Dynamic Channel Assignment in Wireless LANs

Bo Wang[1], William Wu[2], Yongqiang Liu[3]

[1]*Institute of Computing Technology, Chinese Academy of Sciences*
[2]*Electrical Engineering Dept. Stanford University*
[3]*NEC Laboratories China*
bwang@jdl.ac.cn, willywu@stanford.edu, liuyongqiang@research.nec.com.cn

Abstract

A traffic-aware and client-aware algorithm for dynamically assigning channels to Access Points (APs) in Wireless Local Access Networks (WLANs) is proposed. Traffic loads and Received Signal Strength (RSS) values are used to characterize interference. The problem of selecting channels to minimize total interference then reduces to the Max k-Cut problem, which we efficiently solve with semidefinite programming (SDP) relaxation. Testbed experiments demonstrate that our traffic-aware algorithm can significantly improve the quality of a channel assignment in terms of total network throughput and channel utilization. Under a randomly generated continuous traffic distribution, throughput gains of 40% on average over static channel assignment are observed.

1. Introduction

The explosive popularity of Wireless Local Area Networks (WLANs) in recent years has led to a dramatic rise in the density of WiFi Access Points (APs) in wireless environments. The wireless spectrum is divided into "channels" – bands over which APs can operate. A natural problem that arises is how to assign channels to the APs so as to increase the total network throughput. A well-established heuristic approach is to minimize the interference between APs using the same channel. Due to the broadcast nature of wireless, as the AP density of WLANs continues to increase, this problem of interference is becoming more severe and important.

Currently, network administrators often manually decide on a static channel assignment for APs based on RF profiles [1]. However, traffic loads in a network tend to vary with time, and consequently, such assignments do not result in the best performance.

In this work, we propose a centralized dynamic channel assignment algorithm that efficiently adapts the channel assignments alongside variations in traffic. Our approach incorporates the observed traffic demands of wireless APs and clients, as well as their power-inferred locations, in order to increase network throughput as a whole. The key points and advantages of our algorithm may be summarized as follows:

(1) We adopt the heuristic of minimizing sum network interference to increase overall network throughput. Experiments in our paper confirm the validity of this.

(2) Our algorithm is "traffic-aware", constantly monitoring the traffic patterns of both APs and stations. This allows us more efficient channel usage by designing channel assignments that accommodate to the traffic distribution.

(3) Our algorithm is "client-aware". This means that in addition to APs, clients also take part in measuring and reporting received signal power and traffic, thereby giving us a better picture of interference in the network.

(4) We use actual measurements of signal power and traffic load to intuitively define and measure interference. This measurement-based approach allows us to avoid making simplistic modeling assumptions for the wireless channel and MAC protocol, as such theoretical models are known to be especially inaccurate indoors. It also allows us to assess interference at a higher resolution than otherwise.

(5) Characterizing interference in terms of signal power has the desirable property of *additivity*, which will more easily allow us to address simultaneous interferers. Namely, additivity allows us to reduce the minimization problem to Max k-Cut, which can then be relaxed into a semidefinite programming problem that can be efficiently solved.

Due to all of these design characteristics, the techniques proposed in this paper improve channel assignment for WLANs, exploiting opportunities both to make use of idle channels, and also to reuse active channels. Our experiments show that our channel assignment algorithm substantially improves network throughput by 40% on average when compared to a static assignment scheme.

Our paper is structured as follows: In Section 2, we give some background on WLANs. Section 3 presents an analysis and model for interference. Section 4 describes the reduction to Max k-Cut, and the SDP relaxation used to approximately

978-0-7695-3342-1/08 $25.00 © 2008 IEEE
DOI 10.1109/PEITS.2008.105

IEEE computer society

solve it. Section 5 gives a high-level summary of our dynamic channel assignment algorithm. Implementation issues are discussed in Section 6, and experiments are presented in Section 7. Related works are discussed in Section 8. Lastly, we conclude in Section 9.

2. Background on WLANs

A WLAN consists of two kinds of participants: *APs* and *stations* (users). APs are like the service providers of Internet access for stations. When we wish to refer to either APs or users, without distinction, we may call them *nodes*. Each station *associates* itself with exactly one AP to receive service. We assume that stations are always associated with the AP from which received signal power is strongest. As the station surfs the Internet, data is then sent back and forth between the station and its AP. Stations associated with an AP are called that AP's *clients*. There are no direct client-to-client data flows, nor any direct AP-to-AP data flows.

We define a *cell* as an AP together with all stations that are associated with it. Each node in a cell must use the same *channel*, which is a frequency band over which communications take place. The 802.11 signal frequency range is divided into a number of these channels. For instance, 802.11b/g operates in the 2.4GHz unlicensed frequency band, and provides a set of 11 channels, among which there are only 3 non-overlapping channels. 802.11a operates in the 5.4GHz unlicensed frequency band, and provides 12 non-overlapping channels. In this paper, we always assume that channels are non-overlapping; however, our work can be extended to the case of partially overlapping channels easily [2].

Lastly, we address carrier-sensing. When a node wishes to transmit, it first measures the signal strength it is currently sensing on its channel. If this amount exceeds a certain Clear Channel Assessment (CCA) threshold (e.g., -85 dBm for NIC cards in our testbed), then to prevent signal collision, the CSMA (Carrier-Sense Multiple Access) protocol does not allow transmission to be initiated.

3. Interference Model

3.1. Motivations for Minimizing Interference

We first suggest some general rationale for wanting to minimize interference. Ultimately, we hope to increase throughput, which we assume has a relationship proportional with SINR, the signal-to-interference-and-noise ratio. This relationship can be seen in Table 1 [3], which describes the 802.11 a/g specification of SINR requirements for different transmission rates on wireless links.

Table 1 SINR vs. Data rate In 802.11 a/g

SINR (dB)	6	7.8	9	10.8	17	18.8	24	24.6
Data rate (Mbps)	6	9	12	18	24	36	48	54

We express SINR as

$$SINR = \frac{Intended\ \ Signal}{Background\ \ Noise + Interference\ \ Signal}.$$

The "Background Noise" is usually constant. "Intended Signal" refers to the amount of signal the receiver senses from the node that intended to communicate with it. Lastly, "Interference Signal" refers to the superposition of signals received from other nodes that are broadcasting on the same channel. In an environment with few mobile users (e.g., an office), the locations of senders and receivers are roughly fixed over some time scale. Thus we may consider the "Intended Signal" to also be fixed. Hence, only the "Interference Signal" may change with channel assignment, so we focus on reducing this quantity in order to increase SINR.

3.2. Motivation for Traffic-Awareness and Client-Awareness

We now look at a small example showing how client information can help us find opportunities for channel reuse. Consider Figure 1. There are 2 APs (AP1, AP2) and 4 stations (STA 1-4). STA1 and STA2 connect with AP1, STA3 and STA4 connect with AP2. The circles depict the regions that each AP covers. There is an "interference region" in which these circles overlap; to fully cover the network, such regions are always present. Assume all transmissions are on downlink: that is, APs send to stations. Now note that STA2 and STA3 are in the interference region. That means when STA2 is receiving from AP1, it will be subject to interference from AP2. Similarly, STA3 suffers interference from AP1 when receiving from AP2. On the other hand, pairs AP1-STA1 and AP2-STA4 are free from interference. Thus, we can deduce which nodes lie in the interference region only by comparing the relative interference powers received at the stations.

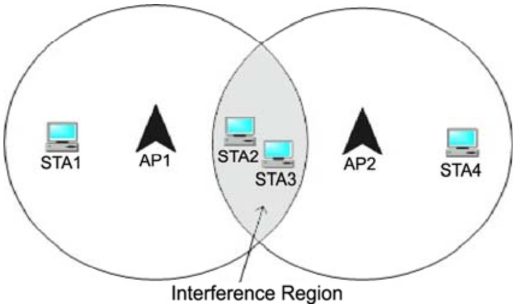

Figure 1 Interference Example.

In this scenario, the decision of how to assign channels to APs depends on the distribution of client traffic. When STA2 and STA3 are quiet with no throughput, and STA1 and STA4 are downloading, we could allocate the same channel to these two APs without interference. This is the notion of *channel reuse*. On the other hand, when STA1 and STA4 are quiet, but STA2 and STA3 are downloading, we must assign different channels to the APs to avoid interference. Thus, the interference in the network is a function of both the locations of nodes and the traffic distribution. The example also shows how it is useful to be aware of client traffic information in addition to AP traffic information. If we relied only on capturing congestion information at the AP, it would have been very difficult to distinguish between the two aforementioned scenarios in an algorithmic manner.

3.3. A Model of Interference

In the absence of interference, the link capacity would be the product of the maximum sending rate of the sender and the delivery ratio of the link. However, interference may reduce the maximum feasible sending rate of the sender. It may also reduce the probability of the receiver successfully receiving a packet, due to collisions. We now seek a model of interference in terms of measurable observables that accounts for both the sender's and receiver's perspective.

Before describing the model, we informally define the term "load". The "sending load" of a node is the fraction of time it's transmissions acquire the medium when it wishes to transmit to various recipients (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 load" is the fraction of time its receptions acquire the medium when the node wishes to receive from various transmitters. Intuitively, the load value reflects the *probability* of the transmission or reception of a node at any time point. In Section V, we discuss how load information can be collected in practice.

We first consider the effects of traffic distribution on interference. Given two nodes *A* and *B*, let $Interference_A^B$ denote the interference *A* experiences due to *B* over some predetermined time interval. Since interference only occurs when there are ongoing transmissions at *B*, $Interference_A^B$ must be proportional to the *sending load* of *B*, denoted by $Lsend_B$. Furthermore, *B* affects *A* only when *A* is *active* – that is, when *A* is either transmitting or receiving. If *A* desires to transmit, CSMA may force it to back off from acquiring the channel due to the transmissions of *B*. And if *A* is receiving while *B* is transmitting, then naturally the SINR is affected. Thus, if we denote the *receiving load* of *A* by $Lrecv_A$, then $Interference_A^B$ must also be proportional to ($Lsend_A$ + $Lrecv_A$), the fraction of time over which *A* is active.

We now consider the effect of received signal power. The shorter the distance between *A* and *B*, and the fewer the obstacles between them, the stronger the strength of the signal received by *A* from *B*, and thus the larger is $Interference_A^B$. This attenuated signal power can be quantified by RSS_A^B (Received Signal Strength), a quantity reported by the NIC.

Ignoring multipath fading effects on the interference, the above considerations naturally lead us to the following proposed interference metric:

$$Interference_A^B = RSS_A^B \bullet Lsend_B \bullet (Lsend_A + Lrecv_A) . \quad (1)$$

We now define the interference *between A* and *B* as

$$I(A,B) = Interference_A^B + Interference_B^A .$$

This is intuitive because there are two directions of interference between two nodes, and we wish to account for both. Due to the broadcast nature of wireless, it is reasonable to assume that the power of interfering signals combines additively. (Note it is crucial here that we use units of mW and not dBm.) We can thus naturally extend our definition of interference between non-singleton sets of nodes as

$$I(\{A_1,\ldots,A_m\},\{B_1,\ldots,B_n\}) = \sum_{1\le j\le m, 1<k\le n} I(A_j,B_k) . \quad (2)$$

4. Relation With Max k-Cut

4.1. Reduction to MkC

In this part, we formally state the interference minimization problem and show how it reduces to MkC. Let cells be denoted by $C_1, C_2, \ldots, C_N$, where C_i contains AP_i and all stations associated with AP_i, and *N* is the number of APs. For a high node density scenario, due to CSMA, there can be only one node transmitting in the same cell at any given time. Also, all available channels are orthogonal. Thus, *interference can only occur between different cells using the same channel.* The total network interference can then be written as

$$I_{Total} := \sum_{1\le i<j\le N} I(C_i,C_j)Ch(i,j) ,$$

where $I(C_i,C_j)$ is the interference that would be produced if cells *i* and *j* use the same channel, and $Ch(i,j)=1$ if *i* and *j* use the same channel, or 0 otherwise. Suppose that there are *k* channels available. The problem is then to minimize I_{Total} over the possible assignments of cells to channels $[1:k]$.

We may think of this as a graph-theoretic problem as follows: First construct a complete, undirected graph $G=(V,E)$, where the vertex set $V=\{C_1, C_2, \ldots, C_N\}$, and the edge set $E=\{E_{ij}\}$ for $i,j \in [1:N]$ and $i \ne j$. To each edge, we assign the weight $W_{ij} := I(C_i,C_j)$. Then choosing an assignment to minimize I_{Total} is the Min k-Partition (MkP) problem, which is NP-hard. Since the sum of all edge weights in the graph is constant, we can instead formulate the equivalent problem of maximizing the sum weight of edges that go *across* partitions. That is,

$$\text{Maximize} \sum_{1\le i<j\le N} W_{ij} D(i,j) ,$$

where $D(i,j) = 1 - Ch(i,j)$. This is the Max k-Cut (MkC) problem.

4.2. SDP Relaxation for MkC

We perform an SDP relaxation of MkC according to the technique suggested in [4]. We will use the following fact

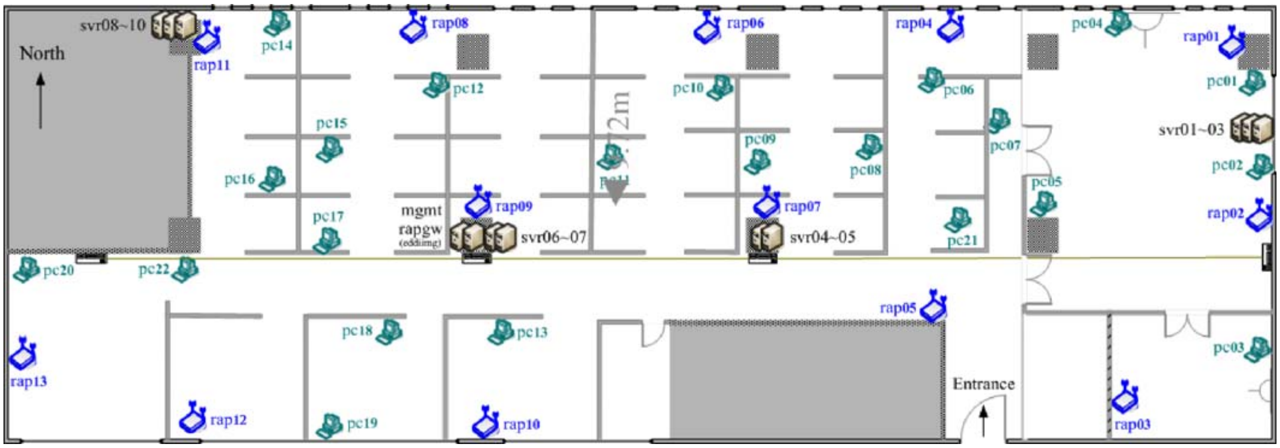

Figure 2 Testbed.

from linear algebra: for $n \geq k$, there exists a set of k vectors $R := \{r_1, \ldots, r_k\}$ such that $r_i^T r_j = -1/(k-1)$ for $i \neq j$, and $r_i^T r_j = 1$ otherwise. We will use these vectors to indicate whether cells i and j use different or identical channels, respectively. Then we formulate our optimization problem as:

$$\text{Maximize (over } x_i \in R\text{): } \frac{k-1}{k} \sum_{1 \leq i < j \leq N} W_{ij}(1 - x_i^T x_j),$$

where x_i is a variable drawn from R that represents the channel assigned to cell i. Defining $X_{ij} := x_i^T x_j$, and discarding constant terms in the objective, we have

$$\text{Minimize (over } X \succ= 0\text{): } tr(WX)$$

$$\text{s.t. } X_{ii} = 1,\ X_{ij} \in \{\frac{-1}{k-1}, 1\}$$

where we wish to minimize over all symmetric positive definite matrices X satisfying the constraints. Relaxing the second constraint as $\frac{-1}{k-1} \leq X_{ij} \leq 1$ then produces an SDP. We can efficiently solve for X with an SDP solver [5], and then apply the polynomial time approximation heuristic of Frieze and Jerrum [6], which uses randomized hyperplane rounding to efficiently generate an approximate solution. To the scale of our testbed, in a few milliseconds the algorithm finds the optimal solution every time.

5. Summary of Dynamic Channel Assignment

Our approach is summarized as follows:

Toward the end of every T second time interval:
1. Report traffic and RSS information to the server.
2. If the network has "sufficiently changed",
   a. Compute co-channel interference between cells.
   b. Compute channel reassignments to minimize sum network interference.
   c. If the potential interference reduction is substantial, then reassign channels.

Note that several variables may be manipulated to adjust the frequency of channel reassignment. Firstly, the time interval T can be made larger to reduce overhead, or smaller to increase resolution in detecting network changes; in our experiment, we used 10 seconds. In the last step, if the network load is too heavy at the moment, or if the expected interference reduction is small, we can opt to not go ahead with channel reassignment. Generally, the frequency of reassignment should be adjusted according to the needs of the application at hand.

6. Implementation Issues

Below we list some issues to clarify our implementation.

- *AP-to-AP RSS Measurements*: To measure RSS values between APs, we conduct a round-robin initialization procedure [7] in which each AP broadcasts a beacon alone while all other APs are quiet and record their RSS values. These measurements are transmitted to the server. Since APs are immobile, we do not expect the network to deviate significantly from these initial measurements, and thus, the AP-to-AP RSS values are kept fixed.
- *Station-to-Station and AP-to-Station RSS Measurements*: Whenever a station has no load, it switches through the available channels and broadcasts probe beacons in each, while also recording RSS values of probe beacons emitted from other idle stations. APs also overhear these beacons and record their RSS values. In this way, station-affiliated RSS values are periodically measured and transmitted to the server, which then updates its RSS information.
- The *sending (receiving) load* of a node is the ratio of observed sent (received) throughput to the maximum possible throughput it can achieve, where the latter is based on its current data rate. In our model, if one wished to replace loads with *demands*, an exponentially weighted moving average may be used to predict demands, as suggested in [8].
- A cell's channel can be changed immediately after the AP broadcasts a special message indicating the change to its clients. This latency would be expected to be around 1-2

ms, which is small compared to the period of time that the cell stays on a channel [9].

7. Performance Evaluation

We set up a 500 M^2 wireless testbed on our office floor containing 13 APs, 22 wireless stations (PCs), and 10 wired servers, as shown in Fig. 2. Each machine is equipped with 802.11/a/b/g NICs using the MadWiFi driver [10]. We used 802.11a, which has more non-overlapping channels in practice. The cards have RTS/CTS disabled, and are set to maximum transmission power. The nodes are positioned sufficiently densely such that almost any two can sense each other.

The 22 wireless stations are uniformly associated with the 13 APs according to their natural spatial proximity; on average, each AP connects with 2 stations. The signal quality between each station to its associated AP is at least 27 dBm, meaning that if there is no interference, an AP or station can use its 54 Mbps physical data rate to achieve a UDP throughput of about 38 Mbps. In all of the following experiments, the traffic pattern is constant bit rate (CBR) UDP traffic, generated at wired servers, and transmitted to the stations through the APs.

Our Dynamic Channel Assignment (DCA) algorithm is compared against a Static Channel Assignment (SCA) that is described as follows: (1) Construct an interference graph by only using RSS information between APs, and (2) Run the MkC solver to compute a one-time channel assignment. SCA models an intelligent RF profiling-based static assignment that does not use traffic or client-side information, and is similar in design to [11]. Such static channel assignment methods are still most commonly used in existing WLANs today.

7.1. Simple Demonstration of DCA

Figure 3 Demonstration of SCA vs. DCA.

We first show a simple experiment to visually illustrate the decisions made by our algorithm. In Fig. 3, we have 7 APs (1,3,5,6,10,11,13) working over 3 available channels, where each channel is indicated by a different color. The small arrows between AP and station indicate traffic flows. We see that there are 4 cells possessing traffic. In the top picture above, we see a channel assignment made by SCA. Since SCA does not consider traffic flows, only two out of three channels are used. On the other hand, the bottom picture in Fig. 3 shows the channel reassignment made by DCA for the same traffic distribution. Large magenta arrows indicate the differences in these two channel assignments. Note that DCA utilizes all three channels. Furthermore, since the number of busy cells is larger than the number of available of channels, we must have at least two cells using the same channel. DCA assures that those two cells are furthest away from each other, thereby incurring little interference. In this simple scenario, throughput gains can be more than 50%.

7.2. Varying Number of Available Channels

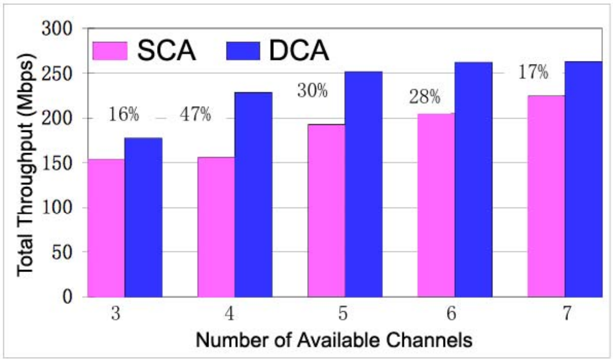

Figure 4 Throughput vs. Number of Available Channels. Bars marked with DCA's percentage improvement over SCA.

In our second experiment, we first specified an initial traffic distribution among 13 working APs, in which each one of the seven APs 1, 2, 3, 6, 10, 11, and 13 was given a heavy flow (> 38 Mbps demand) to send, while the flows through remaining APs are very light (< 1 Mbps demand). Let k denote the number of available channels. We use SCA first and monitor the average sum network throughput over 10 minutes. Afterwards, use DCA instead to dynamically change channel assignments, and monitor throughput for another 10 minutes. We repeat these two steps five times for each k ranging from 3 to 7. Lastly, we average the throughputs over all five trials for DCA and SCA. These averages are plotted vs. k in Fig. 4. Note that since the maximum UDP throughput is 38 Mbps, the sum network throughput can be at most $7 \times 38 = 266$ Mbps. We see that when k=5, DCA almost achieves this upper bound with two channels reused, allowing seven flows to be transmitted without interference. Total throughput under DCA continues to increase and dominate that of SCA as k increases. DCA always makes use of all available channels. However, SCA cannot guarantee this, and thus, on average, it still does not achieve 266 Mbps even when k=7.

7.3. Adaptation to Continuously Changing Traffic

In our third experiment, we set k=5, and change the traffic distribution every 10 minutes. At the start of each iteration, we randomly choose 6 out of the 13 APs to transmit a heavy flow to one of its stations, while the remaining APs engage in light flows. Our DCA will be triggered when there is a sufficiently large change in traffic distribution. The monitored throughput in Fig. 5 plots the continuous traffic throughput under DCA and SCA, and shows that DCA dominates SCA. The throughput gains are roughly 40% on average. This occurs because the traffic is often not uniform amongst channels, and thus DCA can seize opportunities to make use of idle channels, and also reuse channels, at all times. This experiment demonstrates the practical benefits of our algorithm by simulating the continuous traffic flows of the real world.

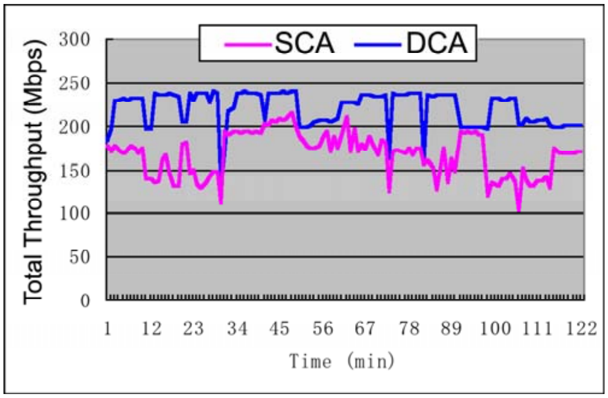

Figure 5 Throughput vs. Time Under Continuous Randomized Traffic.

8. Related Works

The channel assignment problem has been well studied for cellular networks [12], but the solutions there are not readily adapted to WLANs due to many differing characteristics between these networks. For instance, in cellular networks, the regularity of cells allowed for one-shot channel assignments that rarely change; however, the ever-changing nature of WLANs require dynamic solutions that vigilantly monitor interference in an efficient way. In [13], Mishra et al. used the number of clients which are able to associate with two APs to build the metric of interference between any two cells, and reduced the interference minimization problem to graph coloring. This work was extended in [14], where the authors argued that clients have a better view of interference, and hence channel assignment should be conducted from the client perspective. Also, they combined load balancing with channel assignment. However, inaccuracies in their approach arise because client activity is not homogeneous, and neither are the levels of interference they experience. Rozner et al. [8] accounts for traffic, but uses an ideal path-loss model and other assumptions to compute the interference range. Our work is the first to give an approach that accounts for all three of the following: (a) traffic-awareness, (b) client-awareness, and (c) uses a measurement-based interference metric.

9. Conclusions

In this work, we proposed an intuitive model of interference that makes use of both traffic and client awareness, and is measurement-based. We then proposed and implemented an efficient dynamic channel assignment algorithm over WLANs, reducing network interference, and increasing aggregate throughput use by 40% on average over a static channel assignment scheme in our testbed experiments.

For future work, we hope to conduct experiments over a larger testbed, and extend our algorithm to partially overlapped channels.

References

[1] Jim Geier, “Assigning 802.11b access point channels,” Wi-Fi Planet.

[2] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh: "Partially overlapped channels not considered harmful." SIGMETRICS/Performance 2006: 63-74.

[3] V. Mhatre, K. Papagiannaki, and F. Baccelli. “Interference Mitigation through Power Control in High Density 802.11 WLANs,” In IEEE INFOCOM, 2007.

[4] M.X. Goemans and D.P. Williamson. “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming,” J.ACM, 42, 1115-1145, 1995.

[5] CSDP: C Library for Semidefinite Programming. https://projects.coin-or.org/Csdp/

[6] A. Frieze and M. Jerrum. “Improved approximation algorithms for MAX k-CUT and MAX BISECTION.” Algoritmica, 18,1997.

[7] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan, "A General Model of Wireless Interference", Mobicom 2007

[8] E. Rozner, Y. Mehjta, A. Akella, L. Qiu, “Traffic-aware channel assignment in wireless LANs,” in Mobicom 2006.

[9] I. Ramani and S. Savage, “Syncscan:practical fast handoff for 802.11 infrastructure networks,” in Proceedings of IEEE Infocom, 2005.

[10] Madwifi: Multiband Atheros Driver For Wifi. http://madwifi.sourceforge.net.

[11] K. Ishii, T. Osawa, NEC Corporation(JP), “Wireless LAN for reestablishing wireless links between hosts according to monitor desired and undesired signals.” US 6,434,132 B1.

[12] T. Rappaport, “Wireless Communications: Principles and Practice.” Prentice Hall, 1996.

[13] A. Mishra, S. Banerjee, and W. Arbaugh, “Weighted coloring based channel assignment for wlans,” Mobile Computer Communications Review (MC2R), vol. 9, no. 3, 2005.

[14] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, “A client-driven approach for channel management in wireless LANs,” in IEEE Infocom, 2006.