Measurement-Based Channel Management in WLANs

Yongqiang Liu\textsuperscript{1}, William Wu\textsuperscript{2}, Bo Wang\textsuperscript{3}, Tao He\textsuperscript{1}, Su Yi\textsuperscript{1}, Yong Xia\textsuperscript{1}

\textsuperscript{1}NEC Laboratories China  \textsuperscript{2}Stanford University \textsuperscript{3}Chinese Academy of Sciences

Contact email: liuyongqiang@research.nec.com.cn

Abstract — Wireless frequency resources are often the limiting factor for WLAN throughput. Hence, wireless channel management is needed to mitigate co-channel interference and improve channel reuse efficiency in WLANs, particularly in those with high-density access point deployment. The key challenge is to accurately predict interference and its effect on network performance. This paper presents a new interference prediction model that incorporates realistic signal and traffic measurements, and accurately estimates network performance under alternative channel assignments. Using this model, we develop a channel management algorithm that can adapt to dynamic network situations. Experiments on an indoor testbed of 35 wireless nodes demonstrate that our algorithm significantly outperforms existing channel management techniques, with an average throughput increase of about 30%.

Keywords — Interference, Channel Management, WLAN

1. INTRODUCTION

In recent years, the density of IEEE 802.11 wireless networks has dramatically increased. Wireless LANs thus have become more severely affected by interference, which leads to low capacity if not for careful planning. Channel management aims to reduce this interference by carefully scheduling the assignments of frequency channels to access points. The channel management problem is thus to accurately predict network interference under alternative channel assignments so that the best channel assignment can be made.

Let us first analyze the key factors related to wireless interference. Firstly, wireless interference is incurred by simultaneous packet transmissions either over the same frequency channel or an adjacent one. Given two packet transmissions over the same channel, the amount of damage that one packet causes to the other is measured in part by the SINR at a receiver, which is the ratio of the sender’s signal strength to that of the interferers and noise sources.

Early channel management proposals did not involve taking real-world measurements of dynamic quantities such as SINR. They characterized interference using a greatly simplified radio propagation model in which the interference range is twice the communication range [8]. However, this model has been shown to be inaccurate in many scenarios. For example, in Figure 1(a), receivers C2 and C3, who are associated with AP2, both lie within the interference range (dashed circle) of AP1. Assume all distances are drawn to scale. Note that since C3 is both closer to AP1 and farther from AP2 (sender) when compared to C2, C3 may experience stronger interference than C2. Moreover, there may even be no performance degradation to C2 if its sender’s signal power is strong enough to conquer the interfering signal and noise. Therefore, using a fixed, pre-defined interference range does not accurately describe the actual interference level, and more refined signal strength information should be incorporated when doing channel assignments.

Besides signal strength, the amounts of traffic flowing between nodes is another important factor that affects the degree of interference, since the persistence time of an interfering signal is proportional to the amount of traffic from the interferer. Most existing work does not consider traffic amounts due to its dynamic nature, but incorporating traffic is necessary. For example, consider Figure 1(d). Although AP3 contributes a stronger interfering signal to C2 than that contributed by AP1, it is also true that AP3 sends less traffic (2Mbps) than AP1 (10Mbps). If AP2 has to share the same channel as either AP1 or AP3, what channel should be assigned to AP2?

Existing solutions have either incorporated only signal strengths or only traffic. However, neither metric alone is sufficient to make the right decision. Motivated by these examples, the main contribution of this paper is a model that incorporates dynamic real-world signal and traffic measurements to accurately characterize the interference in a wireless network as the network environment changes. Once

Figure 1. Interference varies depending on signal and traffic.
these potential interferences have been calculated, we can assign them as weights on a graph, and finding a channel assignment that maximally exploits channel reuse opportunities then reduces to a graph-partitioning problem, solvable using semidefinite programming. We stress that it is the practical interference model that we are proposing which we consider to be the novel aspect of our work.

For a final example underlining how both signal and traffic are important for channel assignments, consider Figure 1(b), where there are three sender-receiver pairs, and two non-overlapping channels CH1 and CH2. If the traffic over each pair is the same (10Mbps), we should assign different channels to AP2 and AP3 since their potential co-channel interference is high, and let AP1 share the same channel as AP3, since C1 and AP3 are far apart. However, if the traffic between AP2 and C2 is zero, as shown in Figure 1(c), then we should assign AP1 and AP3 to different channels. Even though C2 and AP3 are very close, causing C2 to receive strong interference from AP3, there is no functional impact of this interference since C2 currently has no traffic to receive.

In summary, the merits of our measurement-based channel management scheme are:

i) We incorporate all the signal and traffic factors into one unified model to quantify wireless interference. Most related work considers only one of these factors [2][4][6] or simply adds them together [9], without considering their dependencies. This is not appropriate in many cases. For example, when an interferer has no traffic to send, it actually does not cause interference to others.

ii) We carefully address the feasibility of a measurement-based channel scheme in a realistic environment. The challenge lies in how to deal with dynamic signal and traffic characteristics. The questions are: 1) Does the currently determined channel assignment still work well in future network situations? 2) If not, is the channel assignment algorithm able to adapt quickly enough to match new traffic distributions or new user locations? In this paper, we propose a fast dynamic channel management solution and solve some key practical issues, then validate it on a real testbed rather than simulations.

The rest of the paper is organized as follows. Section 2 describes how to incorporate the real-time measurements into one interference prediction model. Section 3 discusses how to deal with dynamic network characteristics and the key implementation issues. We present the testbed results in Section 4, and summarize the paper in Section 5.

2. A NEW INTERFERENCE MODEL

2.1. Interference Metric

The goal of channel management is to find an alternative channel assignment that optimizes network performance, where in this paper, performance is measured by total network throughput. However, estimating throughput to the letter given signal and traffic parameters can involve a complex computation, such as using a Markov model to approximate the 802.11 distributed coordination function [7]. In a dynamic setting where channel management needs to adapt to the environment quickly, such a time-consuming computation may not be fast enough. Thus we make a practical compromise. Since interference between BSSes is one of the key limiting factors for attaining high throughput—particularly in dense deployment scenarios—we instead try to minimize total interference amongst all BSSes, where interference is now loosely redefined as the amount of extra time required to send a packet in one BSS due to the presence of the other BSS. (More details ahead.) Our experiments show that lowering this notion of interference leads to higher throughput. And since it is not straightforward to maximize throughput directly, we think this is a reasonable approximation.

We now describe our interference model in more detail. The key is to quantify and characterize the interference between any two pairs of BSSes in terms of its effect on network throughput. This interference metric must be easy to compute, use measurements that are easy to monitor, and as discussed in Section 1, it should consider both signal strengths and traffic amounts, as well as their dependencies. To satisfy all these criteria, we propose the following new metric: Virtual Delayed Time (VDT). This describes the potential interference between two BSSes in terms of the extra transmission delays that would be incurred if they were to use the same channel. VDT is motivated by 802.11’s automatic rate adaptation procedure: when the interfering signal at a receiver is strong enough to damage a packet, the sender can lower its data rate to a suitable value to avoid packet loss, as per the SINR-to-data rate lookup table shown in Table 1. Lower rates result in longer transmission times. Therefore, this extra transmission time can be used to indirectly suggest the degree of interference:

\[ VDT = Time^{\text{Interference}} - Time^{\text{Normal}} \]  

(1)

where \( Time^{\text{Normal}} \) is the normal transmission time of one packet in the absence of interference, and \( Time^{\text{Interference}} \) is the prolonged transmission time at the lowered rate due to interference.

The computation of VDT does not require physically changing the channels and measuring changes in transmission time. We only want to predict what channel assignment has the smallest interference, without actually changing channels. As will be shown by our experiments in Section 4, VDT is an accurate enough indicator of interference that our channel management algorithm tends to group BSSes with maximal simultaneous throughput into the same channels.

<p>| Table 1: Some PHY Data Rates and SINR from Atheros Card |</p>
<table>
<thead>
<tr>
<th>802.11 standard</th>
<th>Data Rate (Mbps)</th>
<th>SINR (dbm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>b/g</td>
<td>2</td>
<td>1.6</td>
</tr>
<tr>
<td>b/g</td>
<td>11</td>
<td>7.0</td>
</tr>
<tr>
<td>a/g</td>
<td>18</td>
<td>23</td>
</tr>
<tr>
<td>a/g</td>
<td>36</td>
<td>28</td>
</tr>
<tr>
<td>a/g</td>
<td>54</td>
<td>35</td>
</tr>
</tbody>
</table>
2.2. VDT Calculation

We now present a model for computing the VDT of two BSSes. The model takes traffic load and signal strength of two BSSes as inputs, and outputs a VDT between them. A full list of variables used by the model is presented in Table 2.

Before proceeding to the description of how VDT is computed, some additional remarks on how to estimate SINR are in order, since the transmission time directly depends on the PHY layer data rate, and this rate is closely related to SINR. To indicate signal strength, as in [7], we use the Received Signal Strength Indicator (RSSI), which is reported by many commodity wireless cards upon the successful reception of a packet. For example, in Atheros cards, RSSI is reported as $10 \log \left( \frac{L}{n} \right)$, where $S$ is the strength of the incoming signal, $L$ is the interfering energy in the same channel, and $n$ is the “noise floor” (-95dbm, a negligible value), assuming there are no other noise sources. Therefore, when a receiving node $r$ captures a packet from any other sending node $s$, the firmware of node $r$ reports $\text{RSSI}_r = 10 \log \left( \frac{L_r}{n} \right)$. Note that our convention is to denote the receiver with a subscript, and sender with a superscript. Henceforth, whenever we refer to the power of a signal, or the “measurement” of a signal, we will be referring to the RSSI value of the transmission.

As shown in Figure 2, when the sender $z$ (i.e., AP1) and the interferer $y$ are transmitting simultaneously, assuming the signal power from $z$ is stronger than that from $y$, the SINR at the receiver $x$ can be approximately estimated as

$$\text{SINR}_{x}(\text{dBm}) \approx \text{RSSI}_x^z - \text{RSSI}_x^y$$

(2)

where RSSI$_x^z$ and RSSI$_x^y$ are measured by $x$ when it receives packets from $z$ and $y$, respectively.

To compute the VDT between two BSSes, recalling Eq. (1), we now define the VDT of two nodes $x$ and $y$ as

$$VDT(x, y) = Time_{x,y}^{\text{Interference}} - Time_{x,y}^{\text{Normal}}$$

(3)

where $Time_{x,y}^{\text{Normal}}$ is the normal time required for node $x$’s sender to transmit a packet to node $x$, and $Time_{x,y}^{\text{Interference}}$ is the time to transmit to $x$ under the interference of $y$. We first consider the simplest scenario, where there is only one interferer and one intended receiver (see Figure 2). We have

$$Time_{x,y}^{\text{Normal}} = \frac{l(\text{bit})}{Rate_x}$$

(4)

where the transmission time is normalized by one bit. We multiply by $R_x$ because that is the probability of $x$ having an opportunity in the BSS to receive data. However, in the presence of interference,

$$Time_{x,y}^{\text{Interference}} = \frac{l(\text{bit})}{L_y \cdot Rate_y + (1 - L_y) \cdot Rate_x} \times R_x$$

(5)

where $Rate_x$ is the lowered rate of receiver $x$ under the interference from $y$. This lowered rate is given by

$$Rate_x = \text{Map}(\text{RSSI}_x^y - \text{RSSI}_y^y)$$

(6)

where the function $\text{Map}(\cdot)$ maps SINR to the appropriate rate according to Table 1. $L_y \cdot Rate_y + (1 - L_y) \cdot Rate_x$ in Eq. (5) then gives the average data rate for transmitting to $x$, accounting for interference.

Note that the calculation of VDT does not consider the properties of 802.11 MAC such as Preamble, ACK, Deferrals, because (a) the transmission times for Preamble and ACK frames have no effect on VDT calculation. They have fixed frame sizes, and transmit at fixed rates independent of interference. Thus, these transmission times cancel each other in Eq.(3); (b) taking the deferrals of 802.11 into account would overcomplicate the computations. As stated in Section 2.1, accurately approximating 802.11 DCF requires a Markov model that is too complex for dynamic channel management. We seek a better tradeoff between accuracy and complexity.

To further illustrate how to calculate $VDT(x,y)$, we now use a concrete example with Figure 2. Suppose $Rate_x = 24\text{Mbps}$, $Rate_y = 48\text{Mbps}$, $R_x = 12\text{Mbps}$, $L_y = 12\text{Mbps}$, $\text{RSSI}_x^{\text{PSI}} = 24\text{dbm}$, and $\text{RSSI}_y^y = 10\text{dbm}$. Table 2 indicates that $L_y = 0.25$ and $R_y = 0.5$. With Eq. (4), $Time_{x,y}^{\text{Normal}} = \frac{1}{24\text{Mbps}} \times 0.5 = 20.83\text{ns}$. By
Eq. (6), $Rate^t = \text{Map}(14\text{dbm}) = 11\text{Mbps}$. From Eq. (5),

$$\text{Time}^{(\text{interference})}_{s,b} = \frac{1}{0.25\times11\text{Mbps} + 0.75\times24\text{Mbps}} \times 0.5 = 24.10\text{ns}$$

Therefore, $VDT(x,y) = 24.10 - 20.83 = 3.27\text{ns}$.

For more complex scenarios where there are many nodes in the interfering BSSes, we still use Eq. (4) to calculate $\text{Time}^{(\text{Normal})}_{s,y}$. To calculate $\text{Time}^{(\text{interference})}_{s,y}$, we use

$$\text{Time}^{(\text{interference})}_{s,y} = \sum_{e \in \mathcal{E}} (L_{e} \cdot \text{Rate}_{e}) + (1 - \sum_{e \in \mathcal{E}} L_{e}) \cdot \text{Rate}_{x} \times R_{s}$$

(7)

Accordingly, Eq. (3) changes to

$$VDT(x,y) = \text{Time}^{(\text{interference})}_{s,y} - \text{Time}^{(\text{Normal})}_{s,y}$$

(8)

One thing to note is that if the intended receiver is an AP, the RSSI measurement is different from that of a station since its traffic comes from multiple senders in the measurement period. In this case, the “good signal” (not interference) received by the AP in a BSS$_X$ is $\text{RSSI}_{s,AP} = \sum_{n \in V} (L_n \cdot \text{RSSI}_{n,AP})$.

Now, the degree of the total interference received by BSS$_X$ from BSS$_Y$ can be approximately expressed as

$$VDT_s = \sum_{x \in V} VDT(x,y)$$

(9)

where $VDT(x,y)$ can be calculated using Eqs. (3)–(8).

2.3. Graph-partition channel assignment with VDT

Channel assignment is typically reduced to a vertex coloring problem on a graph. An undirected graph is drawn in which each vertex corresponds to a BSS. The weight of an edge represents the potential interference between two BSSes. The colors represent non-overlapping channels; generally, the number of channels is smaller than the number of BSSes. Minimizing interference then corresponds to minimizing the sum of edge weights connecting vertices of the same color.

We now discuss how to use VDT to allocate channels. In the wireless network graph with vertex set $V = \{1, 2, \ldots, n\}$, each number $n$ represents a BSS in the network. A weight

$$I(X,Y) = VDT^Y_X + VDT^X_Y$$

(10)

is given to each edge $(X,Y)$, where $X$ and $Y$ are vertices. $C$ is a set of non-overlapping channels (e.g. for 802.11b/g, $C = \{1, 6, 11\}$). The channel assignment function CH($X$) assigns BSS$_X$ a specific channel from the set $C$. Our goal is to find the channel assignment that minimizes the total interference:

$$\min \sum_{X,Y \in V} I(X,Y) \cdot 1_{[\text{CH}(X) = \text{CH}(Y)]}$$

(11)

This is the Min K-partition problem [1], and is known to be NP-hard. We use semidefinite programming (SDP) relaxation techniques to get an approximate solution. In particular, the CSDP solver [3] is applied in our experiments and the computation time to find a satisfactory result is 6.2 seconds if $|V|=15, |C|=9$, and 27.5 seconds if $|V|=50, |C|=9$.

3. IMPLEMENTATION ISSUES

In practice, both signal and traffic may change rapidly due to the unpredictable starting and stopping of applications, and user mobility. We now discuss how to handle these dynamic situations, and address key implementation issues.

3.1. Reacting Quickly to Dynamic Traffic

Suppose we have a low mobility application scenario, such as an office environment, where the signal strength among nodes is relatively stable but traffic changes unpredictably. To handle the varying traffic, we use dynamic channel management as a solution, in which the channels are reassigned automatically according to the latest network status. For example, assuming the VDT recalculation period is one minute, we can compute the best channel assignment according to the traffic distribution from the previous minute, if necessary, assign channels accordingly and hold these assignments in subsequent minutes, and so forth. We believe this is feasible because: i) Under a short time granularity, e.g., minute, traffic usually does not change dramatically, and ii) Trivial traffic changes may not impact the algorithm’s performance. For an example of the latter case, suppose the channel assignments are determined by the following policy: the aggregate traffic amount of BSS$_X$ is greater than that of BSS$_Y$. At some point, even if the BSS$_X$ traffic decreases or BSS$_Y$ traffic increases, the channel assignment is still optimal with respect to this policy, as long as the amount of traffic in BSS$_X$ is larger than that in BSS$_Y$.

To implement the fast dynamic channel management algorithm, we must address three practical issues: 1) rapid collection of traffic information, 2) rapid computation of the best channel assignment, and 3) quick execution of channel reassignments. For the first issue, note that since all traffic goes through the APs. And since the APs are connected to a management server via wired lines, collecting all traffic information from the APs can be done very quickly. For the second issue of rapid computation, as shown in Section 2.3, SDP can get the new allocation in a few seconds, which is fast enough for practical use. Then the new channel assignment is distributed from the server to corresponding APs via wired lines and the channel switch time of a BSS is about 20ms in 802.11s.

3.2. Lightweight RSSI Measurements

Aside from traffic information collection, there is also a challenge in measuring the RSSI between nodes from different BSSes operating in different channels, since nodes using different channels normally cannot hear each other. To address this, a measurement scheme from 802.11s can be used in low mobility scenarios. In the scheme, stations pause their communications, switch into the channels of neighboring BSSes to measure the RSSI from neighboring nodes to themselves, and then return to their working channels. However, this scheme does not work well in high mobility scenarios where signal strength pairs vary quickly. In such
and when not employing this compromise, we say it runs in
full information mode. Two natural questions then arise: i)
How much does the discarding of station-side RSSI
measurements affect the algorithm’s performance? and ii)
How well does the algorithm perform in partial information
mode? The experiments in Section 4.3 answer these questions.

4. Evaluation

4.1. Experiment Setting

As shown in Figure 3, the testbed is located in a 500m²
office room. Each machine is equipped with Atheros PCI
802.11/a/b/g cards and uses the madwifi driver. There are 22
wireless stations associated with the 13 APs roughly based on
distance. The signal quality between each station to its
associated AP is good enough to support the maximal
transmission rate if there is no interference. The APs are
interconnected with a control server by wired lines, and we
placed 10 servers (svr01~10) as traffic generators or sinks.

4.2. Performance Comparison

We compared our channel assignment algorithm in full
information mode (abbreviated VDT-CM(F)) and our
algorithm in partial information mode (abbreviated VDT-
CM(P)) with two other schemes. The first is a simple scheme
called Only Signal-based Channel-management (OSC), and is
similar to some manual WLAN planning procedures. In
manual administration, when a new AP joins a WLAN, the
WLAN administrator measures the strength of signals in each
channel, and then assigns the channel with minimal aggregate
signal strength to the new AP. Similarly, under OSC, each AP
selects the minimal interference channel independently when
it starts up, where the interference from AP_i to AP_j is defined
as the power of the received interfering signal sent by AP_j
and captured at AP_i. We call this scheme OSC since it does not
take traffic information into account. The second scheme is
traffic-aware, and is known as LCCS (Least Congested
Channel Search) [5], in which each AP continuously monitors
its assigned channel for data transmissions made by other APs
and their stations. If the volume of traffic in that channel is
greater than a pre-specified threshold, the AP moves to a less
congested channel. How APs scan other channels depends on
the implementation; in our experiment, we use the extra
interface (Section 3.2).

The experimental procedure is shown in Figure 4, where
we run 30 random traffic scenarios with different patterns. For
UDP traffic, it contains 4~8 heavy CBR flows (40Mbps) and
14~18 light voice flows (64Kbps). For TCP traffic, it contains
4~8 heavy FTP flows and 14~18 light HTTP flows. Recall
that unlike UDP, the TCP protocol responds to packet losses
by sharply lowering the size of its sending window, and then

---

Figure 3. The Wireless testbed

Figure 4. High level experiment procedure
increasing it gradually, thereby causing the input traffic to be relatively unstable. In each scenario, the traffic flows are distributed randomly among the APs, independent of channel assignments. Under the same traffic distribution, each of the four aforementioned channel management schemes is used one by one. The achieved total throughput under each scheme is measured. The procedure is repeated for different cases (UDP/TCP, 802.11a/g, for each combination such as UDP traffic in 802.11a network, we run 30 random experiments). Overall, our two schemes always provide superior network performance, as we discuss below.

Due to space limitations, we only show the performance summary of 120 comparative experiments in Table 3 and Table 4. In the first row of the Improvement column of Table 3, a number of 43.5 means that the overall throughput of VDT-CM(F) is 43.5% larger than that of OSC. In the Statistics column, ‘H’ refers to the number of experiments in which VDT-CM(F/P) produced a throughput at least 10% higher than that of the other two schemes, ‘L’ means the throughput was at least 10% lower, and ‘E’ means that the throughputs were almost equal (+10% ~ -10%).

From Table 3, it can be seen that 1) both of our schemes outperform the other two techniques; 2) full information mode is better than partial information mode because it can predict interference more accurately; and 3) VDT-CM(F/P) has more improvement under UDP traffic than under TCP traffic. In fact, LCCS compared against OSC has a similar degradation of improvement (12.3% higher than OSC in UDP while 9.1% higher in TCP). This is because stable input traffic increases the accuracy of interference prediction, making it more amenable to a traffic-aware channel management scheme. By comparing Table 3 and Table 4, we see that the benefits of our VDT algorithms are more pronounced in the case of 802.11g.

To verify that our scheme performs well under a dynamic traffic environment, we ran a long-time experiment with 5 channels in which heavy UDP traffic randomly redistributes itself every 5 minutes. Figure 5 shows the results. Fixed-CH means that the channels are initially assigned by OSC and do not change with the traffic. In contrast, VDT-CM(F) monitors the changes in traffic and dynamically reassigns channels accordingly. The average overall improvement is 40%.

5. SUMMARY

In this paper we analyzed the factors underlying WLAN interference, and proposed a new interference model that incorporates these factors and can accurately quantify network interference under alternative channel assignments. Using the model, we proposed a dynamic channel management algorithm. Experiments on an indoor wireless testbed demonstrate that our algorithm achieves better performance than existing channel assignment techniques, with an average throughput improvement of about 30%.

We close with some additional remarks about our algorithm. Regarding its scalability, if the WLAN is very large, we can split it into multiple smaller clusters by geographic location, and then run our algorithm in each. Lastly, the reader may note that our experiments focused on heavy traffic rather than bursty traffic. Due to the fixed time costs of channel switching and reassociation (100ms to 1s), channel reassignment must operate at coarser time resolutions, such as on the scale of minutes. However, when long-lived elephant flows occupy a major portion of network capacity (as is the case in the current Internet due to P2P file sharing and video streaming), our algorithm will yield good performance.

6. REFERENCES