Traffic-Aware Channel Assignment in Enterprise Wireless LANs

Eric Rozner⋆ Yogita Mehta⋆ Aditya Akella† Lili Qiu⋆
University of Texas at Austin⋆ University of Wisconsin at Madison†
{erozner,yamehta,lili}@cs.utexas.edu⋆ akella@cs.utexas.edu†

***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 WLANs. The state-of-the-art approaches assign channels statically, without considering prevailing traffic demands. In this paper, we show that the quality of a channel assignment can be improved significantly by incorporating observed traffic demands at APs and clients into the assignment process. We refer to this as *traffic-aware channel assignment*. We conduct extensive trace-driven and synthetic simulations and identify deployment scenarios where traffic-awareness is likely to be of great help, and scenarios where the benefit is minimal. We address key practical issues in using traffic-awareness, including measuring an interference graph, handling non-binary interference, collecting traffic demands, and predicting future demands based on historical information. We present an implementation of our assignment scheme for a 25-node WLAN testbed. Our testbed experiments show that traffic-aware assignment offers superior network performance under a wide range of real network configurations. On the whole, our approach is simple yet effective. It can be incorporated into existing WLANs with little modification to existing wireless nodes and infrastructure.**

I. Introduction

Enterprises and university campuses are deploying WLANs at a remarkable rate and effectively managing such networks has become increasingly important. The broadcast nature of wireless communication makes the task of supporting good end-user experience very difficult. Emerging trends such as rapidly growing densities and increasing traffic volumes only exacerbate this problem (see [13] for a detailed analysis). Traditionally, careful *channel assignment* has provided some respite to end-users. In the common case, network administrators conduct detailed site surveys and manually try various configurations to determine the right channel and placement for APs. The state-of-the-art research [16], [18] also offers similar *static* solutions. While there are other solutions for supporting better performance in dense deployments [3], channel assignment is attractive because it is simple and clients do not need to be modified.

Unfortunately, existing approaches to channel assignment are insufficient for enterprise WLAN deployments and usage patterns. Indeed, recent work has shown the traffic volumes in a WLAN can vary significantly both across APs and across time [13]. In the future, as more devices and newer applications contend for wireless access, the variability in traffic will increase further. Due to traffic variability in current and future networks, the performance of static channel assignment is bound to suffer.

Researchers in the wire-line world faced a similar problem when static routing weights were proven to be insufficient for achieving robust intra-domain routing. Several researchers advocated that routing weights be tuned to observed traffic demands [6], [7], [28]. Motivated by the vast 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. Our approach is simple: at regular intervals, collect traffic demand information and use it to determine the channel assignment. We espouse traditional channel optimization objectives and show how they can be modified to incorporate the WLAN traffic demands. Of course, computing optimal channel assignments for traffic-aware objectives is NP-Hard. Hence, we develop simple techniques (based on simulated annealing) for quickly computing close-to-optimal assignments. We show these channel assignments can closely track the prevailing network conditions.

To be effective, we must address a few practical issues. (1) The effectiveness of a channel assignment depends on the availability of an accurate interference map for the WLAN. Since wireless signal propagation and interference patterns are hard to predict using simple heuristics [1], we directly measure wireless interference using active probes. This is done at coarser time-scales than the collection of demand information. (2) While existing work assumes binary wireless interference, we find that in real networks interference across links may not be binary (e.g., two senders may carrier sense each other intermittently due to variation of RSS). We present simple and effective channel assignment schemes for handling non-binary interference. (3) Our approach requires timely and accurate estimation of traffic demands. For this, we simply leverage the SNMP network usage statistics that most APs export. In addition, we develop simple approaches for predicting upcoming traffic demands using only historical SNMP samples and extend our traffic-aware channel assignment algorithms to use these predicted demands. (4) Finally, we address the issue of the overhead experienced by clients when their APs switch channels frequently due to fluctuating traffic loads. We describe and evaluate a suite of simple approaches to minimize this overhead.

On the whole, the traffic-aware approach we propose requires few modifications to existing wireless nodes and infrastructure. It is effective and simple to use. In our evaluation, we first conduct extensive simulations over real topologies and traffic demands (available publicly at [15] and [8]), as well as over several synthetic settings. We start by considering 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 simulation results show that being traffic-aware could substantially improve the quality of a channel assignment in terms of total network throughput. 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 spatial distribution of traffic demands. Our key finding is that traffic-awareness offers the most benefit when the demands in a WLAN are highly skewed. We investigate the quality of traffic-aware assignments that are computed using predicted demands, and find that their performance is mostly within 5% of the ones obtained with access to perfect information. In addition, we also inject artificial errors into traffic demands, and our evaluation shows traffic-aware channel assignment is robust against these errors.

Finally, we implement and evaluate the traffic-aware channel assignment algorithms in a 25-node wireless testbed, deployed on two floors of an office building. We find that traffic-aware channel assignment is effective in real wireless networks under a range of network configurations. It benefits both TCP and UDP flows. Traffic-aware assignment also interacts well with multi-rate adaptation by reducing interference and allowing data communication to use higher data rates. In addition, we find that traffic-aware channel assignment not only improves average network performance, but also helps avoid highly inefficient channel assignments that could arise from traffic-agnostic approaches.

II. Related Work

Assigning channels to APs in WLANs has been a static, one-time approach [14]. First, network administrators conduct an "RF site survey" of the campus and determine the location and number of APs for adequate coverage. Then, the administrators manually configure the APs with 802.11's non-overlapping channels to ensure that close-by APs operate on different channels when possible. Our work shows that such static approaches can result in poor performance in the face of shifting traffic demands.

There are several research proposals for channel assignment in campus WLANs [16], [18]. Unlike our paper, none of them consider the benefit of tailoring the channel assignment to prevailing traffic demands. For example, Lee et. al [16] advocate identifying "expected high-demand points" in a given WLAN deployment and assigning channels to maximize signal strength at the demand points. This is still a static approach. Mishra et. al [18] 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.

Recently, several "spectrum management" products have been developed to automate channel assignment in WLANs. Some perform dynamic channel selection based on the current operating conditions (e.g. AutoCell [5] and AirView [4]). Others also offer interference mitigation via transmit power control and load balancing across APs. Due to their proprietary nature, little is known about the design of these products and the operating conditions they work best under. Our work provides a thorough analysis of these issues for traffic-aware channel assignment.

Next, we briefly review IP traffic engineering approaches and discuss how they motivate our work. Traffic demands have been shown to have tremendous utility for network provisioning and route optimization in ISP networks [6], [7], [28]. 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 the representative demands are derived. Inspired by these results from the IP wire-line world, we ask whether being traffic-aware has similar benefits for managing wireless network spectrum. We develop a parallel set of approaches for deriving traffic demand information in WLANs.

III. Traffic-Aware Channel Assignment

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

Given that modern 802.11 wireless technologies offer very few non-overlapping channels (e.g., both 802.11b and 802.11g offer 3 such channels: 1, 6, and 11), channel assignment can essentially be viewed as an optimization problem: what is the best way to allocate the available channels to BSSs so as to optimize a given metric or objective?

A good optimization metric should satisfy two important conditions: (i) it should be easy and efficient to compute given a channel assignment, and (ii) it should reflect WLAN performance. In Section III-A, we present an overview of metrics commonly used in channel assignment. We argue that these metrics suffer from key drawbacks and, therefore, fail to satisfy condition (ii) above. In order to address these drawbacks, the metrics should be *traffic-aware*, i.e. they should capture prevailing traffic demands in the WLAN. In Section III-A we show how to construct traffic-aware metrics.

Choosing an appropriate optimization metric is only part of the problem. Computing the optimal channel assignment, even for the simplest metrics, is known to be NP-hard [18]. In Section III-B, we develop efficient heuristics for computing close-to-optimal assignments for traffic-aware metrics.

A practical implementation of traffic-aware channel assignment must address a few key challenges such as how to measure wireless interference, how to cope with realistic wireless interference patterns, and how to measure and predict traffic demands.We discuss and address these challenges in Section III-C. Finally, we summarize the traffic-aware channel assignment approach using a flow-chart in Section III-D.

A. Optimization Metrics for Channel Assignment

It is appealing to directly use wireless network performance, such as throughput or delay, as optimization metrics. However, modeling wireless network performance is hard because interference is complicated and difficult to model. In this paper, we focus on the "channel separation" metric, which maximizes the difference in the channels of interfering nodes. This metric is simple to compute and reflects the goal of minimizing interference. While we apply traffic-awareness to the "channel separation" metric, we believe that traffic-awareness can be equally applicable to other optimization metrics to provide more efficient channel assignments.

The channel separation metric is computed as follows: Let C_i denote the channel assigned to AP i. Also, if APs i and j are within interference range of each other, define $Separation(i,j) = min(|C_i - C_j|, 5)$, otherwise $Separation(i,j) = 5$. We use 5 as an upper-bound of channel separation because channels 1, 6, 11 in 802.11b/g are considered orthogonal. Furthermore, separation values between 0 and 5 can be used to support partially-overlapping channels. Our evaluation focuses on orthogonal channels, and [20] can be consulted for a primer on partially-overlapping channels. Let A denote the set of APs. Then the channel separation objective is: $Maximize : \sum_{i,j \in A, i \neq j} Separation(i,j)$. This metric is easy to compute given the interference graph.

However, this metric fails to reflect the performance of the network due to two reasons: (1) The metric ignores whether wireless nodes are active. In fact, the nodes are assumed to always be active. In practice, some wireless nodes are more active than others. Since the number of available non-overlapping channels is very small (only 3 in 802.11b/g), incorporating the activity of nodes can result in better channel assignments. (2) Furthermore, the metric ignores clients completely. In practice, minimizing interference introduced by client transmissions is also important. Our analysis of real wireless traces shows that clients transmit a significant volume of traffic. As we show later, these two drawbacks result in poor channel assignments in terms of overall network performance. Due to the above two properties, we refer to the traditional metric as *traffic-agnostic, client-agnostic*.

1) Client-awareness: When the interference graph induced by clients is available, *client-aware* channel assignment becomes possible. The corresponding metric is: $Maximize : \sum_{i,j \in A \cup B, BSS(i) \neq BSS(j)} Separation(i,j)$. Here B denotes the set of clients in the network. Also, nodes i, j in the sum must belong to different BSSs. This metric is designed to capture the channel separation between any two interfering APs, any two interfering clients that are associated with different APs, and an interfering AP-client pair. Note, however, that the metric is still traffic-agnostic. Mishra et. al [18] propose a *traffic-agnostic, client-aware* metric similar to this.

2) Traffic-awareness: The previous two metrics do not take into account the actual traffic volumes or periods of activity of individual clients and APs. Thus, 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 mitigate interference at other active network locations.

In order to verify that traffic varies across BSSs, we examined the traffic demands at APs from publicly-available traces (circa 2004 [13]). While we omit the details for brevity, we found that traffic volumes could vary substantially both across APs and across time [26]. We observe a similar variation among client traffic. Such variation prevents traffic-agnostic metrics from fully exploiting the capacity of the wireless medium.

Incorporating traffic volumes and the activity of wireless nodes requires a simple change to the traffic-agnostic metrics. Before outlining this modification, we define the term *demand* informally. The *sending demand* of a node is the aggregate amount of data (excluding link-layer ACKs) it wishes to transmit 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 amount of data (excluding link-layer ACKs) the node wishes to receive from various transmitters.

To incorporate traffic-awareness into channel assignment, we simply need to ensure that interfering nodes with high individual demands (specifically the BSSs containing such nodes) are assigned to non-overlapping channels. However, to obtain an effective channel assignment, we must understand how the send and receive demands of interfering nodes affect each other. Observe that whenever two nodes A and B are in interference range, the transmissions of one node will affect 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 that can arise in hidden-terminal settings.

Using this insight, we scale the channel separation between A and B with the following "weight": $W_{A,B} = S_A \times S_B + S_A \times R_B + S_B \times R_A$, where S is the send demand, and R is the receive demand. 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 and B's transmissions interfering with each other. The second (third) term reflects the probability of A's (B's) transmissions interfering with B's (A's) receptions.

Using the above weights, we can define the following *traffic-aware, client-agnostic* metric: $Maximize : \sum_{i,j \in A, i \neq j} W_{i,j} \times Separation(i,j)$.

Similarly, we can define a *traffic-aware, client-aware* metric: $Maximize : \sum_{i,j \in A \cup B, BSS(i) \neq BSS(j)} W_{i,j} \times Separation(i,j)$.

B. Efficient Algorithms for Computing Channel Assignments

Since optimizing a channel assignment is NP-hard, we use simulated annealing (SA) [27] to obtain near-optimal assignments for each metric. SA is appropriate in this context since it can iteratively improve the solution while avoiding being stuck in local optima. To speed up convergence and achieve good performance, we use an informed initialization algorithm that is inspired by Chaitin's approach to the register allocation problem [11].

1) Initialization Algorithms: We first describe an initialization algorithm that does not consider traffic demands and treats every node equally. Then we extend it to account for different traffic demands at each node. The initialization *does not* take clients into account, irrespective of whether the metric in question is client-aware or client-agnostic. When client-aware metrics are used, we rely on SA in Section III-B.2 to effectively incorporate client-side information.

Figure 1 shows the algorithm for the traffic-agnostic case. The intuition of the algorithm is to defer channel assignment for APs that have many conflicts with other APs. 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 with degrees smaller 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 to the graph, and assign it with a channel that is different from all its neighbors up to this point. If a vertex cannot be assigned, mark it.
5) For the marked vertices, assign them a channel that results in minimum interference, where interference is calculated as the number of interfering APs assigned the same channel.

Fig. 1. Initialization algorithm for channel assignment.

To extend the initial assignment to the traffic-aware case, we do the following: First, we modify the degree used in step #2 and #3 by weighing 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 channel 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, otherwise $interference(i,j) = sent(j) + recv(j)$. We then choose the channel that results in the minimum value of $interference(i,j)$ summed over all $j \in A$ and $j \neq i$.

2) Further Improvement via SA: We further improve the initial channel assignment obtained above by using an iterative search. We have compared several options for the search, including random walk, SA, and greedy search. We found that SA offers faster convergence and better assignment.

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. If the new assignment is better, we update the current assignment to the new one. Otherwise, we update the current assignment to the new one with the probability $e^{(f_{new} - f_{curr})/T}$, where T is the current temperature, and f_{new} and f_{curr} are the values of objective functions under the new and current channel assignments. The temperature gradually decreases so initially we are more likely to accept a worse solution and avoid being stuck at local optima. As the temperature approaches 0, we progressively move in the direction of improving the objective function. We set the initial temperature to 10, and each iteration reduces temperature to 0.999 of the current value. We use 1000 iterations and the output is the best solution over all iterations. We note the execution time of this approach is sufficient for practical WLAN settings (e.g., for the traces we study, it takes well under 1 second for SA to compute the optimized metric).

C. Practical Issues

We address several practical issues in channel assignment.

1) Measuring the Interference Graph: The effectiveness of a channel assignment depends on the availability of an accurate interference map. Four measurement and modeling techniques [1] [25] [22] [2] have been proposed recently to estimate wireless interference. The first three schemes are based on the maximum throughput measurement when one or two links are active, while the last scheme sends coordinated probes at specific time instances, which introduces lower traffic overhead but requires fine-grained time synchronization. The first scheme [1] directly measures link-based interference using broadcast probes. The second and third schemes [25] [22] improve the scalability of the first approach by developing an interference model based on RSSI measurement. Each sender sends a series of broadcast probes, and all other nodes measure the received signal strength. Then a model is used to estimate the sending rate based on received signal strength and carrier sense threshold, and estimate the delivery rate based on SNR. In this way, only $O(N)$ broadcast probes are required for measuring interference in an N-node network. The two schemes differ in the type of interference they can model – [25] works for pairwise interference and broadcast transmissions, whereas [22] works for pairwise and non-pairwise interference and for broadcast and unicast transmissions. The fourth scheme, proposed by [2], sends coordinated probes from APs to clients. For example, APs $A1$ and $A2$ estimate the interference on links $A1 - C1$ and $A2 - C2$ by sending a probe on $A1 - C1$ and then sending a probe on $A2 - C2$ at the same time when $C1$ sends an ACK to $A1$. If $C1$'s ACK is not received, it indicates the two links interfere; otherwise, they do not interfere. To further enhance the robustness of this approach (e.g., packet collision caused by an accidental transmission from somewhere else, or data and ACK transmission time is slightly different), one can measure multiple times and use consistent collisions as the indication of interference.

Channel assignment and interference estimation are orthogonal. In our evaluation, we use the first approach due to its simplicity, but our channel assignment approaches can be directly combined with and benefit from other scalable and accurate interference measurement techniques. In the first scheme, we have one node, say A, broadcast packets as fast as it can for 1 minute. Let R_A denote A's broadcast rate when it broadcasts alone. Then, we have two nodes, say A and B, broadcast simultaneously as fast as they can for 1 minute. R_A^{AB} denotes A's broadcast rate when A and B are simultaneously sending. Similarly, R_B^{AB} denotes B's broadcast rate when A and B are simultaneously sending. We then compute $BR = \frac{R_A^{AB}+R_B^{AB}}{R_A+R_B}$. When BR is close to 1, it means that nodes A and B do not interfere. When BR is close to 0.5, it means that these two nodes take turns in transmitting packets and hence interfere with each other. Any values in between indicate different degrees of interference.

2) Handling Non-binary Interference: Wireless interference in real networks may not be binary and converting BR into a binary metric loses accuracy. Thus, we extend our channel assignment approach to work with the measured BR. Figure 2 outlines our extension. As it shows, we first convert BR to a value ranging from 0 to 1, where 0 indicates no interference, 1 indicates complete interference, and any values in between indicate partial interference. This value only depends on the locations of nodes A and B, so it is called $LocInterf$. In addition, we also compute interference across channels based on their channel separation, which is referred to $ChannelInterf$. As $LocInterf$, $ChannelInterf$ ranges from 0 to 1, where 0 means no interference, 1 means complete interference, and other values in between means partial interference due to partially overlapping channels. The final interference metric is the product of $LocInterf$ and $ChannelInterf$. The traffic-agnostic, client-agnostic assignment aims to minimize $\sum_{i,j\in A} OverallInterf(i,j)$, and the traffic-aware, client-agnostic assignment aims to minimize $\sum_{i,j\in A} OverallInterf(i,j) * W(i,j)$, where $W(i,j) = S_i \times S_j + S_i \times R_j + S_j \times R_i$ as defined in Section III-A.2. Similar modifications apply to the client-aware metrics. Under binary interference, the above non-binary objectives are the same as the channel separation metrics defined in Section III-A. Our simulation evaluation uses the channel assignment for binary interference, since NS-2 only has a binary interference model. Our testbed evaluation uses the channel assignment for non-binary interference and we observe it out-performs the binary interference-based assignment due to the presence of non-binary interference in real networks.

3) Estimating Traffic Demand Information: The computation of traffic-aware metrics requires current WLAN demand information. We approximate this using SNMP statistics.

$BR = min(1, max(0.5, BR))$; // ensure BR within range 0.5 .. 1
$LocInterf = 2 - 2 \times BR$; // map BR to range 0 .. 1
$ChannelDiff = min(|C_i - C_j|, 5)$;
$ChannelInterf = 1 - ChannelDiff * 0.2$;
$OverallInterf = ChannelInterf * LocInterf$;

Fig. 2. Handling non-binary interference.

Enterprises routinely employ SNMP-based [10] tools to monitor and manage their WLANs. Most commercial APs export an SNMP management interface that provides the following byte counts every five minutes: (1) 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*). To illustrate, we can calculate the send demands of APs and clients as $Send\_AP\_Demand[t-5,t] = \frac{IfOutOct(t)-IfOutOct(t-5)}{\Delta(t)}$ and $Send\_Client\_Demand[t-5,t] = \frac{IfInOct(t)-IfInOct(t-5)}{\Delta(t)\cdot NumClients(t)}$. Receive demands can be computed in a similar fashion. We note it is possible to obtain finer grained per-client demand information by correlating SNMP, `syslog`, and `tcpdump` statistics [17].

4) Predicting Traffic Demands: Traffic-aware channel assignment accurately reflects network performance only when *current* demand information is available. In practice, we can only use past information to predict the traffic demands at current or future time intervals. To address this issue, we present simple algorithms for estimating future demands based on historical measurements (e.g., the previous SNMP data). We can then use these predicted demands in channel assignment.

We must address two important issues: (1) How to use historical data to identify trends in demands and to predict future demands with reasonable accuracy? (2) How to enhance the robustness of the resulting assignment against significant variation in traffic demands? Next, we present a family of practical traffic-aware algorithms for channel assignment. These algorithms offer varying degrees of trade-offs between these issues, and we evaluate them in Section V.

Exponentially-Weighted Average (EWMA). This approach predicts AP demands at time t by using a weighted moving average of demands in previous intervals. More recent demands are given larger 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 EWMA to first estimate the AP demand and the number of active clients. Then we 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 $w = 1$. PREV is more sensitive to short term traffic fluctuations than EWMA.

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 history window. 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: $Optimize : \sum_{i=1..N} Metric(Demands(t-i))$.

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.

5) Limitations: The traffic-aware metrics do not capture multi-rate adaptation. Incorporating this factor can complicate matters because it requires real time measurement of the received signal strength and/or the rates at clients. Since our metrics do not capture multi-rate adaptation, we say they are "rate-agnostic". In Section VI, we evaluate the impact of ignoring multi-rate using testbed experiments. We find rate-agnostic traffic-aware channel assignment interacts well with multi-rate adaptation. When clients and APs are close to each other, traffic-aware assignment offers similar improvement with and without multi-rate adaptation. This is because in both cases almost all communications use the highest data rate. When clients and APs are farther apart, traffic-aware channel assignment can offer larger improvement under multi-rate adaptation, because it reduces interference and allows communication to use higher data rates.

D. Putting It All Together

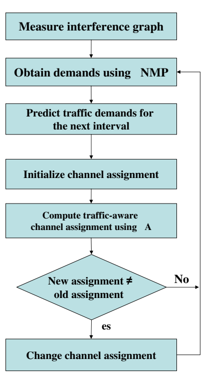


Fig. 3. Outline of traffic-aware channel assignment.

Figure 3 summarizes the steps in traffic-aware channel assignment. The first step, measuring the interference graph, can be conducted infrequently (e.g., a few times a day under light traffic load). All other steps are repeated at the timescale of collection of traffic demands, e.g., every 5 minutes. The traffic-aware channel assignment approach requires no modifications to the clients or the 802.11 standard. When clients are willing to cooperate (e.g., by measuring client-side interference and/or using an efficient re-association scheme described in Section VII), the benefit of our channel assignment increases further.

IV. Evaluation Approach

To understand the benefits of traffic-awareness in different operating conditions, we use two sets of experiments: (1) First, we conduct simulations using both real and synthetic traffic demands and WLAN topologies (Section IV-A). While the simulations allow us to explore the benefits of traffic-awareness in a range of operating conditions, they abstract away important real world effects. (2) To account for such effects, we implement our approach over a modest-sized wireless testbed and evaluate its performance using several field experiments. In Section IV-B, we provide details of our wireless nodes and the traffic demands we imposed in our testbed experiments. We describe the implementation in Section VI.

A. Simulation Methodology

We use NS-2 version 2.29 with support for multiple non-overlapping channels. We use 802.11b with 11 Mbps medium bit rate, RTS/CTS enabled, transmission range set to 60 m and a corresponding interference range of 120 m. We generate constant bit rate (CBR) UDP traffic at a specified rate with data packet sizes of 1024 bytes. Unless otherwise stated, the traffic is bi-directional and symmetric: the send demand at an AP is same as its receive demand. The traffic generated by APs is uniformly distributed to all clients. We study the effect on TCP traffic using testbed experiments (Section VI).

Since these are controlled simulations, we assume that the locations of all wireless nodes are known and use free-space propagation models [24] to estimate if two nodes interfere. In our simulations, all interference is binary. To evaluate the effectiveness of an assignment, we compute the *total throughput* over all connections.

1) 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 approach in [18], [20]. Specifically, we generate topologies that consist of 50 APs and 200 clients in a given area. Like [18], [20], we generate 15 random topologies, where each client has 4 APs on average in its communication range.

Different from [18], [20], we generate two types of CBR traffic to investigate how traffic distribution affects traffic-aware assignments. The two types of demands are (i) uniform random traffic demands and (ii) *hotspot* traffic demands. In uniform random traffic, each AP 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 hotspot is formed by randomly selecting an AP and all 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.

2) Trace-driven Simulation: In addition to synthetic scenarios, we also conduct trace-driven simulations over two publicly available wireless data sets from CRAWDAD: the first was collected at Dartmouth College [13], [15] in 2004 and the second dataset was collected at the IBM T.J. Watson Research Center [8] in August 2002. These simulations allow us to explore the benefits of traffic-awareness in real WLAN deployments with real traffic patterns.

Dartmouth Traces. We analyze the data collected between Feb. 10 and Feb. 12, 2004. Our analysis focuses on two buildings - "ResBldg94" and "LibBldg2" - containing 12 and 20 access points, respectively. Other buildings of similar type (e.g. other ResBldg's) have fewer access points.

The Dartmouth traces include SNMP statistics and the number of active clients per AP sampled every 5 minutes at all APs. We use the SNMP statistics and client-AP association information to derive AP and client-side demands (in Mbps) for every 5 minute interval. In addition, the data contains geographic coordinates for the APs. There is no client location information, so we assume that clients are randomly distributed around their APs within a circle of radius 20m.

IBM Traces. The IBM traces also contain SNMP statistics and the number of active clients per AP for three different buildings: "SBldg", "MBldg" and "LBldg". We focus 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 the data collected between Aug 11, 2002 and Aug 13, 2002.

Our trace-driven simulations progress in rounds, where a single round covers an SNMP interval. Within a round, we apply the channel assignment algorithm, as described in Section III-B, to optimize the channel separation metrics.

To study the benefits 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 10 Kbps. Also, while trace-driven simulation captures real usage patterns, its throughput is limited by the capacity of the current provision scheme (e.g., if the channel assignment in use was ineffective, the throughput of the traces would be too low to see benefits of improved channel assignment). To address this limitation, we scale up the traffic demands in these intervals (on average, we scale 60X across all buildings). Note the 60X scale is chosen to ensure the performance is not limited by the capacity of the existing deployment, even though we also observe benefits of traffic-aware assignment under much smaller scale-up values.

B. Experimental Approach

In addition to simulation, we also implement the channel assignment algorithms in a wireless testbed. Testbed evaluation is valuable because it allows us to evaluate the performance of different channel assignments using realistic wireless signal propagation, interference patterns, and multi-rate adaptation schemes.

We set up a wireless testbed that consists of 25 Dell Dimension 1100 PCs. The testbed spans two floors of an office building. Each machine has a 2.66 GHz Intel Celeron D Processor and runs Fedora Core 4 Linux. Each is equipped with an 802.11 a/b/g NetGear WAG511 card using MadWifi. In our experiments, we use 802.11g. There are 8 APs and 17 clients, with 7 APs having 2 clients and 1 AP having 3 clients. The loss rates from the APs to their clients vary from 0 to larger values (up to 40%). We run the experiments late at night to avoid interference with the resident wireless network. We evaluate both traffic-aware metrics (client-aware and client-agnostic) against a traffic-agnostic, client-agnostic baseline. We measure wireless interference in the testbed using broadcast probes (described in Section III-C.1) once before the experiments start and use the same interference graph for all experiments. This way the quality of a channel assignment is also subject to temporal variation in the interference graph, which is more realistic.

α	APi_1	APi_2	APi_3	APi_4	APi_5	APi_6	APi_7	APi_8
0.0	0.340	0.340	0.340	0.340	0.340	0.340	0.340	0.340
0.5	0.622	0.440	0.359	0.311	0.278	0.254	0.235	0.220
1.0	1.000	0.500	0.333	0.250	0.200	0.167	0.143	0.125
1.5	0.943	0.943	0.333	0.181	0.118	0.084	0.064	0.051
2.0	0.778	0.778	0.778	0.195	0.086	0.049	0.031	0.022

TABLE I

NORMALIZED ZIPFIAN DEMANDS IN THE TESTBED.

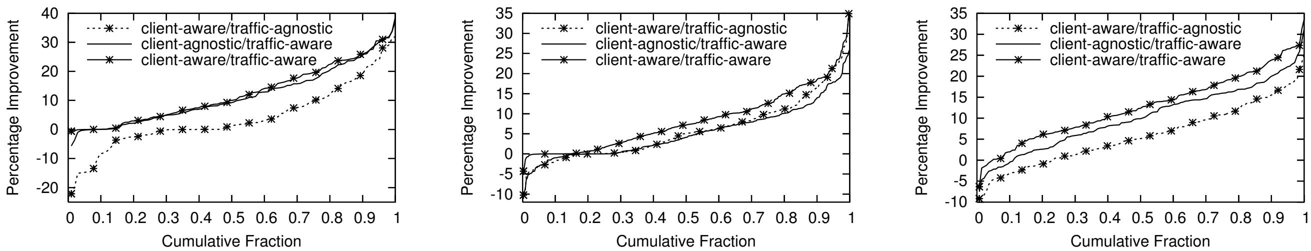
We impose Zipfian demands across the APs in our testbed. We try several different slopes for the Zipf-curve: a slope α means that the top i-th demand is proportional to $1/i^{\alpha}$; we vary α from 0 to 2, where 0 represents uniform demands and a larger α indicates more skewed demands. The demands generated from these slope values are listed in Table I. For each slope value, we evaluate 5 different random mappings of the generated demands to each AP and report the average throughput over these 1 minute runs. Each mapping can give a different traffic-aware channel assignment. We generate either CBR UDP or TCP traffic from APs to clients with packet sizes of 1024 bytes. For both types of traffic, we measure the throughput using nuttcp [21]. We enforce a specified demand in TCP traffic by utilizing the rate-limiting function in nuttcp, which 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.

V. SIMULATION RESULTS

We now present our evaluation based on NS-2 simulation. As mentioned earlier, we quantify the effectiveness of a channel assignment by computing the total throughput achieved by all network flows under the assignment. We have conducted more simulation and testbed experiments than we can present here. Refer to our technical report [26] for the complete results.

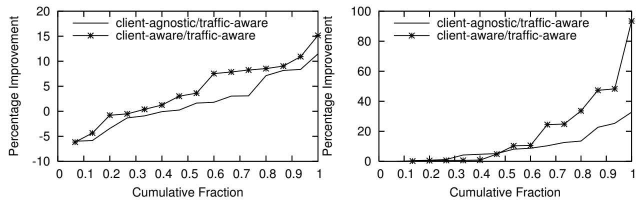
A. Simulations on Synthetic Settings

As described in Section IV-A.1, we create two types of demands to understand the benefit of traffic-aware assignment - uniform and hotspots. Figure 4 shows the cumulative distribution function (CDF) of improvement of traffic-aware channel schemes over their traffic-agnostic counterparts under each demand type. The CDF is plotted over the 15 random topologies that we simulated.



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

Fig. 5. Comparison of various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach as the baseline.



(a) Uniform demands (b) Hotspots demands (1 hotspot)

Fig. 4. Comparison of traffic-aware schemes against their traffic-agnostic counterparts in synthetic topologies.

The improvement of traffic-awareness is mostly within 15% under uniform demands, whereas the improvement under hotspots traffic is significantly higher: in 35% of the cases, the improvement is over 20%, and in 10% cases, the improvement is over 50%. The benefit is larger under hotspots than uniform demands because traffic-aware assignment aims to assign APs with high load to non-overlapping channels; this significantly increases the overall throughput when the demands are skewed. Also, we observe the throughput (in absolute values) is highest when the channel assignment is both traffic-aware and client-aware.

In Figure 4(a), there are a small number of cases having negative throughput improvement under traffic-aware assignments. This is because the current channel separation metric (even after incorporating traffic and client-awareness) is not perfect. For example, consider a setting where two APs do not interfere with each other but some of their clients do. The current metric only takes into account the interference between the clients, and ignores the additional effect of head-of-line blocking at APs caused by the interference at clients. We believe that our traffic-aware metrics can be improved further to correlate more strongly with network performance. We leave this for future work.

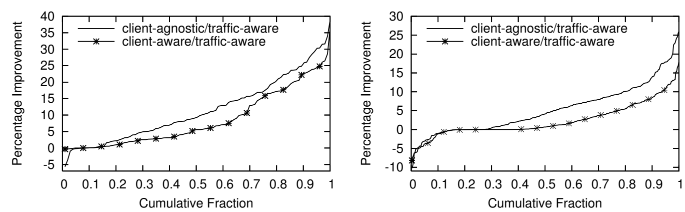
B. Trace-Driven Simulation Results

Next we compare different channel assignments using simulation based on real traffic traces described in Section IV-A.2.

1) Performance Benefits of Traffic-awareness: First we compare the four channel separation metrics assuming that we have perfect knowledge of traffic demands. Figure 5 shows a CDF of performance improvement of various channel assignments against a traffic-agnostic, client-agnostic baseline. Over the three buildings, the average throughput improvement ranges from 4.0%-5.9% after incorporating client-side information alone; it increases to 5.2%-11.5% by incorporating traffic demands alone; and it increases further to 8.3-12.8% by incorporating both traffic demands and client-side information.

As in the synthetic case, the extent of improvement is traffic-dependent. When traffic is more evenly distributed, we see little improvement from traffic-aware assignments. When traffic is more heterogeneous, the improvement is larger. For instance, we compute the classic Jain's fairness index for demands corresponding to the interval with the maximum improvement of 40% and for the interval corresponding to the median improvement of 10% in the ResBldg94 trace, where Jain's fairness index is defined as $(\sum x_i)^2/(n * \sum x_i^2)$ for demands $x_1 ... x_n$. We note that the fairness in the former case is almost one half of the fairness for median case demands. This further confirms the more imbalanced the traffic demands, the larger the benefit from using traffic-aware assignment.

Figure 6 compares the performance improvement of the two traffic-aware metrics against their traffic-agnostic counterparts. Over the three buildings, the average improvement of the traffic-aware, client-agnostic metric over traffic-agnostic, client-agnostic ranges from 5.2-11.5%, whereas the average improvement of traffic-aware, client-aware over traffic-agnostic, client-aware ranges from 2.3-8.6%. The former improvement is larger because the baseline performance is worse. The largest improvement from traffic-awareness is around 35% for ResBldg94 and around 25% for LibBldg2.



(a) ResBldg94 (b) LibBldg2

Fig. 6. Comparison of various traffic-aware schemes against their traffic-agnostic counterparts. Results from IBM's MBldg are omitted for brevity.

Approach	Fairness		
	ResBldg	LibBldg	MBldg
Traffic-agnostic/client-agnostic	0.89	0.87	0.85
Traffic-agnostic/client-aware	0.91	0.89	0.87
Traffic-aware/client-agnostic	0.89	0.90	0.86
Traffic-aware/client-aware	0.91	0.91	0.87

TABLE II

IMPACT OF TRAFFIC-AWARENESS ON FAIRNESS

2) Fairness: Next we ask how traffic-awareness affects fairness. We consider the ratio of the actual throughput obtained

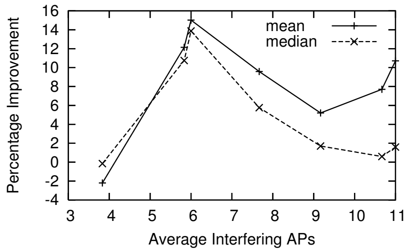


Fig. 7. Mean and median improvement of the traffic-aware, client-agnostic metric against its traffic-agnostic counterpart as a function of density for ResBldg.

at the AP to its original demand and compute Jain's fairness index over this ratio for all individual flows. As summarized in Table II, all the algorithms result in similar fairness. This suggests that traffic-aware assignment improves throughput without compromising fairness.

3) Impact of Zipf-distributed Demands: So far we have considered when the demand of an AP is equally distributed across its clients. A number of studies show that realistic user demands often exhibit Zipf-like distributions [9], [12]. So next we compare various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach when the AP's demand is distributed across its clients according to a Zipf-distribution. Note that the total traffic rate to and from each AP is the same in both cases. While we omit the figure for brevity (see [26] for details), compared with Figure 5(a), we observe the relative performance of the various algorithms is similar.

4) Impact of Network Density: We now study the relationship between the density of a WLAN deployment and the benefits of traffic-awareness. Figure 7 shows the performance improvement when we vary transmission range, and consequently, the average number of interfering AP pairs. The improvement first increases with density and then decreases. When the network density is low, very few APs interfere with each other and all channel assignments yield similar throughput. When network density is higher, 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 the assignment, and the benefit of traffic-awareness is reduced.

5) 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. Can the prediction error offset the potential gain of traffic-aware channel assignment?

	EWMA	PREV	$PEAK_2$	$PEAK_4$
ResBldg	0.48	0.49	0.70	1.02
LibBldg	0.43	0.47	0.57	0.80
MBldg	0.76	0.91	1.03	1.25

TABLE III
PREDICTION ERROR

To answer this question, we first compute the error in predicting traffic demands using various prediction algorithms described in Section III-C. We quantify the prediction error using mean absolute error (MAE), defined as $\frac{\sum_i |predict_i - actual_i|}{\sum_i actual_i}$. Table III shows the MAE in predicting the total demand (both send and receive demands) at the APs. As shown in Table III, the best prediction algorithm is EWMA, which results in MAE ranging from 0.43 to 0.76. 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.

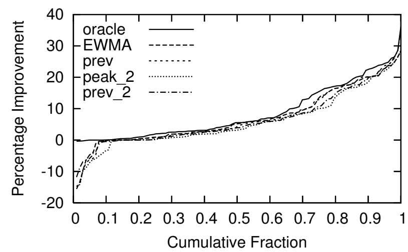


Fig. 8. Comparison of the traffic-aware, client-aware metric against its traffic-agnostic counterpart using various prediction algorithms in ResBldg94.

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"). We evaluate the improvement of the traffic-aware, client-aware metric over its traffic-agnostic counterpart for ResBldg94 (see [26] for the results of other traces). The performance of the prediction algorithms closely tracks that of the oracle. Compared with the oracle, the degradation of predictive algorithms is mostly within 6%. Compared with the traffic-agnostic algorithm, the improvement is still substantial. The median improvement for the client-agnostic channel assignment is 8.13%, while the client-aware channel assignment is 5.26%. The performance degradation for the client-agnostic channel assignment is less than the client-aware channel assignment because we have to predict the client-side demands and this further increases the prediction error for the latter.

Our evaluation suggests that even though wireless traffic is hard to predict accurately, it is still feasible to apply traffic-aware channel assignments, since the assignments are reasonably robust against prediction errors. The robustness arises from the fact that traffic-aware channel assignment does not need accurate demands but only the rough spatial demand distribution so that it can allocate more channels to areas that need them most. To lend further weight to this intuition, next we conduct simulations where we introduce Gaussian errors into demands. Figure 9 shows the CDF of performance improvement of the traffic-aware, client-aware channel assignment scheme against its traffic-agnostic counterpart when we add errors with different standard deviation (the performance is similar for the client-agnostic metric [26]). As we would expect, the performance improvement increases as the standard deviation of the error decreases. Moreover, we observe that even when the standard deviation is 0.5, the performance improvement is mostly close to that under no error. This is true for both client-agnostic and client-aware assignments. These results further demonstrate the robustness of traffic-aware assignment to a range of possible errors in the demand information.

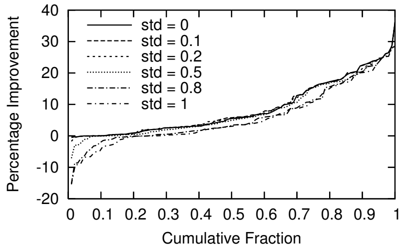
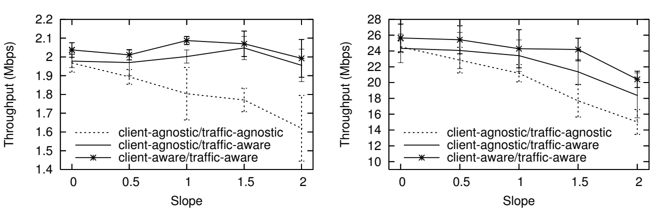


Fig. 9. Comparison of the traffic-aware, client-aware metric against its traffic-agnostic counterpart under Gaussian distributed errors with mean=0 and a varying standard deviation.

VI. Implementation and Experiment Results

We implement the channel assignment algorithms as follows. A centralized controller takes the traffic demands and the interference graph among wireless nodes as the input and computes channel assignments for the channel separation metrics defined in Section III. Then the controller disseminates the new channel assignment to the APs by establishing `ssh` connections through the back-end Ethernet connection and remotely sets the APs' channels using `iwconfig`. After all the APs' channels have been changed, the controller remotely starts the `nuttcp` program with the specified traffic demands to measure network performance.



(a) Fixed rate (b) Multi-rate

Fig. 10. Overall TCP network throughput in 25-node testbed, where the errorbars show the average and standard deviation.

Figures 10(a) and 10(b) show the overall TCP network throughput over 5 runs under fixed-rate and multi-rate, respectively (see [26] for the UDP results). We make the following observations. First, as we would expect, the traffic-aware, client-aware metric performs the best, and the traffic-aware, client-agnostic metric out-performs the traffic-agnostic, client-agnostic metric. Second, the throughput variance of the traffic-agnostic metric is generally higher than that of the traffic-aware metrics. This is because the traffic-agnostic metric ignores traffic demands, and different channel assignments may appear equally good according to the traffic-agnostic metric, but its actual performance varies significantly depending on whether the nodes with high demands happen to be assigned to non-interfering channels.

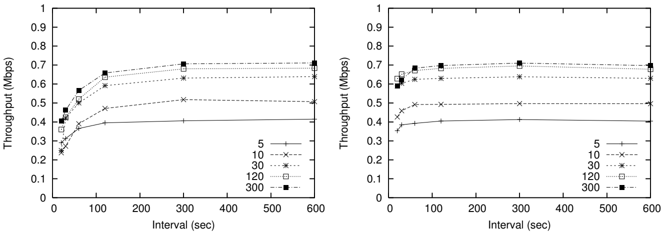
Figures 10(a) and 10(b) also show the improvement of traffic-awareness generally increases with the slope α. When $\alpha = 0$ (i.e., all traffic demands are the same), the traffic-aware, client-agnostic metric performs similarly as the traffic-agnostic, client-agnostic metric. The traffic-aware, client-aware slightly out-performs both the above metrics by accounting for client-side interference. As α increases, traffic becomes more concentrated on a smaller number of nodes and both traffic-aware metrics see larger improvement. Moreover, the improvement of traffic-awareness in some cases can be quite high: we observe up to a 1.69-fold increase for TCP/fixed-rate, and a 2.6-fold increase for TCP/multi-rate. The benefit of traffic-awareness is larger under the multi-rate because traffic-awareness can reduce interference and allow links to operate at higher data rates.

VII. Channel Switching

Channel switching causes two types of overhead: (i) delay incurred by an AP to change its channel - switching delay, and (ii) delay incurred for the clients to associate with the AP on its new channel - re-association delay. As reported in [19], the switching delay varies from 200 us on Intel's ProWireless to 10-20 ms on NetGear Atheros, Cisco Aironet, and Prism 2.5.

The re-association delay depends on the re-association scheme. A simple approach, which is implemented by MadWifi, is for wireless clients to scan all channels to find the AP with the highest RSSI. The re-association delay in this case tends to be long and is dominated by scanning time. To reduce this time, an AP can broadcast the new channel before switching so that the clients can directly switch to the new channel without performing scanning [23]. To protect against packet losses, the new channel information can be sent multiple times.

We refer to the above two re-association schemes as (i) MadWifi default implementation, and (ii) explicit notification. We evaluate the overhead of channel switching under these two re-association schemes using testbed experiments. In explicit notification, the AP broadcasts its new channel 5 times before switching to protect against packet losses. Figure 11 summarizes the results of a 10-minute TCP transfer between an AP and its client using Madwifi's default implementation and explicit notification, respectively. The UDP performance is similar and omitted for brevity [26]. The x-axis tracks how often the AP changes its channel. To evaluate the impact of frequent channel switching on different transfer durations, we use on/off traffic, where both on-periods and off-periods are exponentially distributed. Different curves in the graph correspond to different average on-periods, ranging from 5 to 300 sec. The average off-period duration is 5 seconds. The process is repeated until 10 minutes have elapsed. As shown in Figure 11(a), there is no degradation under the default re-association scheme when the switching interval is 2 minutes or higher. For smaller switching intervals, the overhead of the default scheme increases. In comparison, as we can see from Figure 11(b), the overhead under the explicit notification scheme is negligible for all switching intervals, including switching once per 20 seconds. These results suggest



(a) MadWifi default implementation (b) Explicit notifications

Fig. 11. Channel switching overhead for TCP under two re-association schemes.

the re-association overhead is negligible under the explicit notification; even for the default scheme, switching once every 5 minutes, as considered in this paper, incurs no performance penalty.

An orthogonal approach to further minimizing the impact of channel switching is to reduce the number of APs that change channels. There are several approaches we can use. The first approach is to apply the new channel assignment to the real network only if it improves the optimization metric by a threshold θ. The second approach is to use the channel assignment from the previous interval as the starting point in the SA search and limit the number of channel switches by controlling the number of iterations in SA. This will bias the outcome of the search in favor of assignments that are only slightly different from the current channel assignment. Our evaluation results (refer to [26] for details) show that this approach reduces the number of channel changes without compromising the performance. For example, when the number of SA iterations is limited to 5, the performance improvement is close to that without this limitation, and on average only 0.84 APs are required to switch their channel per interval.

VIII. Summary

The importance of channel assignment for improving the efficiency of spectrum usage in WLANs has been well-studied. Different from the previous work, our work explores the effect of dynamically adapting the channel assignment to prevailing traffic conditions. Using extensive simulations and testbed experiments, we show that *traffic-aware* channel assignment approaches could significantly improve the quality of a channel assignment in practice.

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 the WLAN deployment is too sparse. Our testbed experiments show that the benefits of traffic-awareness extend to both TCP and UDP traffic and both fixed-rate and multi-rate adaptation.

Our paper establishes the importance of traffic-awareness to the management of WLANs. 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.

Acknowledgement

We thank CRAWDAD for providing Dartmouth campus and IBM wireless traces, and anonymous reviewers for their helpful comments. This research is sponsored in part by National Science Foundation grants CNS-0546755, CNS-0627020, and CNS-0626889.

References

[1] S. Agarwal, J. Padhye, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *Proc. of Internet Measurement Conference (IMC)*, Oct. 2005.
[2] N. Ahmed and S. Keshav. SMARTA: a self-managing architecture for thin access points. In *Proc. of CoNext*, Dec. 2006.
[3] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self management in chaotic wireless deployments. In *Proc. of ACM MOBICOM*, Sept. 2005.
[4] Alcatel AirView Software. `http://www.alcatel.com`.
[5] Autocell. `http://www.propagatenetworks.com/product/`.
[6] D. O. Awduche. MPLS and traffic engineering in IP networks. *IEEE Communication Magazine*, pages 42–47, Dec. 1999.
[7] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. *Overview and Principles of Internet Traffic Engineering, RFC 3272*, May 2002.
[8] M. Balazinska and P. Castro. CRAWDAD data set IBM/Watson. `http://crawdad.cs.dartmouth.edu/ibm/watson`, Feb. 2003.
[9] L. Breslau, P. Cao, L. Fan, G. Philips, and S. Shenker. Web caching and Zipf-like distributions: evidence and implications. In *In Proc. of IEEE INFOCOM*, Mar. 1999.
[10] J. Case, M. Fedor, M. Schoffstall, and J. Darvin. A simple network management protocol (SNMP). In *Internet Engineering Task Force, RFC 1098*, May 1990.
[11] G. J. Chaitin. Register allocation & spilling via graph coloring. In *Proc. of ACM SIGPLAN*, 1982.
[12] M. Chesire, A. Wolman, G. M. Voelker, and H. M. Levy. Measurement and analysis of a streaming-media workload. In *Proc. of USITS*, Mar. 2001.
[13] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. of ACM MOBICOM*, Sept. 2004.
[14] Jim Grier. Assigning 802.11b Access Point Channels. `http://www.wi-fiplanet.com/tutorials/article.php/972261`.
[15] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace Dartmouth/campus. `http://crawdad.cs.dartmouth.edu/dartmouth/campus/snmp/fall0304`, Nov. 2004.
[16] Y. Lee, K. Kim, and Y. Choi. Optimization of AP Placement and Channel Assignment in Wireless LANs. In *Proc. of IEEE LCN*, Nov. 2002.
[17] X. Meng, S. Wong, Y. Yuan, and S. Lu. Characterizing flows in large wireless data networks. In *Proc. of ACM MOBICOM*, Sept. 2004.
[18] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *Proc. of IEEE INFOCOM*, Apr. 2006.
[19] A. Mishra, V. SHrivastava, D. Agarwal, S. Banerjee, and S. Ganguly. Distributed Channel Management in Uncoordinated Wireless Environments. In *Proc. of ACM MOBICOM*, Sept. 2006.
[20] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially-overlapped Channels not considered harmful. In *Proc. of ACM SIGMETRICS*, Jun. 2006.
[21] nuttcp. `http://sd.wareonearth.com/~phil/net/ttcp/`.
[22] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *Proc. of ACM MOBICOM*, Sept. 2007.
[23] I. Ramani and S. Savage. SyncScan: Practical fast hand-off for 802.11 infrastructure networks. In *Proc. of IEEE INFOCOM*, Mar. 2005.
[24] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2nd edition, Dec. 2001.
[25] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference. In *Proc. of ACM SIGCOMM*, Sept. 2006.
[26] E. Rozner, Y. Mehta, A. Akella, and L. Qiu. Traffic-Aware Channel Assignment in Enterprise Wireless LANs. In *University of Texas at Austin Technical Report TR-07-23*, 2007.
[27] Simulated annealing. `http://www.cs.sandia.gov/opt/survey/sa.html`.
[28] X. Xiao, A. Hannan, B. Bailey, and L. Ni. Traffic engineering with MPLS in the Internet. *IEEE Network*, pages 28–33, Mar. 2000.