Traffic-Aware Channel Assignment in Enterprise 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 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 assignments 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 the wireless nodes or infrastructure.**

I. Introduction

Enterprises and university campuses are deploying WLANs at a remarkable rate and effectively managing such networks becomes 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 [15] 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 [20], [22] also offers similar *static* solutions. While there are other solutions for supporting better performance in dense deployments [5], channel assignment is attractive due to its simplicity and no need for client modification.

Unfortunately, existing approaches to channel assignments are insufficient for enterprise WLAN deployment and usage patterns. Indeed, recent work has shown the traffic volumes in a WLAN can vary significantly both across APs and across time [15]. 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 [8], [9], [32]. 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 [3], we directly measure wireless interference using active probes. This is done at coarser time-scales than 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 sometimes but not always 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 very 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 [13] and [16]), 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 within 5% of the ones obtained with access to perfect information. In addition, we also inject artificial errors to traffic demands, and our evaluation shows that the 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 very 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 very inefficient channel assignments that could arise from traffic-agnostic approaches.

II. Related Work

We first review past work on the channel assignment problem. Then we consider the channel hopping approach which has been used to leverage the benefit of the entire frequency spectrum. Finally, we discuss related work in IP traffic engineering.

A. Channel Assignment

Channel assignment for improving the efficiency of spectrum usage is a well-studied problem. In particular, the problem has received much attention in the context of cellular networks [19]. In general, approaches for cellular channel allocation are unsuitable for our purposes (i.e., traffic-aware channel assignment in WLANs): Cells in a cellular network are arranged in a very regular fashion and have uniform, large coverage areas, unlike the regions covered by indoor access points (APs). As a result, channel assignment in cellular networks is a static, one-time task. In contrast, depending on the number and location of clients, load on APs and the presence or movement of obstacles, channel assignment across WLAN APs may need to change over time.

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 [17]: 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 [4], [20], [22], [23]. However, unlike our paper, none of them consider the benefit of tailoring the channel assignment to prevailing traffic demands. For example, Lee et. al [20] 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 [22] argue that AP-centric channel assignment approaches (like [20]) capture the interference amongst the APs, but do not account for the interference observed due to clients. They identify scenarios where client-AP and client-client interference reduces the throughput of the system. They propose 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. They identify the interference at the client in two ways: (1) considering all the APs in the range of the client and (2) considering all the APs which are in direct range of some station, where the station is either the client, or an AP within range of the client. This set of APs is called the conflict set and the channel assignment problem is formulated as a conflict set coloring problem with the aim of minimizing the conflicts amongst the clients. A randomized search algorithm is used to find an efficient channel assignment. 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.

The channel assignment scheme used in [4] is similar to [22]. Ahmed et al. [4] use a conflict graph to represent inter-AP, AP-client and inter-client interference. An utility function is defined as per the requirements of the system. The channel assignment is carried out in two phases. In the first phase, only the conflicts amongst the APs are considered. A randomized search algorithm is used to assign channels amongst the AP so that the utility of the system is maximized. In the paper, the authors consider utility in terms of reducing the total number of access-point conflicts. In the second phase, conflicts involving the clients are considered. Every AP locally tries to change its channel so that client conflicts are reduced, keeping constant the number of inter-AP conflicts. In our approach, we propose a new utility function which takes into account the varying traffic at APs and the clients.

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 [7] and Alcatel OmniAccess AirView Software [6]). 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. Raniwala et al. [27], [28] address the limitations of a single-NIC architecture where the entire mesh network has to operate on a single channel. They propose equipping mesh network nodes with multiple network interface cards to utilize multiple orthogonal channels. The different cards can operate on different channels. The approach must first decide which interface is used to communicate with a set of neighbors and then which channel is assigned to that interface. The goal here is to ensure that neighboring nodes are assigned to the same channel. 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 mesh network settings as well.

B. Channel Hopping

Channel hopping is another approach which has been used to improve the performance of wireless networks [10], [23]. In the channel hopping approach, the wireless nodes spend a fixed amount of time on a single channel, called a slot, and then switch to a subsequent channel as given by the hopping sequence. In the WLAN scenario, the hopping sequence is defined for the APs and all the clients associated to an AP change their channels along with it. Similar to the channel assignment schemes, the channel hopping approach has been studied for both the multi-hop mesh networks and WLANs.

Bahl et al. [10] advocate a new link-layer mechanism called SSCH, wherein neighboring mesh nodes perform synchronized channel hops to better exploit frequency diversity. As this scheme is applied for multihop mesh networks, the focus of the their channel scheduling scheme is to ensure nodes are synchronized in a slot and there is less overlap amongst nodes not communicating with each other.

In [23], the authors use channel hopping to solve the channel assignment problem for WLANs. The AP computes its hopping sequence such that its observes less interference. It first obtains the hopping sequence of the interfering APs. For each slot, the AP finds the appropriate channel which minimizes interference. Channel hopping ensures that one AP is not associated with a bad channel for a long time and the network as whole uses a 'good' static assignment. The evaluation results presented in the paper show that using channel hopping improves the fairness of the system, but degrades the throughput when compared to the channel assignment approach in [22].

The throughput under a channel hopping with hopping sequence: C_1, C_2, ... C_n is the average throughput over the channels used in the sequence. In comparison, channel assignment aims to assign the best combination of channels, which is likely to be better than the average throughput used over all channel sequences. We mainly aim to improve the performance of wireless LANs and hence focus on the channel assignment. We later show in our evaluation that using traffic information for computing intelligent channel assignments does not degrade the fairness in throughput allocation at APs.

C. Traffic Engineering in ISP Networks

Traffic demands have been shown to have tremendous utility for network provisioning and route optimization in ISP networks [8], [9], [32]. 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. For example, Agrawal et al. [2] use a traffic matrix in a one-hour window during daily peaks as the representative demand. Zhang et al. [33], [34] consider multiple representative traffic matrices and find an optimal set of routes to minimize expected or worst-case cost for these representative matrices. TeXCP [18] and MATE [14] conduct online traffic engineering and react to instantaneous traffic demand.

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. Traffic-Aware Channel Assignment

The goal of channel assignment is to ensure that wireless nodes belonging to distinct Basic Service Sets (BSSs), but within interference range, operate on distinct channels whenever possible. A wireless Basic Service Set (BSS) includes an access point (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 the 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 a part of the problem. Computing the optimal channel assignment, even for the simplest metrics, is known to be NP-hard [22]. 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, in Section III-D, we summarize the traffic-aware channel assignment approach using a simple flow-chart.

A. Optimization Metrics for Channel Assignment

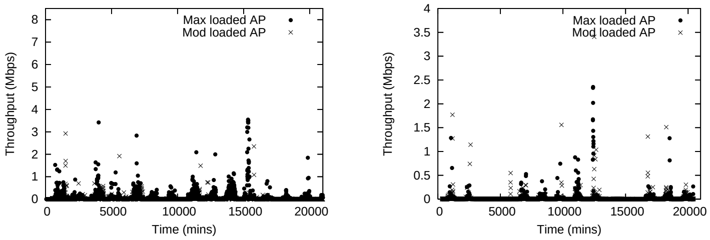
Common optimization metrics attempt to quantify the extent of interference in a WLAN due to a given channel assignment. One example is the "channel separation" metric, which maximizes the difference in the channels of interfering nodes.

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 as orthogonal. If A denotes the set of APs, then the channel separation objective is:

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

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 the 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



(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).

channels is very small (only 3 in 802.11b/g), taking the activity of nodes into account can result in better channel assignments. (2) 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 and 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 [22] propose a traffic-agnostic, client-aware metric similar to this one.

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 [15]). Figure 1 shows that traffic volumes could vary substantially both across APs and across time. We observe similar variation among client traffic. Such variation prevents traffic-agnostic metrics from fully exploiting the capacity of the wireless medium.

Incorporating traffic volumes and activity of wireless nodes requires a simple change to the client-aware metric. 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) terms 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-aware* metric:

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

Similarly, we can define a traffic-aware, client-agnostic metric:

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

B. Efficient Algorithms for Computing Channel Assignments

Since optimizing channel assignment is NP-hard [22], we use simulated annealing (SA) [31] 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 achieve good performance and to speed up the convergence, we use an informed initialization algorithm that is inspired by the Chaitin's approach to the register allocation problem [12].

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-B2 to effectively incorporate client-side information.

Figure 2 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. 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.

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: $degree(i) =$

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.

$\sum_{j \in G} interfere(i,j)$, 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, otherwise $interference(i,j) = sent(j) + recv(j)$. We then choose the color that results in the minimum value of $interference(i,j)$ summed over all $j \in A$ and $j \neq i$.

2) Further Improvement via Simulated Annealing (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 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 we are more likely to accept a worse solution initially and avoid being stuck at local optimal. 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., it takes well under 1 second for SA to compute the optimized metric value in the traces we study).

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. Three measurement and modeling techniques [3], [4], [30] have been proposed recently to estimate wireless interference. The first scheme [3] directly measures link-based interference using broadcast probes. This is the approach we use for our evaluation due to its simplicity. The second scheme improves 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 N-node network. The third scheme, proposed by [4], 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 sends a probe on $A2-C2$ at the same time when $C1$ sends ACK to $A1$. If $C1$'s ACK is not received, it indicates 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 sent time is slightly different), one can measure multiple times and use consistent collisions to indicate interference.

We choose the first approach due to its simplicity. 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. To apply the channel assignment metrics in Section III-A, we convert the measured BR to a binary interference metric as follows: When $BR > 0.9$, the two nodes are considered not to interfere with each other; otherwise, they are considered to interfere. This interference information is then directly used as the input to the channel assignment algorithms.

The probing-based approach assumes that nodes are immobile. APs can be safely assumed to be stationary, but not clients. Hence this approach may not be effective at capturing the interference graph induced by clients, which impacts the traffic-aware, client aware metric. While the traffic-aware, client-aware metric gives the best performance, our evaluation shows that traffic-awareness alone (i.e. traffic-aware, client-agnostic metric) can offer significant improvement compared with any traffic-agnostic metric. Finally, we note that it is possible to accommodate client-awareness partially using radio resource management techniques, such as 802.11K client report [1].

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 3 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. 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-A2. Similar modifications apply to traffic-agnostic, client-aware and traffic-aware, client-aware metrics. Note that our simulation evaluation uses the channel assignment for binary interference, since NS-2 only has binary interference model and in such cases the performance of channel assignments for non-binary interference is similar to those for binary interference. Our testbed evaluation uses the channel assignment for non-binary interference, and we observe they out-perform binary interference-based assignment due to presence of non-binary interference in real networks.

$BR = min(1, max(0.5, BR))$; // 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. 3. Handling non-binary interference.

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

Enterprises routinely employ SNMP-based [11] 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 [21].

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 the past information to predict the traffic demands at the 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 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 reasonably accurately? (2) How to enhance the robustness of 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. 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 $w = 1$. 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: $Optimize : \sum_{i=1..N} Metric(Demands(t-N))$.

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 two key factors: (1) multi-rate adaptation and (2) the dependence of wireless cell capacity on the number of clients and their transmission rates. Incorporating these factors can complicate matters since it requires real time measurement of the received signal strength and/or the rates at clients.

Since our metrics do not capture multi-rate adaption, 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

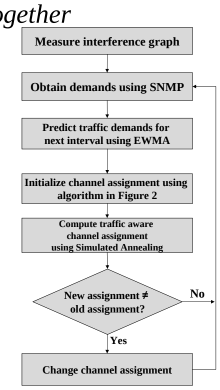


Fig. 4. Outline of traffic-aware channel assignment.

Figure 4 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 standard. When clients are willing to cooperate (e.g., by measuring client-side interference and/or using more 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 WLANs 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 the publicly available version 2.29 of NS-2 with support for multiple non-overlapping channels. We use 802.11b and enforce 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. Unless otherwise stated, the traffic is bi-directional (from APs to clients and vice versa) 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 in our testbed experiments (Section VI). We use the Simulated Annealing approach of Section III-B to optimize the channel separation metrics.

Since these are controlled simulations, we assume that locations of all wireless nodes are known and use free-space propagation models [29] to estimate if two nodes are interfering with each other. 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 [22], [24]. Specifically, we generate topologies that consist of 50 APs and 200 clients in a given area. Like [22], [24], we generate 15 random topologies, where each client has, on average, 4 APs in its communication range.

Different from [22], [24], 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 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 as follows. Each hotspot 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. Each send demand is simulated by creating CBR flows from an AP to its client and each receive demand is simulated by creating flow from client to its AP.

2) Trace-driven Simulation: In addition to synthetic scenarios, we also conduct trace-driven simulations over two publicly available wireless data sets: 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 [16] 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 10th and Feb 12th, 2004. In our analysis, we focus 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 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. Similar to the Dartmouth data, the IBM traces contain SNMP statistics and 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 a given SNMP measurement interval. Within a round, we apply the channel assignment algorithm, as described in Section III-B, to optimize the channel separation metrics. As mentioned earlier, we quantify the effectiveness of an assignment by computing the aggregate throughput over all connections.

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, it has a major limitation – its throughput is limited by the capacity of the current provision scheme (e.g., if 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 that 60X scale up is chosen to ensure that the performance is not limited by the capacity of the existing deployment, even though we also observe benefits of traffic-aware assignment under 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 with realistic wireless signal propagation, interference patterns, and multi-rate adaptation schemes.

We set up a wireless testbed that consists of 25 DELL Dimensions 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 802.11 a/b/g NetGear WAG511 using MadWiFi. We run the experiments late at night to avoid interference with the resident wireless network. We conduct two sets of experiments. (1) For the first experiment, we use a subset of our testbed (12 nodes). Half of the PCs act as APs and the other half act as clients, and each AP has one client. We construct several "toy" demands for our smaller scale testbed. In the small testbed, we use a binary interference graph and evaluate the client-agnostic, traffic-aware metric against the client-agnostic, traffic-aware approach. (2) For the second experiment, we use the entire testbed. There are 8 APs and 17 clients, with all but one AP having 2 clients. The loss rates from the AP to its clients vary from 0 to larger values (up to 40%). We evaluate both traffic-aware metrics (client-aware and client-agnostic) against a client-agnostic, traffic-agnostic baseline. 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 V. 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 measure non-binary wireless interference in the second testbed using broadcast probes (described in Section III-C1) once before the experiments start and use the same interference graph for all runs. This way the quality of channel assignment is also subject to the temporal variation in the interference graph, which is more realistic.

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 [25]. 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 from NS-2 simulations. 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 first simulate synthetic topologies and show that the benefit of traffic-awareness is larger when the load is imbalanced. Then we compare different channel assignments using trace-driven simulations under accurate and inaccurate traffic demands.

A. Simulations on Synthetic Settings

As described in Chapter IV-A1, we create two types of demands to understand the benefit of traffic-aware assignment - uniform and hotspots. Figure 5 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.

As we can see, the improvement of traffic-awareness is mostly within 15% under uniform demands, whereas the improvement under hotspots traffic is significantly higher: in 20-35% cases, the improvement is over 20%, and in 10% cases, the improvement is over 50% in 1 hotspot and over 30% in 2 hotspots. The improvement in 1-hotspot case is higher than 2-hotspot case because 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 benefit is larger under hotspots than uniform demands because traffic-aware assignment aims to assign APs with high load to non-overlapping channels as much as possible; this significantly increases the overall throughput when the demands are highly skewed. Also, we observe the throughput (in absolute values) is highest when the channel assignment is both traffic-aware and client-aware.

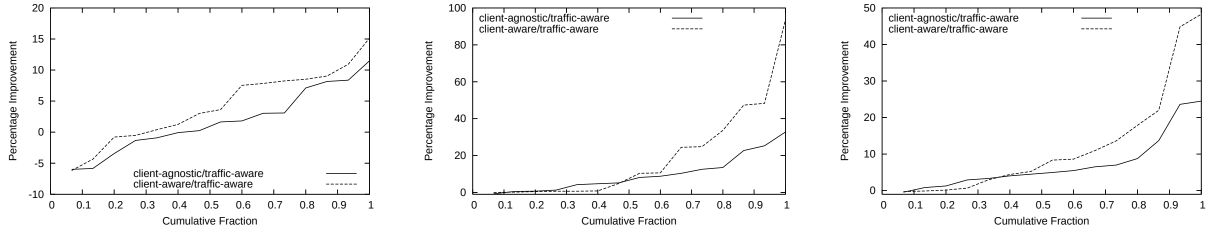
Note that in Figure 5(a) there are a small number of cases where we observe negative throughput improvement. This is because the current channel separation metric (even after incorporating traffic and client awareness) is not perfect. For example, consider a setting where two APs do not interfere with 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 their 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 subsection IV. First, we present a comparison of the performance improvement from traffic-awareness relative to traffic-agnostic assignment. Then, we investigate if traffic-aware assignment introduces unfairness into a WLAN by favoring transfers at heavily loaded APs. We also study the relation between the benefits of traffic awareness and the density of the WLAN network (in terms of the mean distance between wireless nodes) Finally, we present an evaluation of practical demand prediction algorithms discussed in subsection III-C3.

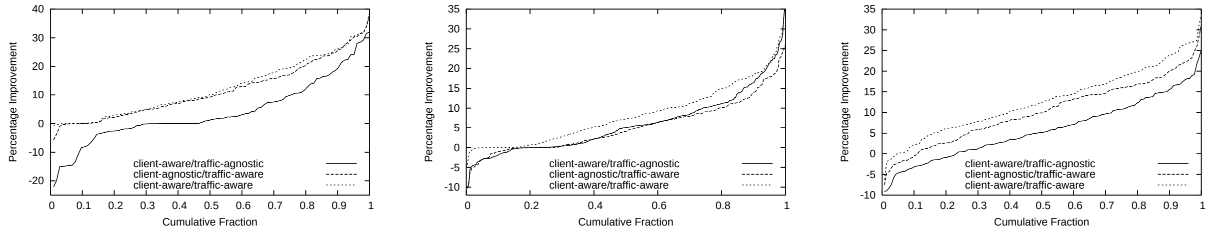
1) Performance Benefits of Traffic-awareness: First we compare four channel separation metrics assuming that we have perfect knowledge of traffic-demands. Figure 6 shows a CDF of performance improvement of various channel assignments against a traffic-agnostic/client-agnostic baseline. We note that the average throughput improvement is 4.0%-5.9% after incorporating client-side information alone; it raises to 5.2%-11.5% by incorporating traffic-demands alone; and further to 8.3-12.8% by incorporating both traffic-demands and client-side information.

As we can see in Figure 6, the client-aware/traffic-aware metric shows improvement over the client-agnostic/traffic-aware metric for LibBldg2 and MBldg. However, the improvement is less for ResBldg94. To see why the client-aware metric does not provide as much improvement in ResBldg94, we examine the interference patterns in the buildings. If a client interferes with the same set



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

Fig. 5. Comparison of traffic-aware schemes against their traffic-agnostic counterparts in synthetic topologies.



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

Fig. 6. Comparison of various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach as the baseline.

of APs as the AP it is associated with, then the benefit of client-aware channel assignment becomes smaller. However, if clients interfere with a different set of APs than that of the AP they are associated with, client interference becomes more important. We report the average number of clients that fall into the two former categories in the left and right columns of Table I, respectively. For ResBldg94, a relatively higher proportion of clients have the same interference pattern as their APs, and thus client-aware channel assignment has less impact. In comparison, LibBldg2 and MBldg see larger benefit of client-aware channel assignment since more clients interfere with different APs.

	Interfere with same APs	Interfere with different APs
ResBldg94	58.4	5.75
LibBldg2	14.9	6.56
MBldg	59.2	24.8

TABLE I
DETAILED BREAK-DOWN OF CLIENT-SIDE INTERFERENCE.

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 assignment. When traffic is more heterogeneous, the improvement is larger. For instance, we computed 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%, both in Figure 6(a). (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 that the more imbalanced the traffic demands, the larger benefit from using traffic-aware assignment.

Figure 7 compares the performance improvement of the two traffic-aware metrics against their traffic-agnostic counter-parts. The average improvement of traffic-aware/client-agnostic metric over traffic-agnostic/client-agnostic is 5.2-11.5%, whereas the average improvement of traffic-aware/client-aware over traffic-agnostic/client-aware is 2.4-8.6%. The former improvement is larger because the baseline performance is worse. For ResBldg (Figure 7(a)), the largest improvement of traffic-awareness is $> 35\%$ for either metric.

Approach	Fairness		
	ResBldg	LibBldg	MBldg
Traffic-agnostic/client-agnostic	0.89	0.87	0.85
Traffic-unaware/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

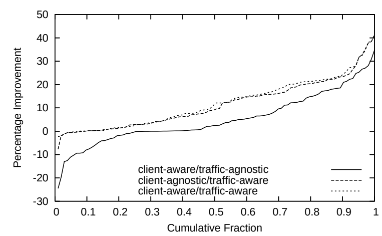


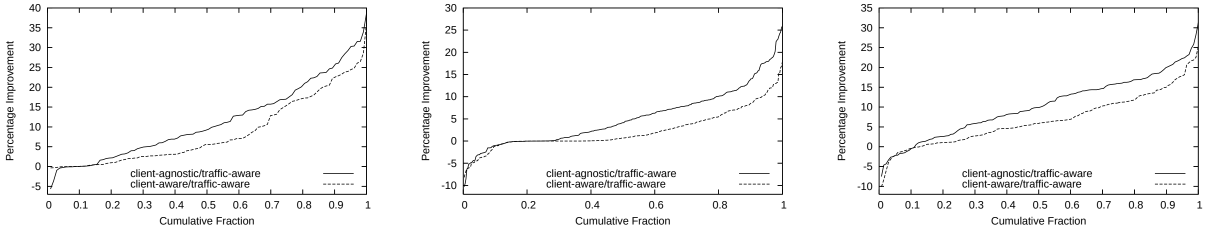
Fig. 8. Comparison of various traffic-aware schemes against their traffic-agnostic counterparts under Zipf-distributed traffic demands.

2) Fairness: Next we ask if traffic-awareness creates unfairness among APs. We consider the ratio of the actual throughput obtained 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 benefits from traffic-aware assignment do not come at the expense of the fairness in throughput allocation to individual flows.

3) Impact of Zipf-distributed Demands: In the above simulations, we assumed that the demand of an AP was equally distributed across its clients and its client demands were also equal. Now, we study the benefits under a more skewed client demand distribution.

Figure 8 compares various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach as the baseline when Zipf-distributed traffic demands are used for ResBldg. Compared with Figure 6(a), we observe the relative performance of various algorithms is similar.

Note that the total traffic rate to and from each AP is the same in both cases. The lack of any significant difference between uniform and Zipfian client demands indicates that the aggregate traffic volume in a BSS is a more important factor in traffic-aware assignment than the actual distribution of the traffic among clients



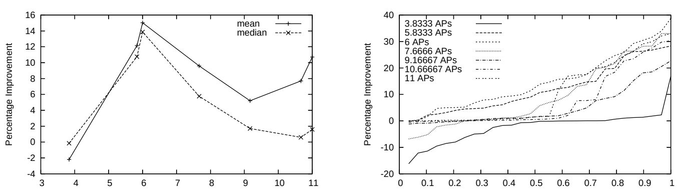
(a) ResBldg94 (b) LibBldg2 (c) MBldg

Fig. 7. Comparison of various traffic-aware schemes against their traffic-agnostic counterparts.

in the BSS.

4) Impact of Network Density: We now study the relationship between the density of a WLAN deployment and the benefits of traffic-awareness. More specifically, we want to understand if the benefits are higher in dense deployments.

Figure 9 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 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 channel assignments and the benefits of traffic-aware channel assignment are reduced.



(a) Average improvement (b) CDF of improvements

Fig. 9. 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.

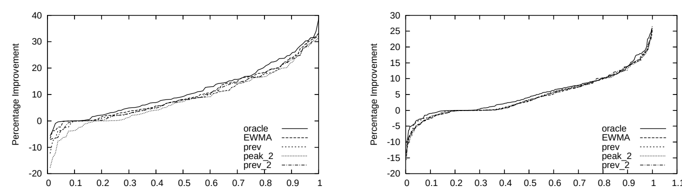
5) Evaluation of Practical Traffic-aware Algorithms: In the previous evaluation, we assume that traffic-aware channel assignments have perfect knowledge of traffic-demands. In practice, such information is not known a priori, but has to be estimated based on historical information. A natural question arises: can the prediction error offset the potential gain of traffic-aware channel assignment?

	EWMA	Previous	$Peak_2$	$Peak_4$
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

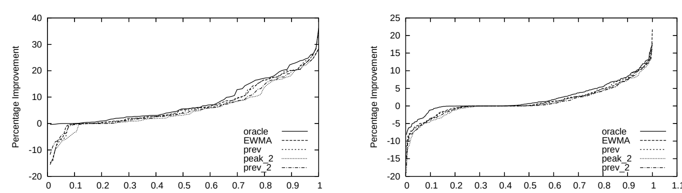
To answer this question, we first compute the error in predicting traffic demands using various prediction algorithms described in subsection III-C. We quantify the prediction error using mean absolute error (MAE), defined as $\frac{\sum_i |predict_i - actual_i|}{\sum_j actual_i}$. Table III shows the error involved 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.

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 seen by two traffic-aware metrics over their traffic-agnostic counterpart. Figure 10 compares the client-aware metric, while Figure 11 compares the client-unaware metric. The performance of the prediction algorithms closely tracks the the oracle. Compared with the oracle the degradation of predictive algorithms is within 6% (e.g. see the EWMA algorithm for client-unaware in ResBldg94). Compared with the traffic-agnostic algorithm, the improvement is still substantial. The performance degradation for client-unaware channel assignment is less than that for the client-aware channel assignment. For client-aware, we have to predict the client side demands too and this further increases the prediction error. The median improvement for client-unaware channel assignment is 8.13% while for client-aware channel assignment it is 5.26% (both values for ResBldg94).



(a) ResBldg94 (b) LibBldg2

Fig. 10. Comparison of client-unaware channel assignments using various prediction algorithms.

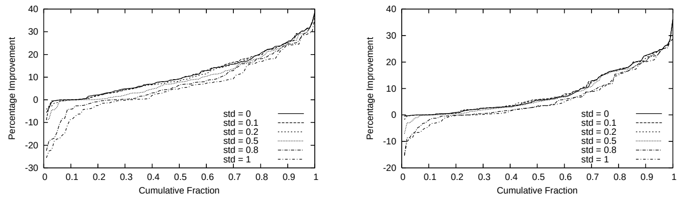


(a) ResBldg94 (b) LibBldg2

Fig. 11. Comparison of client-aware channel assignments using various prediction algorithms.

Our evaluations suggest that even though wireless traffic is hard to predict accurately, it is still possible 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 confirm this further, we conduct simulations where we introduce Gaussian errors into the traffic demands.

6) Impact of Incorrect Information: To evaluate the robustness of traffic-aware channel assignment, we “poison” the traffic demands with artificially generated error and use the poisoned



(a) Traffic-aware, client-agnostic (b) Traffic-aware, client-aware

Fig. 12. Comparison of traffic-aware schemes against their traffic-agnostic counterparts under Gaussian distributed errors with mean=0 and different standard deviation.

demands as input to channel assignment. Figure 12 shows the CDF of performance improvement of traffic-aware channel assignment schemes against their traffic-agnostic counterparts when we add errors with different standard deviation. 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 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.

VI. Implementation and Experiment Results

We implement the channel assignment algorithms as follows. We have a centralized controller that takes 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 APs' channels have been changed, the controller remotely starts the `nuttcp` [25] program with the specified traffic demands to measure network performance. For our evaluation, the controller collects the throughput reports from all APs again via the back-end Ethernet connection after each experiment.

A. Testbed 1

First, we use a smaller-scale testbed of 12-nodes to understand the potential benefits of traffic-aware channel assignment. Most of the 12 nodes interfere with each other. We generate possible but undesirable traffic-agnostic assignments by iteratively assigning the two access points with the highest remaining demands to the same channel. Such assignment is possible because traffic-agnostic assignment tries to balance the number of interfering APs on a channel without considering their traffic load. Table IV compares throughput under traffic-agnostic and traffic-aware channel assignments in eight different settings, derived from changing the following three factors: (i) TCP vs. UDP, (ii) with fixed rate and with multi-rate adaptation, and (iii) with and without RTS/CTS. When using a fixed rate, the data rate is set to 1 Mbps. In the throughput distribution column, APs assigned to channel 1 are in boldface, ones assigned to channel 6 are in italics, and ones assigned to channel 11 are in normal font.

We make the following observations. First, traffic-aware channel assignment consistently out-performs traffic-agnostic channel assignment. The largest improvement is 90.73%. The only exception arises in the first traffic demand under TCP, with multi-rate and RTS/CTS, where traffic-aware assignment yields 5.43% lower throughput. A closer examination reveals that the performance loss is caused by inaccurate assumption of interference patterns. We assume that all nodes interfere with each other, which is indeed the typical case in the testbed. However wireless interference relationships change over time. In this particular experiment, the fourth and fifth APs do not always interfere with each other, and achieve total throughput of 29.12 Mbps, which is significantly higher than 25.27 Mbps – the highest throughput that a single channel observed from our testbed.

Second, the throughput improvement has a strong correlation with "fairness index"; this is the Jain fairness index computed over the traffic demands. We calculate the Jain fairness index as described in Section V-B2. A lower index indicates more imbalance in traffic distribution, and results in larger benefit 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.

Third, we observe similar performance with and without RTS/CTS. We also note that the relative performance improvement from traffic-aware assignment under multi-rate adaptation is comparable to that without using multi-rate. This is because the clients and APs are close together, and their links generally operate at the highest data rate under both traffic-aware and traffic-agnostic assignments. When the client-AP link quality degrades, we expect a higher benefit of traffic-aware assignment, because traffic-aware assignment helps reduce interference and allow links to operate at higher data rates. We observe this phenomenon in our larger testbed, which we describe next.

B. Testbed 2

After evaluating the channel assignment in the subset of the testbed, we evaluate our channel assignment scheme on the entire testbed. As mentioned in Section IV-B, in our evaluation, we generate Zipf-distributed traffic demands with different slopes, where a larger slope indicates a higher imbalance in traffic pattern. The demands generated from these slope values are listed in Table V.

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

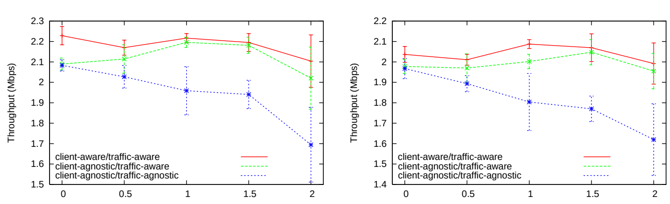
TABLE V
Normalized Zipfian demands in the testbed.

Since the effect of RTS/CTS is small as shown in Section VI-A, here we mainly focus on the other two factors, namely TCP/UDP and multi-rate adaptation.

Figure 13 and 14 show the overall network throughput over 5 runs under fixed-rate and multi-rate, respectively. We make the following observations. First, as we would expect, client-aware, traffic-aware performs the best, and client-agnostic, traffic-aware out-performs client-agnostic, traffic-agnostic. Second, the throughput variance of 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

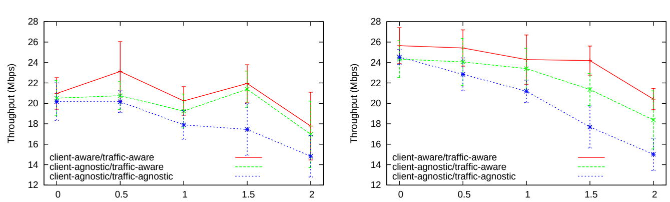
Normalized traffic demands (AP1, AP2, AP3, AP4, AP5, AP6)	Throughput for traffic-aware assignment (Mbps)		Throughput for traffic-unaware assignment (Mbps)		Improv-ement	Fair-ness
	Distribution	Total	Distribution	Total		
UDP Results - Fixed Rate						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(0.80, *0.28*, *0.28*, **0.43**, **0.42**, *0.28*)	2.49	(**0.66**, 0.28, *0.28*, *0.42*, **0.43**, 0.28)	2.35	5.98%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**0.51**, *0.17*, *0.67*, **0.51**, 0.17, 0.72)	2.74	(0.51, *0.17*, **0.46**, 0.51, *0.17*, **0.41**)	2.22	23.35%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **0.47**, 0.80, *0.81*, **0.39**)	2.47	(**0.00**, **0.00**, *0.45*, 0.40, 0.45, *0.42*)	1.73	42.88%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(0.17, *0.00*, *0.42*, 0.17, 0.17, **0.80**)	1.73	(**0.17**, **0.00**, 0.42, *0.17*, *0.17*, 0.45)	1.38	24.95%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 0.82, 0.00, *0.80*, **0.82**)	2.45	(**0.00**, *0.00*, 0.49, **0.00**, *0.80*, 0.38)	1.67	46.55%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *0.81*, 0.00, *0.00*, **0.83**)	1.64	(**0.00**, *0.00*, 0.47, **0.00**, *0.00*, 0.40)	0.86	89.90%	0.33
TCP Results - Fixed Rate						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(0.74, *0.28*, *0.18*, **0.39**, **0.41**, *0.28*)	2.30	(**0.58**, 0.28, *0.28*, *0.42*, **0.42**, 0.28)	2.27	1.04%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**0.46**, *0.17*, *0.60*, **0.50**, 0.17, 0.69)	2.59	(0.48, *0.17*, **0.43**, 0.49, *0.17*, **0.39**)	2.13	21.99%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **0.39**, 0.76, *0.77*, **0.41**)	2.34	(**0.00**, **0.00**, *0.42*, 0.40, 0.38, *0.40*)	1.61	45.66%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(0.17, *0.00*, *0.42*, 0.17, 0.17, **0.78**)	1.71	(**0.17**, **0.00**, 0.39, *0.17*, *0.17*, 0.43)	1.33	28.98%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 0.77, 0.00, *0.76*, **0.76**)	2.29	(**0.00**, *0.00*, 0.42, **0.00**, *0.77*, 0.39)	1.59	44.03%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *0.77*, 0.00, *0.00*, **0.76**)	1.54	(**0.00**, *0.00*, 0.42, **0.00**, *0.00*, 0.39)	0.81	89.32%	0.33
UDP Results - Multi-rate						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(24.66, *9.17*, *8.75*, **13.99**, **13.98**, *8.63*)	79.18	(**19.30**, 9.24, *9.24*, *14.00*, **14.00**, 9.24)	75.02	5.55%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**16.80**, *5.60*, *20.19*, **16.47**, 5.60, 25.17)	89.84	(16.80, *5.60*, **14.98**, 14.01, *5.60*, **12.79**)	69.78	28.74%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **15.40**, 22.65, *24.25*, **12.94**)	75.23	(**0.00**, **0.00**, *19.27*, 11.43, 15.16, *7.87*)	53.72	40.03%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(5.60, *0.00*, *14.00*, 5.60, 5.60, **26.08**)	56.88	(**5.60**, **0.00**, 14.00, *5.60*, *5.60*, 13.86)	44.66	27.37%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 26.80, 0.00, *24.45*, **26.02**)	77.27	(**0.00**, *0.00*, 16.79, **0.00**, *23.64*, 11.64)	52.07	48.38%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *26.06*, 0.00, *0.00*, **26.80**)	52.86	(**0.00**, *0.00*, 16.66, **0.00**, *0.00*, 11.98)	28.64	84.53%	0.33
TCP Results - Multi-rate						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(24.47, *8.38*, *8.85*, **12.34**, **13.52**, *7.69*)	75.25	(**18.74**, 6.16, *9.24*, *14.00*, **14.00**, 9.24)	71.38	5.42%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**16.80**, *5.60*, *18.24*, **16.80**, 5.60, 24.67)	87.71	(16.80, *5.60*, **13.22**, 15.40, *5.60*, **12.82**)	69.44	26.31%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **13.38**, 19.92, *23.73*, **12.56**)	69.60	(**0.00**, **0.00**, *13.09*, 10.47, 14.48, *13.20*)	51.24	35.82%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(5.60, *0.00*, *14.00*, 5.60, 5.60, **25.50**)	56.30	(**5.60**, **0.00**, 13.99, *5.60*, *5.60*, 12.32)	43.11	30.58%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 25.12, 0.00, *24.20*, **25.18**)	74.49	(**0.00**, *0.00*, 14.81, **0.00**, *23.64*, 12.28)	50.73	46.83%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *24.35*, 0.00, *0.00*, **25.64**)	49.99	(**0.00**, *0.00*, 15.33, **0.00**, *0.00*, 11.14)	26.46	88.90%	0.33
UDP Results - Fixed Rate - RTS/CTS						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(0.75, *0.28*, *0.26*, **0.40**, **0.42**, *0.26*)	2.37	(**0.41**, 0.28, *0.28*, *0.42*, **0.42**, 0.28)	2.10	13.06%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**0.43**, *0.17*, *0.61*, **0.39**, 0.17, 0.68)	2.44	(0.36, *0.17*, **0.42**, 0.42, *0.17*, **0.35**)	1.90	28.90%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **0.42**, 0.75, *0.76*, **0.34**)	2.28	(**0.00**, **0.00**, *0.43*, 0.43, 0.36, *0.38*)	1.60	42.73%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(0.17, *0.00*, *0.42*, 0.17, 0.17, **0.73**)	1.67	(**0.17**, **0.00**, 0.38, *0.17*, *0.17*, 0.42)	1.31	27.47%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 0.76, 0.00, *0.77*, **0.74**)	2.27	(**0.00**, *0.00*, 0.39, **0.00**, *0.77*, 0.42)	1.57	44.35%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *0.77*, 0.00, *0.00*, **0.77**)	1.53	(**0.00**, *0.00*, 0.40, **0.00**, *0.00*, 0.41)	0.81	89.76%	0.33
TCP Results - Fixed Rate - RTS/CTS						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(0.73, *0.27*, *0.27*, **0.36**, **0.42**, *0.22*)	2.27	(**0.44**, 0.28, *0.28*, *0.42*, **0.35**, 0.28)	2.05	10.37%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**0.49**, *0.17*, *0.56*, **0.29**, 0.17, 0.63)	2.32	(0.37, *0.11*, **0.38**, 0.39, *0.17*, **0.38**)	1.81	28.34%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **0.40**, 0.73, *0.73*, **0.36**)	2.22	(**0.00**, **0.00**, *0.46*, 0.38, 0.37, *0.32*)	1.53	45.52%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(0.17, *0.00*, *0.42*, 0.17, 0.17, **0.73**)	1.66	(**0.17**, **0.00**, 0.38, *0.17*, *0.17*, 0.39)	1.28	30.20%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 0.74, 0.00, *0.74*, **0.72**)	2.20	(**0.00**, *0.00*, 0.40, **0.00**, *0.73*, 0.38)	1.51	45.72%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *0.74*, 0.00, *0.00*, **0.73**)	1.47	(**0.00**, *0.00*, 0.35, **0.00**, *0.00*, 0.42)	0.77	90.73%	0.33
UDP Results - Multi-rate - RTS/CTS						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(25.27, *6.86*, *9.01*, **11.84**, **13.77**, *7.90*)	74.66	(**18.23**, 9.24, *9.24*, *14.00*, **13.86**, 9.24)	73.81	1.14%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**16.80**, *5.60*, *15.41*, **13.45**, 5.60, 24.61)	81.47	(16.80, *5.60*, **14.56**, 11.53, *5.60*, **10.65**)	64.74	25.83%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **11.72**, 17.27, *24.37*, **13.17**)	66.54	(**0.00**, **0.00**, *8.41*, 6.72, 18.67, *15.10*)	48.90	36.07%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(5.60, *0.00*, *13.84*, 3.72, 5.60, **24.59**)	53.35	(**5.60**, **0.00**, 12.08, *3.73*, *5.60*, 13.53)	40.54	31.60%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 24.21, 0.00, *24.95*, **24.87**)	74.03	(**0.00**, *0.00*, 13.45, **0.00**, *24.32*, 12.11)	49.87	48.44%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *18.90*, 0.00, *0.00*, **24.88**)	43.78	(**0.00**, *0.00*, 11.94, **0.00**, *0.00*, 13.46)	25.40	72.32%	0.33
TCP Results - Multi-rate - RTS/CTS						
(1.00, 0.33, 0.33, 0.50, 0.50, 0.33)	(22.34, *5.85*, *8.18*, **9.27**, **14.00**, *7.38*)	67.00	(**15.13**, 9.24, *9.24*, *14.00*, **13.99**, 9.24)	70.84	-5.43%	0.82
(0.60, 0.20, 0.90, 0.60, 0.20, 0.90)	(**14.58**, *5.59*, *12.29*, **16.80**, 5.60, 23.37)	78.23	(16.80, *5.60*, **11.03**, 11.26, *5.60*, **11.68**)	61.96	26.26%	0.80
(0.00, 0.00, 1.00, 1.00, 1.00, 1.00)	(*0.00*, *0.00*, **12.39**, 13.16, *23.30*, **10.42**)	59.26	(**0.00**, **0.00**, *9.55*, 4.89, 17.82, *12.48*)	44.74	32.45%	0.67
(0.20, 0.00, 0.50, 0.20, 0.20, 1.00)	(5.60, *0.00*, *13.17*, 5.60, 5.60, **23.18**)	53.15	(**5.60**, **0.00**, 8.76, *5.60*, *5.60*, 14.46)	40.02	32.80%	0.54
(0.00, 0.00, 1.00, 0.00, 1.00, 1.00)	(**0.00**, **0.00**, 21.62, 0.00, *22.95*, **22.81**)	67.37	(**0.00**, *0.00*, 10.64, **0.00**, *22.41*, 13.01)	46.06	46.27%	0.50
(0.00, 0.00, 1.00, 0.00, 0.00, 1.00)	(0.00, 0.00, *14.96*, 0.00, *0.00*, **23.06**)	38.01	(**0.00**, *0.00*, 10.48, **0.00**, *0.00*, 12.97)	23.45	62.11%	0.33

TABLE IV
SUMMARY OF 12-NODE TESTBED EXPERIMENT RESULTS.



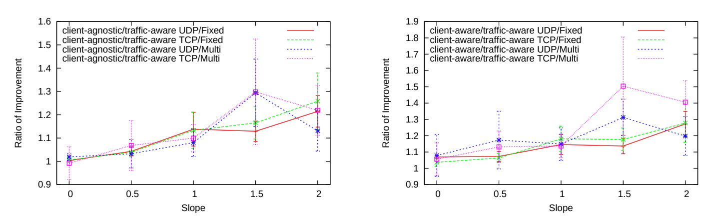
(a) UDP (b) TCP

Fig. 13. Overall network throughput in 25-node testbed under fixed MAC data rate, where the errorbars show the average and standard deviation.



(a) UDP (b) TCP

Fig. 14. Overall network throughput in 25-node testbed under multi-rate, where the errorbars show the average and standard deviation.

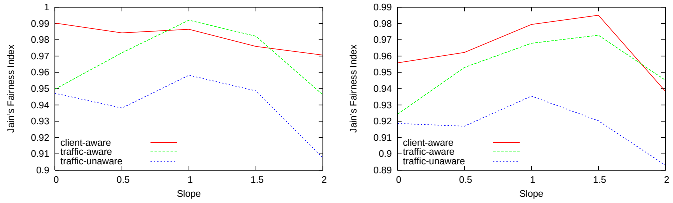


(a) Client-agnostic, traffic-aware (b) Client-aware, traffic-aware

Fig. 15. Average and standard deviation of throughput improvement over client-agnostic, traffic-agnostic baseline.

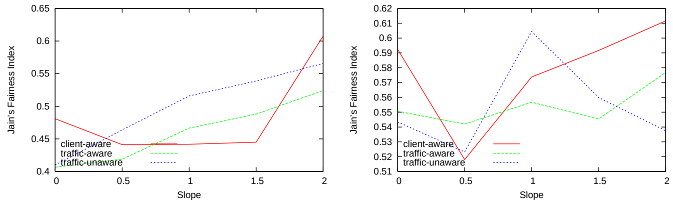
assignments may appear equally good according to the traffic-agnostic metric, but its actual performance varies significantly depending on whether the nodes with high demands happen to be assigned to non-interfering channels.

Figure 15 further shows the improvement of traffic-aware, client-agnostic and traffic-aware, client-aware over the traffic-agnostic, client-agnostic baseline. As it shows, the improvement of traffic-awareness generally increases with the slope α. When



(a) UDP (b) TCP

Fig. 16. Jain's Fairness Index in 25-node testbed under fixed MAC data rate.



(a) UDP (b) TCP

Fig. 17. Jain's fairness index in 25-node testbed under multi-rate.

$\alpha = 0$ (i.e., all traffic demands are the same), the client-agnostic, traffic-aware metric perform similarly as the client-agnostic, traffic-agnostic metric. The client-aware, traffic-aware slightly out-performs both the above metrics by accounting client-side interference. As α increases, traffic becomes more concentrated on a smaller number of nodes and both traffic-aware metrics see larger improvement. Furthermore, we observe the variance of throughput improvement can be quite high. This is due to the performance fluctuation in the traffic-agnostic metric, as explained earlier. The improvement of traffic-awareness in some cases can be quite high: we observe up to a 1.52-fold increase for TCP/fixed-rate, and a 1.8-fold increase for TCP/multi-rate (see slope 1.5 in Figure 15). 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.

We now study the impact of Jain's fairness index on TCP and UDP traffic for fixed-rate and multi-rate. Figure 16 and Figure 17 shows the fairness index for the different metrics under fixed rate and multi-rate respectively. It represents the values averaged over the five runs. We make following observations.

(1) For all the metrics, the Fairness index value is better under fixed MAC data rate as compared with the multi-rate.

(2) Under both the fixed rate and multi-rate scenarios, for either UDP or TCP traffic, the fairness index of the metrics is comparable. This suggests that traffic aware channel assignment does not degrade the fairness in demand allocation at different APs.

VII. Discussion

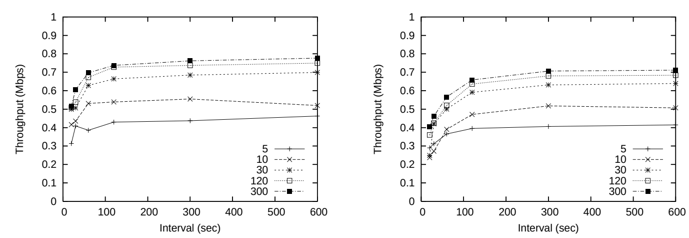
The evaluation of our algorithms show that we can achieve effective channel assignments by taking traffic demands into consideration. However, some practical considerations, such as the impact of channel switching, have not been analyzed. In this section we investigate these issues.

A. 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 [23], the switching delay varies from 200 us on Intel's ProWireless to 10-20 ms on NetGear Atheros, Cisco Aironet, and Prism 2.5.

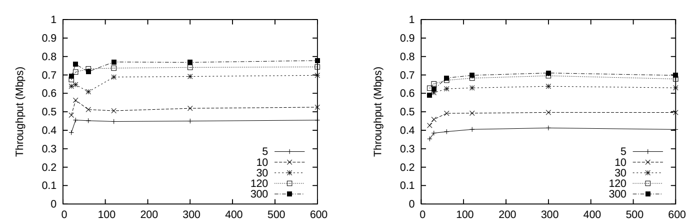
The re-association delay depends on the re-association scheme. A simple approach, which is implemented by MadWiFi, is for wireless clients to scan all channels to find the AP with the highest RSSI. The re-association delay in this case tends to be long and is dominated by the scanning time. To reduce 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 [26]. 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 18 and Figure 19 summarize the results of a 10 minute transfer between an AP and its client using both TCP and UDP. The x-axis tracks how often the AP changes its channel. To evaluate the impact of frequent channel switching on different transfer duration, we use on-off traffic, where both on-periods and off-periods are exponentially distributed. Different lines in the graph correspond to different average on-period, ranging from 5 to 300 sec. The average off-period duration is 5 seconds. The process is repeated until 10 minutes elapse. As shown in Figure 18, there is no degradation under the default re-association scheme when the switching interval is 2 minutes or higher. For a smaller switching interval, the overhead of the default scheme increases. In comparison, as we can see from Figure 19, under the explicit notification scheme, the overhead is negligible for all switching intervals, including switching once per 20 seconds. These results suggest the re-association overhead is negligible under the explicit notification; even for default implementation, switching once 5 minutes, as considered in this evaluation, incurs no performance penalty for both TCP and UDP traffic.



(a) UDP traffic (b) TCP traffic

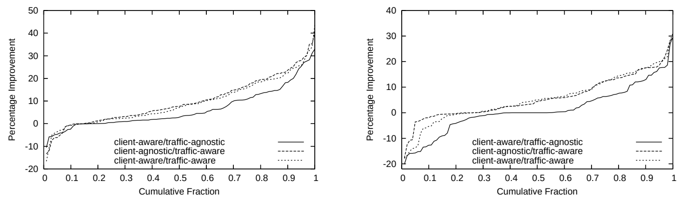
Fig. 18. Channel switching overhead under MadWiFi default implementation.



(a) UDP traffic (b) TCP traffic

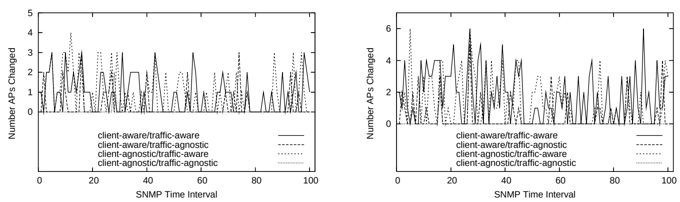
Fig. 19. Channel switching overhead under explicit notifications.

An orthogonal approach to further minimize the impact of channel switching is to reduce the number of APs that change the channels. To address this issue, we use the channel assignment from the previous measurement interval as the starting point in the simulated annealing algorithm. We can further limit the number of channel switches by controlling the number of iterations in simulated annealing. This will bias the outcome of the search in favor of assignments that are only slightly different from the current channel assignment. We evaluate the effectiveness of this approach using our simulations of Dartmouth's ResBldg.



(a) 5 iterations in SA (b) 10 iterations in SA

Fig. 20. Comparison of various channel assignment schemes against a traffic-agnostic, client-agnostic channel assignment approach as the baseline.



(a) 5 iterations in SA (b) 10 iterations in SA

Fig. 21. A time series plot of the number of APs that change their channel.

Figure 20 plots the CDF of performance improvement against the traffic-agnostic, client-agnostic assignment under the enhanced approach. We limit the number of simulated annealing iterations to 5 and 10. Compared with Figure 6(a), we observe the performance improvement is similar. Figure 21 shows the number of APs that change their channel over time. Traffic-aware, client-aware assignment results in at most 3 APs changing channel (per interval) under 5 iterations in SA, and at most 5 APs changing channels under 10 iterations in SA. The average numbers of APs that change channels are even lower: 0.93 APs under 5 iterations and 1.55 APs under 10 iterations. Therefore, the enhancement is effective in reducing the number of channel changes without compromising the performance. Another complementary approach is to apply the new channel assignment to the real network only if it improves the optimization metric by a threshold θ. This is not a part of our current implementation and we hope to incorporate it in our future work.

B. 802.11a

Our analysis has focused on 802.11b and g networks, which support fewer operating frequencies than 802.11a. It might appear that traffic-awareness 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.

VIII. Summary of Results and Concluding Remarks

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 the 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, both fixed rate and multi-rate adaptation.

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] IEEE 802.11k radio resource measurement. http://standards.ieee.org/802news/.

[2] S. Agarwal, A. Nucci, and S. Bhattacharyya. Measuring the shared fate of IGP engineering and interdomain traffic. In *Proceedings of the 13th International Conference on Network Protocols (ICNP) '05*, 2005.

[3] 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)*, 2005.

[4] N. Ahmed and S. Keshav. Smarta: A self-managing architecture for thin access points. In *Proc. of CoNext*, Dec. 2006.

[5] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self management in chaotic wireless deployments. In *Proc. of ACM MOBICOM*, Sept. 2005.

[6] Alcatel AirView Software. http://www.alcatel.com.

[7] Autocell. http://www.propagatenetworks.com/product/.

[8] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. *Overview and Principles of Internet Traffic Engineering, RFC 3272*, May 2002.

[9] D. O. Awduche. MPLS and traffic engineering in IP networks. *IEEE Communication Magazine*, pages 42–47, Dec. 1999.

[10] 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.

[11] 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/.

[12] G. J. Chaitin. Register allocation & spilling via graph coloring. In *Proc. of SIGPLAN*, 1982.

[13] Dartmouth campus-wide wireless traces. http://www.cs.dartmouth.edu/ campus/.

[14] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *Proceedings of IEEE INFOCOM '01*, Anchorage, AK, April 2001.

[15] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. of ACM MOBICOM*, Sept. 2004.

[16] Wireless traces at IBM corporation. http://nms.lcs.mit.edu/ mbalazin/wireless/.

[17] Jim Grier. Assigning 802.11b Access Point Channels. http://www.wi-fiplanet.com/tutorials/article.php/972261.

[18] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proceedings of ACM SIGCOMM '05*, Philadelphia, PA, August 2005.

[19] I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunications: A comprehensive survey. *IEEE Personal Communications*, pages 10–31, 1996.

[20] 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.

[21] X. Meng, S. Wong, Y. Yuan, and S. Lu. Characterizing flows in large wireless data networks. In *Proc. of ACM MOBICOM*, Sept. 2004.

[22] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *INFOCOM*, Barcelona, Spain, APR 2006.

[23] A. Mishra, V. SHrivastava, D. Agarwal, S. Banerjee, and S. Ganguly. Distributed Channel Management in Uncoordinated Wireless Environments. In *Proc. of MobiCom*, 2006.

[24] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially-overlapped Channels not considered harmful. In *Proc. of ACM SIGMETRICS*, June 2006.

[25] nuttcp. http://sd.wareonearth.com/~phil/net/ttcp/.

[26] I. Ramani and S. Savage. Syncscan: Practical fast hand-off for 802.11 infrastructure networks. In *Proc. of IEEE Infocom*, Mar. 2005.
[27] A. Raniwala and T. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE Infocom*, Mar. 2005.
[28] 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)*, Apr. 2004.
[29] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2nd edition, Dec. 2001.
[30] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference. In *Proc. of ACM SIGCOMM*, 2006.
[31] Simulated annealing. http://www.cs.sandia.gov/opt/survey/sa.html.
[32] X. Xiao, A. Hannan, B. Bailey, and L. Ni. Traffic engineering with MPLS in the Internet. *IEEE Network*, pages 28–33, Mar. 2000.
[33] C. Zhang, Z. Ge, J. Kurose, Y. Liu, and D. Towsley. Optimal routing with multiple traffic matrices: Tradeoff between average case and worst case performance. In *Proceedings of the 13th International Conference on Network Protocols (ICNP) '05*, Boston, MA, November 2005.
[34] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley. On optimal routing with multiple traffic matrices. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, April 2005.