh networks and uncoordinated deployments.

REFERENCES

[1] Alcatel AirView Software. `http://www.alcatel.com`.
[2] Autocell. `http://www.propagatenetworks.com/product/`.
[3] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. *Overview and Principles of Internet Traffic Engineering, RFC 3272*, May 2002.
[4] D. O. Awduche. MPLS and traffic engineering in IP networks. *IEEE Communication Magazine*, pages 42–47, December 1999.
[5] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping For Capacity Improvement in IEEE 802.11 Ad Hoc Wireless Networks. In *ACM MobiCom*, Philadelphia, PA, August 2004.
[6] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. DAIR: A Framework for Managing Enterprise Wireless Networks Using Desktop Infrastructure. In *Workshop on Hot Topics in Networks (HotNets-IV)*, Nov 2005.
[7] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2002)*, pages 775–784.
[8] J. Case, M. Fedor, M. Schoffstall, and J. Darvin. A simple network management protocol (SNMP). In *Internet Engineering Task Force, RFC 1098*, May 1990. http://www.snmp.com/.
[9] G. J. Chaitin. Register allocation & spilling via graph coloring. In *Proc. of SIGPLAN*, 1982.
[10] Dartmouth campus-wide wireless traces. http://www.cs.dartmouth.edu/ campus/.
[11] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. of ACM MOBICOM '2004*, Sept. 2004.
[12] Wireless traces at IBM corporation. http://nms.lcs.mit.edu/ mbalazin/wireless/.
[13] Jim Grier. Assigning 802.11b Access Point Channels. `http://www.wi-fiplanet.com/tutorials/article.php/972261`.
[14] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proceedings of NSDI 2005*, Boston, MA, May 2005.
[15] Y. Lee, K. Kim, and Y. Choi. Optimization of AP Placement and Channel Assignment in Wireless LANs. In *Workshop on Wireless Local Networks (WLN), IEEE LCN*, Nov. 2002.
[16] X. Meng, S. Wong, Y. Yuan, and S. Lu. Characterizing flows in large wireless data networks. In *Proc. of ACM MOBICOM '2004*, Sept. 2004.
[17] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *INFOCOM06*, Barcelona, Spain, April 2006.
[18] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially-overlapped Channels not considered harmful. In *Proc. of ACM SIGMETRICS*, June 2006.
[19] nuttcp. `http://sd.wareonearth.com/~phil/net/ttcp/`.
[20] A. Raniwala and T. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE Infocom*, March 2005.
[21] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized algorithms for multi-channel wireless mesh networks. In *Proc. of ACM Mobile Computing and Communications Review (MC2R)*, April 2004.
[22] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, Englewood Cliffs, NJ.
[23] Simulated annealing. `http://www.cs.sandia.gov/opt/survey/sa.html`.
[24] X. Xiao, A. Hannan, B. Bailey, and L. Ni. Traffic engineering with MPLS in the Internet. *IEEE Network*, pages 28–33, March 2000.