# A Self-Management Approach to Configuring Wireless Infrastructure Networks

by

Nabeel Ahmed

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2006

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

Wireless infrastructure networks provide high-speed wireless connectivity over a small geographical area. The rapid proliferation of such networks makes their management not only more important but also more difficult. Denser network deployments lead to increased wireless contention and greater opportunities for RF interference, thereby decreasing performance.

In the past, wireless site surveys and simplified wireless propagation models have been used to design and configure wireless systems. However, these techniques have been largely unsuccessful due to the dynamic nature of the wireless medium. More recently, there has been work on dynamically configurable systems that can adapt to changes in the surrounding environment. These systems improve on previous approaches but are still not adequate as their solutions make unrealistic assumptions about the operating environment. Nevertheless, even with these simplified models, the network design and configuration problems are inherently complex and require tradeoffs among competing requirements.

In this thesis, we study a self-management system that can adjust system parameters dynamically. We present a system that does not impose any restrictions on the operating environment, is incrementally deployable, and also backwards compatible. In doing so, we propose, (i) framework for modeling system performance based on utility functions, (ii) novel approach to measuring the utility of a given set of configuration parameters, and (iii) optimization techniques for generating and refining system configurations to maximize utility. Although our utility-function framework is able to capture a variety of optimization metrics, in this study, we focus specifically on maximizing network throughput and minimizing inter-cell interference. Moreover, although many different techniques can be used for optimizing system performance, we focus only on transmit-power control and channel assignment. We evaluate our proposed architecture in simulation and show that our solution is not only feasible, but also provides significant improvements over existing approaches.

# Acknowledgments

I am very grateful to have had the opportunity to work under the guidance of Prof. Keshav. He has always supported me in pursuing my own research interests. He has also gone above and beyond in providing me all the necessary resources that allowed me to make the maximum impact in my work. But more than anything else, I have learnt from him the importance of critical thinking and the need to constantly align one's visions and goals. For all these reasons, I am truly indebted to him.

I would also like to acknowledge Prof. Jay Black and Prof. Paul Ward for agreeing to read my thesis and providing their feedback.

The Tetherless Computing Lab has been an especially exciting place to work, and I have all my fellow colleagues to thank for that: Adi, David, Mirza, Shimin, Darcy, and Majid. My late-night discussions with Adi have been especially helpful in improving the quality of this work. All the members of the Networks and Distributed Systems (NDS) group also deserve thanks for providing their feedback on the ideas presented in this thesis.

I would also like to acknowledge Omar Zia Khan and Kamran Jamshaid in providing for me a window outside of work, and making my time here at Waterloo especially pleasant.

# Dedication

Dedicated to my parents.

For always encouraging me to follow my goals and ambitions and being there for me when I needed them the most.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Today's society is at the crossroads of the next big leap into broadband wireless communications. The grand vision of ubiquitous connectivity, shared by visionaries in industry and academia alike, is making the goal of fourth generation (4G) communication systems an increasing reality. Wireless Local Area Networks, also known as Wireless LANs, play their part in this vision by supporting a variety of multifaceted roles, from serving as low cost alternatives for in-home wireless coverage, to providing complete blanket city-wide coverage [10, 8]. Of course, Wireless LANs have come a long way since the early days, when the technology was used primarily to support secure military communications in combat environments [32]. Today, there are an estimated 32,800 WiFi hotspots throughout the United States [85], with these numbers expected to grow to over 60,000, by 2008. Moreover, wireless solutions providers such as Cometa Networks [36] have announced intentions for a nationwide WiFi build-out consisting of an additional 20,000 hotspot nodes, by the year 2007. The wireless user base is also growing at an alarming rate. Gartner estimates indicate that the number of wireless users was expected to grow from 9.3 million to over 30 million, across the globe [81], in one year alone.

The cost of wireless technology has been the major driving factor in its widespread adoption. In earlier years, as the cost of wireless technology decreased and its quality improved, enterprises found it economical to use the technology for their corporate needs (e.g., as an inexpensive alternative to connecting corporate campuses). Today, wireless technology is easily within grasp of the general consumer. In fact, based on recent figures, an IEEE 802.11-compliant wireless USB adapter that provides wireless connectivity for the home or office, can easily be purchased for under $10.00! [35] As the sales of these wireless devices continue to increase and the cost of manufacturing falls, we will likely see a further growth in the number

of wireless deployments. Furthermore, we can also expect this technology to find application beyond providing just traditional wireless LAN access. We are already seeing examples of such applications in industry [21, 70].

The rapid adoption of wireless LANs will require an equally strong emphasis on their management. Wireless LAN management is surprisingly more complex than managing a wired LAN. As the size of the network increases, the management complexity multiplies. As an example, for a wireless network deployed at an IT department in Microsoft, there were an average of over 600 network-related faults per month, over a period of 6 months [28]. This resulted not only in lost productivity but also in an increase in the maintenance cost of the network. Due to the inherent complexities of the wireless environment, it is hard even for a network administrator to pinpoint the root cause of such problems. This calls for mechanisms that are provisioned in the infrastructure itself to support such management. Not surprisingly, similar efforts are being made even for wired networks, where the complexity of the system undermines the ability of humans to manage successfully [44].

## 1.1 Problem Definition

Self-management capabilities equip Wireless LANs with the ability to configure and heal themselves. They prove useful for supporting two key tasks: network deployment, and network management.

- *Network Deployment:* Wireless network deployment refers to the process of placing and configuring wireless access points to meet a specified set of objectives. Standard industry practices involve manual or virtual site surveys that can be time-consuming, costly, and inaccurate. A self-configuring system can enable rapid network deployment by determining the most suitable configuration automatically, based on the surrounding environment. This can reduce deployment time and cost significantly, as well as improve accuracy.

- *Network Management:* After deployment, the next step involves management of the network. "Management" can refer to a number of tasks. It could describe the process of monitoring and configuring the system to maximize performance. It may also refer to monitoring for the purposes of troubleshooting and fault-diagnosis. Finally, it could also be used in the context of security, where the system performs introspection to detect malicious entities in the network (e.g., rogue hardware). Nevertheless, whatever the goal, all

such management tasks require an intimate understanding of the internal structure of the network and its corresponding state transitions, something that is best understood by the system itself.

## 1.2 Problem Characterization

In this thesis, we address self-management with the goal of improving network performance. We first delineate the steps that are required for building an effective self-managing infrastructure to support this goal.

1. *Identifying System Objective(s):* The first step for self-management involves identifying the objectives of the system, including deciding what metric(s) define performance. Examples include maximizing network throughput, minimizing end-to-end delay, or providing sustainable client bandwidth (for real-time applications). Moreover, these objectives may also evolve in space and time. For instance, system objectives may vary across the deployment region, where some regions optimize for one metric and others optimize for other metrics. Performance metrics may also change over time, depending on the usage requirements of the system. Many commercial systems [4, 6, 23] do not provide the flexibility to alter system objectives, resulting in deployments that have multiple overlapping and often conflicting vendor solutions [18].

2. *Performing Introspection:* The second step involves deciding how the system measures these performance parameters. The level of sophistication used in this process determines the amount of accuracy that can be achieved in measurement. For instance, in detecting RF interference, if we simply measure the drop in application throughput as an indication of interference, our degree of accuracy in inferring interference would be low. However, if we perform introspection at the MAC or physical layer, our ability to accurately infer RF interference would be high. Therefore, the layer of the network stack at which the introspection is performed critically determines the accuracy of the technique. Recently, there has also been an interest in applying tools from AI and cognitive systems to infer high-level system state [44]. These techniques will likely prove useful for self-managing systems in the near future.

3. *Reacting to Change:* Once the mechanisms to measure performance are in place, the final step involves deciding how to react in response to a change.

This is necessary to allow the system to provide good performance consistently. To achieve this, there could be multiple *tuning knobs* that are available for the system. The choice of tuning knobs determines how accurately performance can be tuned. Note, tuning knobs may also be inter-dependent, where the tuning of one affects the tuning of the other. Channel assignment and transmit-power control are an example of such a dependency. To address this difficulty, we can induce separability between parameters where we ignore the effects of one, while doing tuning for the other. This approach is well-known in classical control theory and can be applied to the problem we discuss here [46].

Note, the general approach to self-management we described above bears striking resemblance to ideas from classical control theory, that posit the use of a "control-feedback loop" to manage different types of control systems. This similarity in objectives did not occur by accident, as these are exactly the ideas that embody self-management. In the architecture we present in this thesis, we apply similar concepts to construct a dynamically configurable system that is able to maximize wireless LAN performance.

## 1.3 Motivation

To motivate the problem we address in this thesis, we briefly describe a few challenges and unsolved issues for wireless LANs that make their self-management a key requirement.

- *RF Interference:* RF interference is the major culprit in degrading network performance. It is defined as the suppression of communication between two nodes due to simultaneous communication by two or more other nodes. Although much work has been done to address this problem, this issue remains largely unsolved due to poor modeling of the problem. Moreover, detecting the existence of RF interference manually can be extremely difficult, due to its high degree of variability. This requires system-level mechanisms for proper detection and resolution. The need for such systems is crucial as more and more devices penetrate the consumer market, with all of them sharing the same unlicensed spectrum with wireless LANs.

- *Non-uniform Coverage Areas:* A major pain-point for wireless networks is that they have irregular coverage areas. As a result, it becomes impossible

to accurately predict the signal coverage for a wireless network. This unpredictability leads to the creation of dead-spots in the deployment. One way to solve this problem is to simply deploy additional access points to guarantee coverage. However, this not only increases cost but can also introduce a significant amount of RF interference. Therefore, we need to make a difficult tradeoff between the amount of coverage we desire and the amount of interference that can be tolerated. This difficult decision, however, can be avoided if we can somehow acquire information on the actual coverage. A self-managing system has the ability not only to furnish this type of information, but also to react in response to it. Note, sophisticated techniques to obtain this information manually can also be used, however, we run into a second problem with coverage areas, which we highlight next.

- *Dynamic Coverage Areas:* In addition to irregular coverage, coverage areas are also dynamic. Therefore, the actual signal coverage at a given point in space can change over time. This can be induced simply by people moving in the environment, opening/closing doors, or moving a metal cabinet from one location to another. Without mechanisms to detect these changes in signal coverage, wireless LANs cannot provide any guarantees on service quality. Dynamic self-managing systems provide the only means to solve this problem that cannot be addressed using static approaches.

- *Asymmetric Channel Conditions:* Wireless LANs were built on the principle of decentralized communication, where each device itself decides when to access the shared wireless medium, without any centralized coordination. Although this motivates a simpler design, it also creates problems where some nodes receive poor service, due to asymmetry in channel conditions. This is also referred to as the Near/Far problem, where a node that is far away from the access point gets poor service due to a higher-powered node closer to the access point. Unfortunately, there is no way to address this problem without first creating mechanisms to detect it and then taking appropriate actions to resolve it. In this scenario, the access point could initiate an RTS-CTS exchange with the client to ensure the client gets access to the medium. This represents yet another situation where self-management can prove useful, by catering to the needs of the clients.

Based on this discussion, the need for a self-management infrastructure for Wireless LANs is evident. However, we identify a set of additional requirements that need to be fulfilled, in order to construct rapidly deployable wireless networks with good performance.

- *Backwards-Compatibility:* A key requirement for any self-managing system is the need to support existing technology and protocols. It is unreasonable to expect existing systems to cater to any specific needs for self-management. Doing otherwise restricts the wider application and easier integration of the system.

- *Incremental Deployment:* One can expect that any self-management solution will be required to support richer functionality than that for which it was originally intended. Therefore, it is imperative that the architecture provide the appropriate hooks for its extension, to cater to the future needs of wireless LANs. This feature is lacking in almost all the existing self-management solutions we analyzed.

- *No Modeling Constraints:* Finally, any solution to self-management ought not to make any unrealistic assumptions about the network or its operating environment. Doing so will yield solutions that will almost surely fail in a realistic setting. Ideally, the infrastructure should be able to provide good performance despite the existence of RF interference, irregular/dynamic coverage areas, and asymmetric channel conditions.

These are the issues that we address in our solution.

## 1.4 Contributions

This thesis makes the following contributions towards building a practical self-management architecture for wireless LANs.

1. An infrastructure-based solution that does not require client-side modifications. This allows the architecture to be *backwards compatible*.

2. A utility model that provides a unified framework for capturing any performance objective, and even multiple objectives. The use of an extensible utility function supports *incremental deployment*.

3. An extension of the general conflict graph (called the 'Annotated' Conflict Graph or ACG) to represent utility and assist in optimizing performance.

4. An novel experiments approach for detecting and quantifying RF interference that is used to annotate the conflict graph. This frees us from the constraints of using an abstract *network model* for identifying RF interference.

5. The use of a centralized architecture to support global optimization and access-point coordination that provides better results than local tuning. Heuristics for optimal channel assignment and power control, using this approach, are proposed.

6. A dynamically re-configurable framework that can refine configurations in response to the changing RF environment. Techniques for inferring and detecting change are proposed.

7. Preliminary simulation results to validate the design and performance aspects of our proposed architecture.

## 1.5 Outline

Chapter 2 presents some background material that is relevant to understanding the techniques we use in our architecture. Chapter 3 discusses related work in three categories: work that addresses similar problems, work that proposes a self-managing system for wireless LANs, and finally work on sub-problems related to self-management. Chapter 4 presents a preliminary solution to static optimization, which we explore to develop a better understanding of the form for the more complex dynamic (self-managed) optimization solution. Chapter 5 presents the dynamic optimization solution that we propose in this thesis. And finally, Chapters 6 and 7 present our conclusions and some directions for future work.

# Chapter 2

# Background

This chapter briefly covers some details of the characteristics of wireless networks and the IEEE 802.11 standard that are relevant to this thesis.

## 2.1 Wireless-Medium Basics

In this section, we briefly describe characteristics of the wireless medium that are relevant to our work.

In any wireless environment, the goal of a transmitter is to transmit a radio-frequency signal that can be decoded correctly by the receiver. However, this goal cannot be met if the receiver is not within a certain distance of the transmitter. Because the wireless signal undergoes RF attenuation (i.e., weakening of the signal), if the receiver is far from the sender, it may not be able to decode the signal correctly. Furthermore, if the receiver is too far from the transmitter, the received power may be too weak to even be detected by the receiver. The ability to detect a signal is based on the carrier-sensitivity threshold (CST), defined by the receiver. The CST indicates the minimum power/energy that a node must receive in order to detect a change in the state of the wireless channel. Most wireless-card manufacturers conservatively set this threshold to -85dbm to prevent RF interference. The effect of signal attenuation can be captured with the help of ranges that are defined by the transmitter. These ranges define the quality of the received signal, based on the distance from the transmitter, and are illustrated in Figure 2.1.

- **Transmission Range:** The transmission range is the range within which the receiver of a packet can receive and decode the packet correctly. This

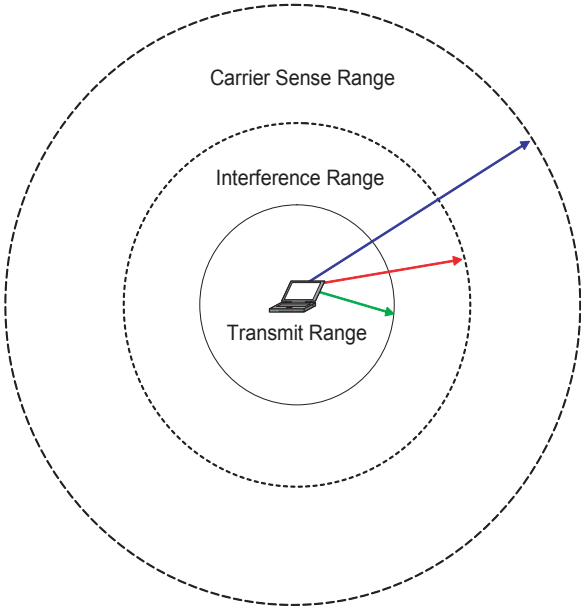


Figure 2.1: The transmission, interference, and carrier-sense ranges defined by the transmitting node.

is typically much smaller than the interference/sensing range of the receiver (e.g., it is typically half the interference range).

- **Interference Range:** The interference range is the range within which the transmission cannot be decoded correctly by the receiver but is of sufficient power/energy to disrupt the correct reception of other packets that the receiver could also be receiving.

- **Carrier Sense Range:** This is the range where the transmission does not necessarily interfere with other packets being received by the receiver. However, the power of the transmission is sufficiently high to exceed the CST of the receiver. The receiver will sense the channel to be busy and thus will not initiate transmissions. The carrier-sense range is affected by the transmission power of the transmitter. The receiver can choose to de-sensitize itself with respect to this range by reducing its sensitivity threshold [81].

The three ranges specified above are affected by the power of the transmitter. The greater the transmit power, the more nodes that can receive the transmission, and also the more nodes whose communication with other nodes may be affected.

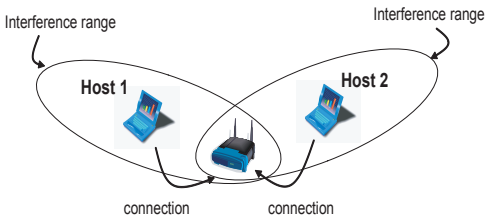


Figure 2.2: Host 1 is hidden from Host 2 and vice versa.

Further, we discuss two specific problems that arise as a result of these characteristics.

## 2.2 Hidden and Exposed Terminals

Two problems we encounter in the wireless world are the hidden and exposed terminal problems.

The hidden terminal problem occurs when a transmitter is unaware of or not able to detect the presence of other transmitters, which are interfered with by the transmission of this node. The interfered transmitters are *hidden* from the interfering transmitter. There many scenarios in which hidden terminals can arise. One of these scenarios is illustrated in Figure 2.2. This particular scenario can be solved trivially with the help of the RTS-CTS mechanism, discussed below. However, the general problem is hard. The experimental framework we discuss later detects hidden terminals that may arise in infrastructure deployments.

The exposed terminal problem refers to the inability of transmitters to utilize the wireless medium for transmission simultaneously, even when there is an opportunity to do so. In this scenario, the transmitters do not transmit because they lie in each other's carrier sense range. In this situation, the transmitters can transmit without affecting with each other, but do not because they sense the channel to be busy. These transmitters therefore act as *exposed* terminals for each other. This is illustrated in Figure 2.3. Such problems can be resolved by adjusting the carrier sensitivity threshold (CST) of the transmitters [81]. However, because we do not use this parameter in our study, we do not address these problems in our work. For the sake of simplicity, we assume that the CSTs at the receivers are set so that a

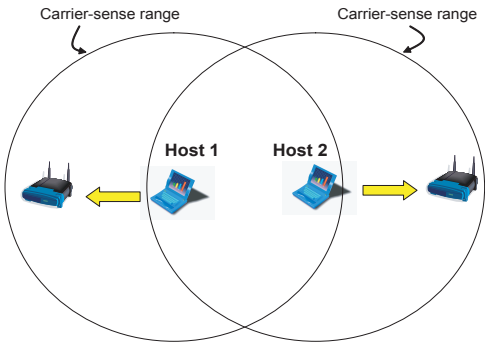


Figure 2.3: Host 1 and Host 2 reside in each other's carrier-sense ranges and thus cannot transmit simultaneously.

transmitter's carrier-sense range is equal to its interference range, with respect to the receivers. This can be done trivially for modern wireless LAN hardware.

## 2.3 IEEE 802.11 MAC Layer

The properties of the IEEE 802.11 MAC layer that we discuss here are the access mechanism (CSMA/CA), the use of RTS-CTS control packets, and finally the implementation of broadcast packets.

### 2.3.1 Channel Access

The IEEE 802.11 MAC layer provides a physical carrier-sensing mechanism called Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), to coordinate access to the shared wireless medium. The fundamental difference between wired and wireless networks is the mechanism for detecting collisions on the shared medium. In the wireless world, it is impossible to detect collisions "on-the-air." Hence, the protocol uses a Collision *Avoidance* mechanism, as well as positive acknowledgments (or ACKs) whenever a packet is transmitted. ACKs assure the sender that there was a correct reception of the packet by the receiver. The protocol avoids collisions by using a random back-off period (called back-off slots), if stations detect the medium is busy. Aside from the random back-off slots, standardized time spaces, called Inter-Frame Spaces (IFS) are also used to coordinate access to the

medium. IFS intervals provide a mechanism to synchronize transmission events in the wireless network and also to prioritize different types of traffic. We briefly discuss the IFS intervals relevant to our work.

- *Short Interframe Space (SIFS):* SIFS is the shortest fixed interframe space. SIFS intervals are used before the sending of RTS, CTS, or ACK packets on the medium. Because SIFS is the shortest of the IFS intervals, RTS/CTS and ACK packets get the highest-priority access to the medium, over regular data traffic. The typical duration of a SIFS period is $10\mu s$.
- *Distributed Coordination Function Interframe Space (DIFS)*: DIFS is the longest interframe space and is used by default on all 802.11-compliant stations that use the distributed coordination function (DCF). DIFS is therefore the interval after which data frames can be transmitted. Any station wishing to transmit data needs to wait a DIFS period before gaining access to the medium. The typical duration of a DIFS period is $50\mu s$.

There are two other IFS intervals that are specified by the standard. However, we do not present their details in this thesis. In the next section, we discuss how virtual carrier sensing is supported in the IEEE 802.11 standard.

### 2.3.2 RTS-CTS Control Packets

The virtual sensing mechanism supported by the IEEE 802.11 standard uses optional RTS/CTS control packets and a *Network Allocation Vector* (NAV) at each node. The purpose of the virtual sensing mechanism is to allow the sender to reserve the channel before transmitting the actual data packet. Failures in reservation, due to a collision, cause less wasted air time. This procedure works as follows. The sender first sends a Request to Send (RTS) packet to the receiver. Upon receiving the RTS, the receiver replies with a Clear to Send (CTS) packet. Once the CTS is received correctly by the sender, the channel has been reserved for data transmission. During the RTS-CTS, each node in the vicinity of the sender that can hear the RTS backs off from accessing the medium. The back-off value is the time required to complete the transmission requested by the sender; this is encoded in the NAV field of the RTS packet. The same information is also specified in the CTS packet for nodes in the vicinity of the receiver. Therefore, using this technique, if the RTS-CTS exchange is successful, a sender is able to successfully complete a data transmission without experiencing collisions from other nodes.

### 2.3.3 Broadcasts

The last part of the IEEE 802.11 MAC layer that we describe here is broadcast packets. Broadcast (and multicast) packets are packets where the intended recipient is not unique. In this case, it is neither possible nor required for the recipients to acknowledge the reception of the broadcast packet. They are useful for sending *one-way* (UDP-style) packets (at the MAC layer), where acknowledgements are not required by the sender. Such mechanisms prove useful for our experimental framework, described later.

## 2.4 IEEE 802.11 PHY Layer

The parts of the PHY layer that are relevant to our work are the physical-layer frame structure, and the process by which frames are received by the air interface.

The IEEE 802.11 PHY Layer is divided into the Physical Layer Convergence Procedure (PLCP) sub-layer and the Physical Medium Dependent (PMD) sub-layer. The PMD sub-layer interfaces directly with the wireless medium and provides modulation/demodulation capabilities for frame transmissions. The PLCP sub-layer provides a mapping for frames communicated between the MAC layer and PMD sub-layer. This partitioning of functionality is done to reduce the MAC layer's dependence on the PMD sub-layer.

### 2.4.1 PPDU Structure

We now discuss the PLCP protocol data unit (PPDU) constructed by the PLCP sub-layer and communicated to the PMD for transmission. The PPDU structure for the IEEE 802.11b Direct Sequence Spread Spectrum (DSSS) standard is illustrated in Figure 2.4.

The transmission rate of the sender is encoded in the signal field of the PLCP header. The signal field also identifies the type of modulation that the receiver must use to decode the signal. The transmission rate can be obtained from this information by dividing the value in the signal field by 100kbps. In Section 5.3.1, we discuss how this information can be inserted into our utility-based objective function, for quantifying the utility of wireless clients connected to a network.

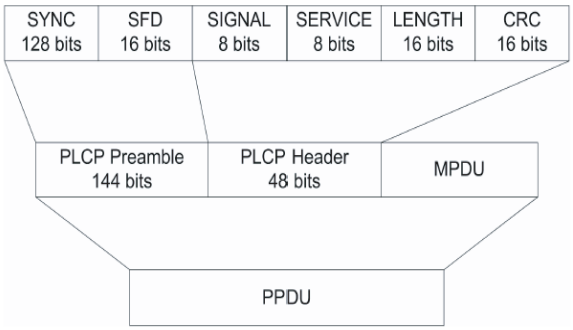


Figure 2.4: DSSS PPDU, 802.11-1999 (Revision 2003) Standard

### 2.4.2 PLCP Function

We now discuss the procedure by which the PLCP sub-layer detects and receives a packet transmitted by some other node (shown in Figure 2.5). At the time instant before the reception of a signal, the PMD sub-layer is in carrier sensing mode. At first, when it detects a signal, it receives the PLCP preamble, which consists of the SYNC (or synchronization) and Start Frame Delimiter (SFD) fields. The synchronization field is used by the receiver to synchronize with the received signal. Once it detects the SYNC field, the PMD sub-layer notifies the PLCP layer (PMD_CS), which subsequently notifies the MAC layer (PHY_CCA.ind(BUSY)), that the channel is busy. This provides the MAC layer an indication that a PLCP header is going to being received by the PHY layer. If the header is received correctly, then the PLCP layer receives and forwards the packet as octet streams of data, to the MAC layer. If the header or any subsequent data streams are not received correctly, it sends a PHY_RXEND.ind(RXERROR) message to the MAC layer indicating an error in reception, and then transitions to the idle state. This error status and change in channel state can be used by the MAC layer to identify packet *corruption* and the presence of RF interference. These types of state changes of the wireless channel can thus be used to detect the presence of interference. We elaborate on how we use these techniques in Chapter 5.

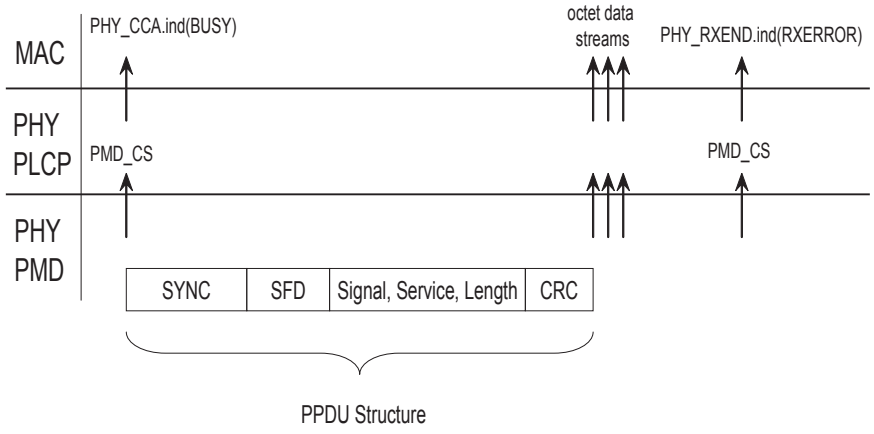


Figure 2.5: PLCP reception procedure

## 2.5 RF Interference

Radio-frequency interference is the single biggest culprit that affects performance of wireless LANs. Recall, it is defined as the suppression of communication between a wireless sender and receiver because of simultaneous or overlapping communication between one or more other wireless senders and receivers. Interference affects both the coverage and capacity of the wireless network, and is an increasing concern for networks that co-exist in the same geographical space. Because RF interference is intimately related to the self-management problem, we discuss it in detail.

We first identify two sources that can contribute to RF interference.

1. *Internal Devices:* Internal network devices that constitute a part of the network may induce co-channel interference on one another if they are not spaced far enough apart. There are many scenarios that can be constructed in a wireless LAN deployment where interference between such communicating parties can occur. In Chapter 5, we provide a thorough evaluation and classification of different scenarios of RF interference between internal network devices. We call this type of interference *internal interference*.

2. *External Devices:* External devices can either be other network devices that are not part of the network under consideration, or simply uncontrollable RF interference sources such as cordless phones and microwave ovens. Because

wireless LANs operate in the unlicensed ISM/UNII bands, they experience interference from other devices operating in the same band.

There are two steps that need to be followed to reduce the impact of such interference. The first is detecting the source of the interference, while the second is mitigating its effects. It is not always possible to detect sources of interference, especially those that are external to the network. Recently, some sophisticated techniques (such as *RF fingerprinting*) for detecting external interference have been used for deploying high-end systems [15]. RF fingerprinting involves receiving a raw RF signal, processing it, and performing pattern matching on it. Pattern matching is used to detect the source of the interference by comparing the signal against well-known RF signal patterns [80, 75]. This method, however, is not perfect. It is also sensitive to the frequency range considered for analysis, the sensitivity of the analyzer, and its frequency resolution. Moreover, it requires the use of expensive spectrum analyzers that are not always within the budget of most wireless deployments. Therefore, for our purposes, we focus only on detecting internal interference.

In the next section, we discuss a variety of wireless LAN tuning parameters that can be used to reduce the impact of RF interference, and also improve performance in other ways such as maximizing coverage and/or capacity. With the exception of a few, most of these parameters can be adjusted in modern wireless LAN hardware. We present them to illustrate the degree of flexibility in combating performance problems in wireless LANs that is typically available in modern hardware.

## 2.6 Performance Tuning Knobs

We discuss eight distinct parameters that can be used to tune wireless-LAN performance.

1. *Channel Assignment*: Every access point must be assigned a certain frequency (or channel) on which to operate. If we consider IEEE 802.11b, we only have three non-orthogonal channels to choose from. In many scenarios, the number of deployed APs is larger than the number of available channels. In these cases, channel re-use needs to be employed to allow multiple APs to use the same set of channels. The general rule of thumb in these cases is to assign different channels to APs that are co-located, to prevent RF interference. However, this approach is far from precise, and RF expertise

coupled with sufficient deployment experience are required to minimize the effects of interference.

2. *Transmit-Power Control*: Transmit-power control provides a number of features such as increasing network coverage, reducing RF interference, improving network capacity, and reducing energy consumption. Due to the multidimensional nature of the parameter, transmit-power must be controlled carefully to maximize performance. It is also affected by the channel assignment process, where a different assignment of channels yields a different solution for power control. This is because APs that are assigned the same channel can interfere with each other if their coverage areas overlap, which is determined by the transmit power of the APs. In Chapter 5, we discuss how we go about solving this problem in our architecture.

3. *Fragmentation Threshold:* The fragmentation threshold defines the upper limit on the packet size that can be supported during a single transmission. Packet sizes larger than the threshold are fragmented before transmission. This is useful in the following way. Channels on which nodes communicate can exhibit temporal changes due to interference. This results in an increase in the bit-error-rate (BER) for that channel. In this scenario, in order to increase the probability of successful transmissions, a smaller transmission frame size can be used [59]. The frame fragmentation threshold can be set so that frame transmissions match time periods within which the channel is free of interference.

4. *Receiver Sensitivity Threshold:* The receiver sensitivity threshold, also known as the carrier sense threshold (CST) provides a means to solve the exposed terminal problem in wireless LANs. In this case, increasing the CST allows a node to desensitize itself from other nodes' transmissions that may not necessarily impact its own transmissions. Interest in using CST thresholds for performance tuning purposes has only recently gained momentum [81].

5. *Transmission Rate:* The transmission rate dictates the bandwidth at which the access point communicates with wireless clients. It can be tuned to improve client performance. Although this feature was not supported separately from transmit power control in the past, most modern wireless cards do support tuning transmission rates separately.

6. *Time-Slotted Activation*: Time-slotted activation refers to the ability to activate/deactivate APs at scheduled intervals, so that no two APs that are

co-located and using the same channel are activated at the same time. Therefore, the goal is to assign APs that use *orthogonal* channels to a common time slot, in order to maximize performance.

7. *Phased Antenna Arrays*: Phased (or Sectorized) antenna arrays have only recently been commercialized in commodity wireless LAN hardware [26, 16]. Their goal is to increase spatial diversity in a wireless LAN environment [83]. They facilitate this by supporting fewer users per region (or sector) and thereby reducing the amount of user contention. They also allow parallel transmissions in each sector and can therefore provide significant improvements in network capacity.

In this section, we discussed background material relevant to the ideas we discuss in Chapters 4 and 5. In the next section, we discuss existing work related to self-management of wireless LANs.

# Chapter 3

# Related Work

This chapter discusses work related to self-management of wireless LANs. We first describe some techniques that address similar problems. In doing so, we identify the limitations of these approaches. Then we discuss some existing systems that have been proposed for self-managing wireless LANs, followed by a discussion of some sub-problems related to self-management. We do this to identify the limitations of currently proposed approaches to self-management. Finally, we present emerging standards that complement both our work and the area of self-managing wireless LANs in general.

Techniques that solve management problems for wireless LANs fall into two broad categories: static optimization and dynamic optimization. Self-managing systems are a form of dynamic optimization.

## 3.1 Static WLAN Optimization

A static optimization solution involves placing and configuring access points optimally at deployment time. This configuration is assumed to be static for the lifetime of the deployment. Static optimization is used widely in practice and relies on wireless site surveys for its solution. There are two standard site-survey techniques that are used, manual and virtual.

- *Manual Site Survey:* The earliest known static optimization solutions perform a manual site survey of the building to determine the best placement and configuration of access points [32]. In such surveys, an RF expert typically obtains floor plans of the building and annotates them with RF measurements

taken at different locations of the site. Using this information and basic rules of thumb, she then places access points and configures them appropriately. This process, although summarized here, involves many detailed steps. A manual site survey can quickly become very complex, depending on the deployment size, and requires both extensive RF expertise and experience in wireless LAN deployment.

- *Virtual Site Surveys:* Virtual (or software-based) site surveys speed up deployment time and reduce cost [20, 22, 48]. They allow the network planner to import a map of the building in software and annotate it with building-specific information (e.g. wall thickness, construction material, etc.). Access points can then be placed on the map and their signal coverage predicted using a variety of wireless propagation models. Propagation models approximate the physical effects of the environment on the propagation of wireless signals within a given geographical space. There are two types of models: empirical models, and models based on ray-tracing. Empirical models are based on statistical measurements that are collected in real environments. Common empirical models include the Log-Distance Path Loss model and the Partition and Floor Attenuation Factor model [75]. Ray-tracing models attempt to simulate the physical effects of the environment on the wireless signal. Depending on the complexity of the modeling, effects such as reflection, refraction and scattering can be captured. However, the greater the complexity, the greater the computational requirements of the model. Therefore, many techniques try to reduce modeling complexity by performing simplifications on these models to lower their computational cost [49, 86].

We now outline some limitations of static optimization. First, it is costly and time-consuming. Depending on the size of the deployment, manual site surveys can take from a few weeks to several months. Virtual site surveys solve this problem but still require collecting detailed information about the site in order to produce good configurations. Second, static optimization is based on a premise that is fundamentally flawed. It assumes that the RF environment is stable and predictable over time [32]. On the contrary, the RF environment can change significantly, even by the movement of people and minor restructuring of the environment (e.g., moving a metal cabinet a few centimeters). Moreover, with wireless LANs sharing the same spectrum with many other devices, RF interference can become an increasing problem in wireless deployments (e.g., with the presence of cordless phones). In such scenarios, at any point in time, a static network configuration cannot exploit parts of the spectrum that are under-utilized, to provide better performance. Finally, organizational changes (e.g., corporate restructuring) also affect static deployments,

where previous RF measurements and/or predictions are no longer valid because of changes in the propagation environment. This leads us to an important point. There likely will never be a single optimal configuration that can last the lifetime of the deployment. In fact, the configuration will need to evolve over time to provide consistent performance. This calls for a dynamic solution to the problem.

## 3.2 Dynamic WLAN Optimization

Dynamic wireless LAN optimization provides a means for dynamically configuring the wireless network. Such techniques fall into two categories: network monitoring with manual configuration and network monitoring with automatic configuration. Self-managing systems support automatic configuration.

### 3.2.1 Manual Configuration/Automated Monitoring

An automated monitoring system periodically acquires network state to decide whether changes are required to the network configuration. Monitoring is done using either the existing network, or through out-of-network devices (e.g., wireless sensors) that periodically probe and measure different statistics of the network. This information is then collected at a central site where an administrator can analyze it and make any necessary configuration changes. SNMP-based management tools have been proposed for this purpose, and are typically supported for enterprise wireless LAN management [32].

Many specialized hardware monitoring tools have been developed for accurately diagnosing performance problems in wireless networks. These devices support techniques such as passively sniffing on-air traffic to discover network state. They can also be used for actively probing the network, by pumping packets into it. Example tools include AirMagnet's Enterprise device [14], WildPacket's RFGrabber [25], and the Omni-Wireless Sensor [24]. Cheaper alternatives using general-purpose hardware have also been proposed, such as a measurement system that uses a dense deployment of wireless USB adapters plugged into the existing desktop infrastructure for monitoring purposes [35]. The reduced cost per device leads to denser deployment. However, for hardware monitoring tools in general, their effectiveness is dependent on their capabilities, their spatial location in the environment, and the techniques used to combine statistics from all monitoring devices [87]. Additionally, the network view captured by these devices may not reflect the actual view of the network, due to path diversity in the wireless environment.

Alternative techniques include monitoring the wireless medium through the in-network devices themselves, i.e. the clients/APs. Qiu et al. [28], provide a fault-diagnosis architecture where clients report problems to the network through other connected clients. Another approach [66, 67] uses client site reports to obtain network state. These techniques accurately capture the actual view of the network but may come at the cost of reduced performance or increased energy consumption. These issues are elaborated in Chapter 5.

### 3.2.2 Automated Configuration/Automated Monitoring

Recently, there has been an interest in developing self-managing systems that configure and heal themselves in response to changes in the wireless environment. Many startups have emerged in the last few years that propose such systems [4, 6, 11, 19, 22, 23]. At a fundamental level, all these systems can be classified into one of two categories: decentralized fat-access-point architectures, or centralized thin-access-point architectures.

- *Decentralized Fat Access Points:* Decentralized fat access points are access points that have all the intelligence (i.e., measurement and configuration capabilities) built into them. Therefore, they either sense the wireless environment and unilaterally decide the best configuration for themselves, or coordinate with each other to agree globally on the best configuration. AutoCell [4] supports access-point coordination where access points communicate with each other in a decentralized fashion. On the other hand, Engim [2] access points do not coordinate, and simply decide on the best configuration by sensing configurations of neighbouring access points. Akella et al. [31] also use similar mechanisms (i.e., using RSSI and packet delivery statistics) for the purposes of configuring the transmit power of access points. However, in many situations, configuring access points based on local information is not sufficient for reaching good, and more importantly, stable configurations, as is discussed in [84]. Therefore, in our work, we explore coordinated approaches to configuring access points.

- *Centralized Thin Access Points:* In this architecture, a centralized controller (or switch) connects to all access points [53]. The access points do not configure themselves but observe the wireless environment and send reports to the central controller. The controller then decides the best configuration for each access point. The access points are thin because they are not intelligent enough to make decisions by themselves. The advantage of this architecture

is that it is cheaper to maintain because access-point costs are lower. As a result, equipment-replacement costs go down. Meru [19], Aruba [12], Extricom [17], and Trapeze Networks [22] are examples of companies that use such an architecture for wireless LAN management.

Some academic researchers have also pursued the idea of centralization. Echos [81] is an architecture that uses a centralized controller to manage the power and carrier-sense threshold of access points and clients. The goal is to minimize access-point interference. However, the proposed solution only addresses exposed-terminal problems and does not deal with other types of RF interference. MiFi [37] uses an alternate approach with a centralized controller for time multiplexing wireless access points such that that no two access points that interfere with each other are activated at the same time. Although this technique is well grounded, it only works well in sparse deployments, where the delays due to time-multiplexing are small. In dense deployments, this approach does not scale well.

"Third generation" architectures have also been proposed that combine thin and fat access-point capabilities. Xirrus [27] provides a single integrated device that incorporates multiple access points into a single wireless LAN array. All the access points use a common MAC layer and therefore only consist of three components: the baseband, RF circuitry, and power amplifier. This single device can then be used to provide complete coverage over the desired space. This significantly decreases management overheads of the system. However, the solution does not provide fault tolerance and, in particular, represents a single point of failure, where a fault in the device can disrupt service for everybody.

## 3.3 Detecting Interference

In this section, we briefly discuss some techniques in the literature for detecting RF interference. The detection process occurs during the network monitoring phase.

- *Analytical Methods:* Analytical methods [29, 37, 54] capture interference by assuming wireless signal attenuation occurs uniformly as a function of distance [50]. As a result, the coverage of a wireless network is assumed to be circular and static. In such scenarios, the interference range of a transmitter is typically twice the transmission range, and a node experiences interference if it lies in the interference range of another transmitter. A vast amount of

literature on wireless networks assumes this model for interference. However, because this model does not represent reality accurately, many of the proposed protocols do not work well in practice. Alternatively, complex modeling techniques such as Stochastic Geometry [34] have also been proposed, but they fail to account for the dynamic nature of the wireless medium. Therefore, analytical methods, in general, do not capture interference accurately in practical settings.

- *Statistical Methods:* Statistical methods measure the performance of a wireless network and use a variety of metrics for *inferring* the existence of interference. Akella et al. [31] use packet-loss rate and values of signal-to-noise ratio (SNR) at the receiver to detect the existence of interference. Maniezzo et al. [62] estimate the impact of interference between two nodes based on the estimated Signal-to-Interference-Noise Ratio (SINR) at the receiver. Adya et al. [28] use packet-loss rate and round-trip time to detect performance problems, and for inferring the existence of interference.

  However, these techniques also perform poorly in isolating the existence, degree, and source of RF interference correctly. This is because there are too many environmental variables that can simultaneously affect the parameters being measured. This motivates the need to develop a controlled environment in which we can isolate the problem properly. Qiu et al. [73] adopt this approach by using a trace-driven simulation framework. They collect traces from the real environment and replay them in the simulator. The simulator acts as a controlled environment in which accurate root-cause analysis can be done. Parallel to the ideas in this thesis, Padhye et al. [71] discuss an approach of running pairwise experiments, in a controlled environment, between nodes to detect and quantify interference. However, the number of experiments they require is quite large and can take up to 28 hours for an entire deployment, making their approach infeasible for operation in a realtime environment. In our work, we show that it is possible to run similar experiments "on the fly" and with little performance overhead to participating nodes.

## 3.4 Parameter Configuration

In this section, we briefly cover some techniques that have been proposed in the literature for configuring the wireless LAN parameters we presented in Chapter 2. We focus particularly on channel assignment and power control, and only briefly list a few others.

1. *Channel Assignment:* Channel assignment has been studied extensively in the literature [51, 66, 75, 76, 77], especially in the context of cellular networks [57, 58, 78]. It is a well-known NP-hard problem. Intuitively, it seems that channel assignment strategies for cellular networks can also be applied to channel assignment in wireless LANs. However, this is not true for the following reasons. First, cellular-network deployments are usually planned symmetrically, where the coverage of a base station does not extend beyond a well-defined region of space. This allows the network planner to exploit symmetry in the deployment to apply relatively straightforward channel re-use techniques, where the same channel may be used anywhere outside a base station's coverage area. Second, due to the large number of cellular channels available, the granularity at which channel assignment is done in cellular networks is much finer than in wireless LANs, which have a limited number of orthogonal channels. Finally, the coverage area of a cellular-network base station is more uniform and less dynamic, making the resulting channel assignment more stable. Cellular base stations are mounted atop tall towers, where the impact on wireless signal propagation is much smaller than that seen indoors, where wireless LANs typically operate. These reasons make channel assignment strategies in cellular networks infeasible for wireless LANs. However, a number of heuristics and techniques have been proposed in the optimization literature that are suitable for this problem. Greedy heuristics are the simplest techniques that iteratively assign channels to access points using some selection criteria [40, 47, 56, 61, 69, 88]. Techniques such as local search, simulated annealing, and genetic algorithms (discussed in Chapter 4) can also be applied for channel assignment [41, 42, 60, 82, 89]. For instance, Mishra et al. [67] use a randomized search algorithm that incorporates client interference in the channel assignment process. In our work, we also use a particular optimization technique for performing optimal channel assignment.

2. *Power Control:* For the coverage-planning problem, power control has been shown to be NP-complete (it has been reduced to the well known 3-SAT problem). In terms of interference minimization, Sheth et al. [79] modify transmit power dynamically as part of the MAC protocol they propose to minimize interference between pairs of communicating nodes. They also propose an implementation of transmit-power control to minimize energy consumption, where the sender obtains feedback from the receiver on the strength of the received signal. The transmitter uses this information to decide the optimal power to use for transmitting all subsequent packets [33]. Akella et al. [31] also use a similar feedback mechanism for estimating the SNR of the signal at

the receiver. They use this for adjusting the transmission power of the access points. Daji et al. [72] compute optimum power levels off-line, for each destination node. The optimum power level can then be used after an RTS/CTS exchange with the destination, to prevent hidden terminal problems.

3. *Other Techniques:* Other tuning parameters that have also been used for improving wireless network performance are briefly mentioned further.

   Transmission-rate adaptation has been studied for improving performance for wireless clients. Auto Rate Fallback (ARF) [32] was the first algorithm publicly available for adaptive rate adjustment, based on observed packet-loss rates. Due to the limitations of ARF, an extension of this algorithm called Adaptive ARF has also been proposed that modifies the threshold used by ARF for deciding between rate changes [59]. Bicket et al. [39] propose an algorithm called *sampleRate* that samples different rates to find the best one, based on the achieved packet-delivery ratio.

   Phased antenna arrays using smart antennas and beam-forming technology have been used commercially for improving wireless LAN performance. Xirrus and Vivato [23] use this technology to maximize spatial diversity, which in turn improves network performance. Vivato uses narrow beams that follow clients, providing consistent coverage for them wherever they are situated.

   In conjunction with phased antenna arrays, Multiple-Input Multiple-Output (MIMO) systems are also being used for exploiting spatial diversity in indoor environments. MIMO systems prove useful in situations where non-line-of-sight (NLOS), multi-path, and scattering effects are dominant. Examples include MIMO-based access-point offerings from Linksys and D-Link [16, 26]. Spatial diversity has also been exploited with the help of multiple access points. Signals from a client are picked up by two or more access points, and the client is then serviced by the access point that receives the best signal [68]. These techniques are similar in spirit to MIMO systems that use multiple antennas to achieve the same objective.

## 3.5 Related IEEE Standards

We briefly cover some standards proposed recently for supporting management capabilities in 802.11x-compliant wireless technology.

- *IEEE 802.11k:* The IEEE 802.11k standard [5] describes mechanisms by which clients provide site reports to access points. These site reports con-

tain information such as the channel quality with respect to the client, and information on neighbouring access points and clients that this client can hear. Specific functionality that the 802.11k standard defines includes the collection of accurate RF channel information, hidden node information, and client statistics.

- *IEEE 802.11h:* The IEEE 802.11h standard [7] is meant primarily for use in Europe, and defines automatic mechanisms for performing transmit-power control and channel assignment. The goal is to mitigate interference that wireless LANs cause on radar and satellite systems. The standard is being proposed solely for IEEE 802.11a, operating in the 5Ghz band. Although defined separately, 802.11k and 802.11h use a variety of techniques that are common to both standards.

- *IEEE CAPWAP Standard:* The IEEE CAPWAP standard [1] defines a common framework to allow inter-operability between different vendors' access points. In existing self-managing systems, vendors typically use automatic discovery mechanisms or tunneling protocols for this purpose. CAPWAP tries to streamline this process for wireless access points that need to communicate with a variety of different wireless LAN vendor solutions.

- *IEEE 802.11v:* The IEEE 802.11v standard [9] is the latest standard that proposes full-featured network management support for IEEE 802.11x networks. 802.11v complements the 802.11k standard by providing the necessary support at the infrastructure end. This allows ease of deployment and management and also provides support for other services such as load-balancing between wireless access points. The standard also envisions building a common platform to allow multiple vendors' access points to inter-operate. To achieve this goal, it plans to use mechanisms proposed in the IEEE CAPWAP standard.

We have briefly outlined both past and present work related to self-management of wireless LANs. Traditional approaches are not well-suited to dynamically changing wireless environments whereas existing approaches fall short in providing the desired functionality. Moreover, accurate approaches to detecting RF interference are also lacking and need to be considered in the context of self-management, in conjunction with appropriate mechanisms for parameter configuration. In the next chapter, we discuss a simpler problem related to self-management, to gain insights into the general problem. Chapter 5 then discusses the self-management architecture that we propose in this thesis.

# Chapter 4

# Static Optimization

Recall from the previous chapter that there are two approaches to managing wireless LANs, i.e., static optimization and dynamic optimization. In this chapter, we study a problem that attempts to meet the objectives for static optimization. We explore this simpler problem to gain some intuition on the form for the more complex dynamic optimization solution, presented in Chapter 5. The description of this problem is outlined next.

## 4.1 Problem Description

The goal for static optimization is to output a set of access-point locations and configurations that maximize a given objective function. The problem that we explore here only looks at computing optimal access-point configurations. In an ideal world, we envision that a wireless infrastructure installer can place a number of APs roughly equally spaced in a geographical area, without necessarily doing a site survey, and then simply walk away. The APs should manage their channel and power allocation to maximize coverage. If there are persistent dead spots, then the system should automatically detect them and tell the installer where to add an AP. Conversely, if some AP's power level has been set to zero, the installer could be asked to remove that AP. Moreover, the system should adapt its parameters dynamically in response to changing workloads and environmental conditions. Based on these observations, we believe that the access point configuration problem is of greater importance and therefore explore it here.

For this problem, we select the channel and transmit power of the access points as tuning parameters to maximize the objective function. The objective function

we consider attempts to maximize coverage of the deployment region, and not necessarily capacity. Because we are addressing the static optimization problem, we assume that in the final solution we generate, the coverage areas of the APs will remain static. However, as we discuss later, the model we develop for this problem is general enough to cater to dynamic optimization as well. This model is presented further.

## 4.2 Utility Model

We first state the general problem more formally. This allows us to state our assumptions crisply and delineate the scope of our solution.

We assume that the wireless network infrastructure is meant to cover a given geographical area, $A$. At a point with coordinates $(x, y)$ in $A$, we define a utility function $U(x, y)$. This utility function is proportional to the transmission rate that can be obtained by a client at that point, and is zero at points where there is no coverage. The transmission rate at a given point, in turn, depends on the load from other clients at the closest AP, and the signal strengths and the degree of interference among multiple APs that cover that point. For instance, if there is a single AP serving that location, with a high signal strength and no other clients, then the transmission rate is high. On the other hand, a point that is far away from all the APs or is too near multiple APs would have a low transmission rate.

We model the degree of interference, for locations in overlapping AP coverage areas, as being proportional to the sum of the traffic loads in each such AP. We can summarize this discussion as follows. Let $AP(x, y)$ be the AP with the highest signal strength at $(x, y)$, where $AP(x, y) = \phi$ if no AP has a signal strength higher than the signal floor at that point. Then, a mobile at $(x, y)$ will associate with $AP(x, y)$. We define the set $Interfere(x, y)$ as the set of APs and clients that have a signal strength greater than the signal floor at $(x, y)$ and are not $AP(x, y)$. Then:

$$U(x, y) \propto \frac{1}{load(AP(x, y))} \tag{4.1}$$

$$U(x, y) \propto signal\ strength(AP(x, y)) \tag{4.2}$$

$$U(x, y) \propto \frac{1}{\sum_{i \in Interfere(x,y)} load(i)} \tag{4.3}$$

We would like to choose channel assignments and power levels so as to maximize the overall utility, subject to constraints on the number of available channels,

the number of available power levels, the traffic load at each AP, and the (x,y) placements of the access points.[1]

Formally, the objective function we wish to maximize is:

$$Maximize \int_{(x,y)\in A} U(x,y)dxdy \tag{4.4}$$

Given that we need to assign channels and transmit power levels to each of the APs, the problem is therefore a joint channel assignment and power control (CAPC) optimization problem. Our long-term goal is to solve the general CAPC problem in realistic settings. For this solution, we solve a simpler version of the CAPC problem by choosing a simpler form of the utility function. Harder versions of the CAPC problem correspond to more complex utility functions.

## 4.3 Geometric Model

Our model attempts to maximize the objective function of Equation 4.4. We make the following simplifying assumptions:

- *APs in 2-D plane*: We assume APs are located in a two-dimensional plane.
- *Omni-directional Antennas*: We assume all APs are equipped with omni-directional antennas.
- *Physical Interference Model*: We adopt the interference model used in [50] for modeling signal path loss in our model. Using this model and the assumptions listed above, our coverage areas can be represented as circular disks in a 2-dimensional plane.
- *Centralized Solution*: We assume that a single central coordinator determines the optimal solution. Given that most real deployments have a centralized controller for authentication, authorization, and accounting (AAA), this assumption is not particularly strong.
- *Cooperation*: We assume that the APs are cooperative.

[1]Though the discussion so far has assumed static coverage areas and traffic loads, it can be trivially extended with a time parameter to allow us to compute the overall utility at each point in time.

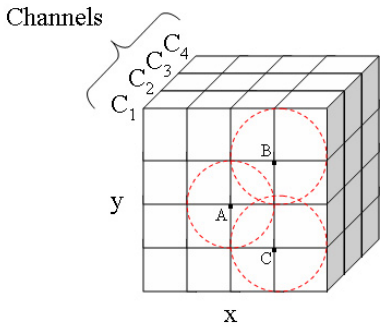


Figure 4.1: Model lattice that represents (x,y) location coordinates in the vertical plane and channels in the horizontal plane.

- *Access Point Interference*: We only consider AP-AP interference for our model.
- *Symmetric Channels*: We assume channels are symmetric.
- *Identical APs*: We assume all APs have identical discrete power levels and choices of channels.

Based on these assumptions, we can represent our model geometrically as shown in Figure 4.1. The vertical plane on the lattice embeds the locations of each of the access points, which are fixed. The channels are represented in the third dimension as the horizontal plane on the lattice. The transmit powers and corresponding coverage areas of each of the APs are represented by dashed circles around the APs A, B, and C. Larger transmit powers correspond to larger circles on the 2-D plane. Therefore, using this model, overlapping circles indicate interference between neighbouring APs.

With this model and the stated assumptions, we translate the CAPC optimization problem into two simple geometric problems:

1. *Packing variable-size disks on a rectangle (PACK-RECT)*: Here, we model the utility function as follows:
   - $U(x, y) = 0$ if there is no coverage at $(x, y)$, i.e., $AP(x, y) = \phi$,
   - $U(x, y) = 1$ if $Interfere(x, y) = \phi$,
   - $U(x, y) = -\infty$ if $Interfere(x, y) \neq \phi$.

Here, we study power control only (i.e., a single channel), ignoring the effects of client load and assuming uniform signal strength in a coverage area. Given the utility function above, it is easy to see that no coverage overlap between adjacent APs is allowed. The problem thus reduces to a packing problem for *fixed-location* variable-sized disks on a 2-dimensional plane where the objective is to maximize the coverage of the plane. This problem is computationally hard because there are $p^n$ possible solutions where $n$ is the number of access points and $p$ is the number of discrete power levels for each AP. For even small deployments with 10 APs and considering only 5 possible power levels, there are more than 9 million possible solutions.

2. *Packing variable-size disks on a stack of rectangles (PACK-ST)*: This problem extends the previous one for multiple channels, keeping the utility function the same. In this case, each rectangle represents a separate channel. Due to the additional degree of freedom, we now need to solve the channel assignment problem as well. It has been shown in [51] that the channel assignment problem for wireless LANs is NP-hard.[2]

Optimal solutions to even these simplified problems are computationally hard. Therefore, in an effort to build practical solutions, we devise heuristics to approximate the optimal solution. We then compare their performance relative to the optimal solution, computed using exhaustive search.

## 4.4 Heuristics

Many techniques can be used for generating assignments of power levels and channels to access points. We first present a classification of techniques, and then propose heuristics for solving the problem.

- *Greedy Algorithms:* Greedy algorithms are the simplest of those techniques that iteratively select access points and assign them a channel and power level [40, 47, 56, 61, 69, 88]. The order in which access points are selected can be based on a number of factors (e.g., their location, degree of interference, etc.). Some of the heuristics that we propose also use a greedy approach to maximizing the objective function.

[2]The authors reduce the channel assignment problem to a maximum $k$-colourable graph problem on an unweighted graph, where $k$ is the number of channels.

- *Search Optimization Techniques:* There are many search techniques in the optimization literature that can be applied to the problem of power level and channel assignment. We present a few illustrative examples here.

    1. *Randomized Search:* Randomized search probes a random sequence of configurations in the search space to look for good solutions. The strength of this approach is that it provides better coverage of the search space. However, the algorithm is not guaranteed to converge quickly to the optimal solution, or converge at all for that matter. Therefore, an algorithm that uses randomized search needs to be run multiple times to approximate a good solution to the problem. We propose one such approach for static optimization.
    2. *Local Search:* In local search, an initial solution is generated either randomly, or through some other method, and is then improved in subsequent steps through neighbourhood search. Multiple passes can be made over access points, and their configurations altered. Guided Local Search (GLS) [82] is an extension of local search that avoids the possibility of getting stuck in local minimas.
    3. *Tabu Search:* Tabu search is a form of local search that allows moves that are non-improving, to escape local minimas [42]. The algorithm moves to the best solution in the neighborhood while simultaneously maintaining a record of previously visited *bad* solutions in a Tabu-list. In each iteration of Tabu search, all bad solutions visited in that iteration are placed in the Tabu-list. The length of the Tabu-list is given by some value $k$, which is specified as a tuning parameter of the search algorithm. If there isn't enough space in the list to add bad solutions that are visited in the current iteration, some old solutions are removed. This also prevents the technique from getting stuck in cycles during the search. In Chapter 5, we discuss how we extend this technique for channel assignment.
    4. *Simulated Annealing:* Simulated annealing also allows non-improving moves, to escape from local minimas [89]. At each step, a determination of the feasibility of a configuration is made. If the configuration improves the solution, it is accepted. Otherwise, the solution is accepted based on a temperature parameter that decides how large of a move is allowed in the search space. The value of the temperature parameter is gradually decreased based on a cooling schedule. Therefore, simulated annealing requires two inputs: a mechanism to probe the search space and a cooling schedule. The search is terminated if either a sufficiently good solution has been found, or the temperature becomes low enough.

5. *Genetic Algorithms:* Genetic algorithms apply concepts from Darwin's theory of evolution, to evolve a population of solutions and improve them through the means of selection, mating, and mutation [60]. Initially, a random set of solutions is generated (referred to as individuals of the population) which are then ranked based on the concept of Pareto dominance. In the selection process, individuals are selected for breeding based on their goodness (i.e., fitness value). In the mating process, crossover is performed between previously selected configurations using a chosen crossover point. Finally, mutation is also performed to allow non-improving moves in the search space, with a small probability.

Despite the variety of optimization techniques available to solve the channel assignment and power control problem, we find that simple heuristics approximate the optimal solution quite well. Further, we present four simple heuristic power-control algorithms for the PACK-RECT problem and two algorithms for joint channel assignment and power control (PACK-ST). All APs are initialized to the lowest power level (i.e., a transmit power of zero) when the algorithms begin execution.

### 4.4.1 Randomized Incremental Algorithm (RIA)

The idea behind this algorithm is to pick an AP at random and increase its power level, until either the maximum power is reached, or the AP begins to interfere with another AP. More formally, the algorithm first places all APs into an unordered *feasible* set. It then randomly picks an AP from the set and increases its power level by one step. If the transmit power of the AP cannot be increased any further or increasing its power causes interference, it is removed from the set, otherwise it is kept. The algorithm then selects another AP at random and repeats this process until eventually all APs have been removed from the set. This process is illustrated in Algorithm 1.

Due to randomization, a single run of this algorithm does not always yield a good solution. Therefore we run the algorithm many times and choose the run with the best performance. In the worst case, no APs interfere and the algorithm incrementally increases the power of each AP until all APs reach maximum transmit power. Therefore, the running time of RIA is bounded by $O(p * n)$, where $p$ and $n$ are the number of discrete power levels and access points respectively.

**Algorithm 1** Randomized Incremental Algorithm ($Tx$ = Transmit Power)

1: $f = \{ap_1, ap_2, ap_2, \ldots, ap_i\}$ /* feasible set of APs */
2: **while** true **do**
3: Randomly select an access point $ap_i$ from $f$.
4: **if** $ap_i's\ Tx \neq max.\ Tx$ **then**
5: Increase $ap_i$'s power by one.
6: **if** ( $\exists(x, y)$ s.t. $U(x, y) = -\infty$ ) **then**
7: Decrease $ap_i$'s power by one and allocate it this power level.
8: Remove $ap_i$ from $f$.
9: **end if**
10: **else**
11: Remove $ap_i$ from $f$ and allocate it its current power level.
12: **end if**
13: **if** $f = \emptyset$ **then**
14: Terminate.
15: **end if**
16: **end while**

### 4.4.2 Generalized Greedy Power-Allocation Algorithm

Algorithm 2 illustrates the general steps followed by the other three power-control algorithms. The generalized algorithm greedily increases the transmit power of an AP, chosen in turn from an ordered feasible set, to the maximum possible power, given AP interference and power constraints.

#### Distance-based Ordering Algorithm (DOA)

The DOA algorithm orders the feasible set by decreasing distance of an AP from the center of mass (or centroid) defined by: ( $\sum_i(x_i/n)$, $\sum_i(y_i/n)$ ), where ($x_i$, $y_i$) are the coordinates of $AP_i$ and $n$ is the number of APs. The DOA algorithm is based on the idea that APs farthest from the center of mass are likely to experience less interference and thus should be the first to have their power level increased greedily. An illustration of the computed *centroid* is shown in Figure 4.2. Using an efficient sorting algorithm such as quick-sort for set ordering, the worst-case running time of DOA is bounded by $O(n \log n)$, where $n$ is the number of access points.

**Algorithm 2** Generalized Greedy Power Allocation Algorithm ($Tx$ = Transmit Power)

1: $f = \{ap_1, ap_2, ap_2, \ldots, ap_i\}$ /* feasible set of APs */
2: Order set based upon power control algorithm used.
3: **for** $i = 1 \ldots \mid f \mid$ **do**
4: Expand coverage of $ap_i$ until ($\exists(x, y)$ s.t. $U(x, y) = -\infty$) or $ap_i$'s $Tx = max.$ $Tx$.
5: **if** ( $\exists(x, y)$ s.t. $U(x, y) = -\infty$ ) **then**
6: Decrease $ap_i$'s power by one and allocate it this power level.
7: **end if**
8: **end for**

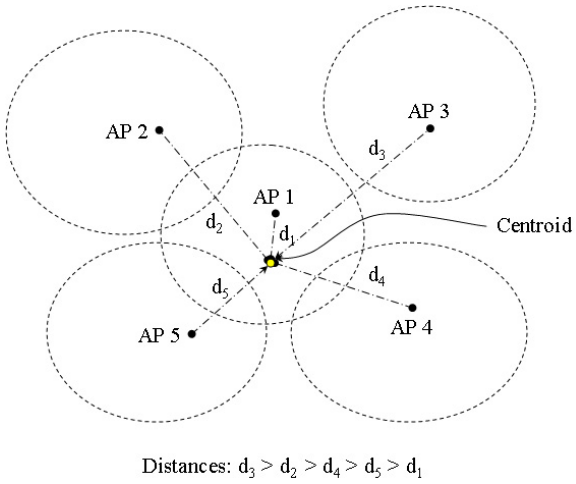


Figure 4.2: The center point indicates the center of mass (or *centroid*) of the five APs.

### Voronoi-based Ordering Algorithm (VOA)

The VOA algorithm makes use of the Voronoi diagram, a geometrical model commonly used for a variety of problems in computational geometry [45]. VOA partitions the deployment region into $n$ cells, where $n$ is the number of access points. Each access points resides in exactly one of these cells. The cells are called *Voronoi regions*, where in any given region, every point in that region is closer to the access point of that region than any other access point in any other region. This partitioning is also referred to as a Voronoi tessellation of the deployment region and is illustrated in Figure 4.3.

There are many ways to construct a Voronoi diagram. The most efficient tech-

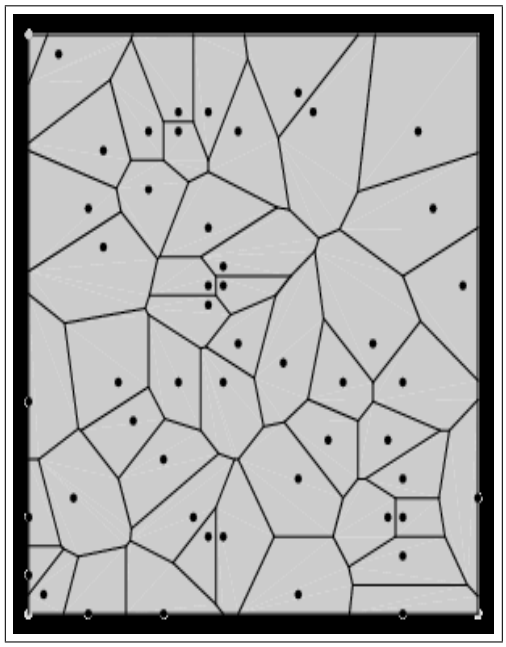

Figure 4.3: An example Voronoi diagram of a set of access point locations.

nique is based on *Fortune's Algorithm*, which can compute the diagram in $O(n \log n)$ time, where $n$ is the number of access points [45]. Using this diagram, the greedy algorithm then orders access points based on the size of the Voronoi region with which they are associated. Access points with larger Voronoi regions are placed first in the list and so on. The intuition behind the ordering is that access points that have larger Voronoi regions are less restricted in terms of their ability to maximize coverage. Therefore, these access points are given priority for coverage maximization. Not surprisingly, access points that reside at the corners of the deployment space have on average larger coverage areas because of fewer neighbouring access points that restrict their expansion. In this way, DOA also approximates VOA because it sorts APs in order of their distance from the centroid, giving access points at the boundary larger coverage than those closer to the center.

### Interference-based Ordering Algorithm (IOA)

The IOA algorithm uses the degree of interference at each AP to order the feasible set. IOA first instructs all APs to transmit at maximum power. Using this configuration, IOA calculates a degree of interference at each AP as the amount of overlap

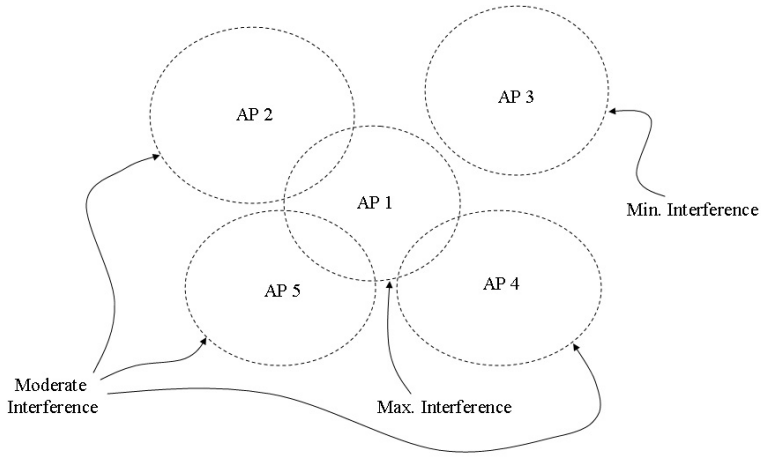


Figure 4.4: The classification of APs based on interference performed by IOA. In this figure, all APs transmit at the maximum transmit power. Since $AP3$ does not interfere with other APs, it is the first AP whose transmit power is increased.

that an AP experiences in its coverage area with neighbouring APs. APs are then placed in the feasible set in increasing order of interference. The ordering thus gives priority to APs with low interference, ensuring that the aggregate interference is minimized while the coverage area is also maximized. An illustration of how IOA might classify APs based on the degree of interference is shown in Figure 4.4. The worst-case running time of IOA is also $O(n \log n)$.

### 4.4.3 Multi-Channel Algorithms

So far, we have assumed all APs share a single channel. We now study the multi-channel case. We assume $n$ access points and $m$ channels, where $m$ is typically much smaller than $n$. Therefore, the objective here is to devise algorithms that construct good channel re-use configurations. The following issues need to be addressed:

1. Which channel does each AP use?
2. What power-level should each AP use?

We assume separability of the two issues and first allocate channels to APs before allocating power levels. Power-level assignment is done using the RIA, DOA, VOA and IOA algorithms presented earlier. Therefore, we concentrate on the first issue.

The general solution to channel assignment is known to be NP-hard [51], and therefore, we discuss a heuristic algorithm that approximates the optimal solution. We also describe a naive random channel-allocation algorithm that is used as a "straw man" for comparison with our proposed algorithm.

### Two-Phase Channel Assignment

This channel assignment algorithm operates in two phases. In the first phase, it generates a set of APs that are either ordered based on the metric used for power control (i.e., for DOA/VOA/IOA), or are in a random order (i.e., for RIA). In the second phase, the algorithm begins by removing the first AP from the set and assigning it to the first channel. Then, using this AP as a reference point, the algorithm removes $\frac{n}{m} - 1$ APs farthest in distance from this reference AP and also adds them to the first channel. Assuming that APs are uniformly distributed within an area, an assumption that is likely to be valid for most practical scenarios, this not only assigns the same channel to APs that interfere minimally with each other, but is also likely to divide the load evenly across the channels. This process is repeated for each available channel in turn.

This algorithm has several variants. For example, instead of sequentially allocating $\frac{n}{m}$ APs to each channel, we can assign just two APs to each channel at each iteration of the algorithm and repeat this process in a round-robin fashion across all the channels until all APs have been assigned. We found that this variant performs almost exactly the same as the algorithm discussed above, thus we only present results for the first algorithm.

### Random Channel Assignment

For random channel assignment, we begin with an unordered set of APs. We proceed sequentially through the set and, uniformly at random, assign a channel to each AP. Thus, although there is no limit to the number of APs that can be assigned to a channel, on average, we expect to assign approximately $\frac{n}{m}$ APs to each channel. Nevertheless, since this algorithm does not consider interference or distance between APs in its assignment process, we expect it to perform poorly in comparison with our two-phase channel-assignment algorithm.

## 4.5 Evaluation

We now evaluate our algorithms for power control and channel assignment. We first compare our power control algorithms with each other and the optimal configuration. We then compare the two-phase channel assignment algorithm with random channel assignment.

### 4.5.1 Evaluation Methodology

We have written a compact simulator in Java to compare our algorithms. We emulate a random deployment scenario by randomly placing APs on a two-dimensional grid of fixed size (i.e. 500x500). APs are placed such that no two APs occupy the same location, but they may be within interference range of one another (even if they transmit at minimum power). This may cause some APs to be effectively blocked out during the configuration-generation process. We discuss the implications of this problem in later sections. The inputs to the simulation include:

- The number of deployed APs,
- The number of available channels,
- The number of transmit powers to choose from,
- The maximum transmit power of all APs,
- The power control algorithm being used, and
- The channel assignment algorithm being used.

Coverage areas of APs are represented as uniform circular areas on the grid. As indicated in Section 4.3, since we are solving the PACK-RECT and PACK-ST problems, the objective here is to maximize coverage of the grid while keeping the interference zero. The maximum transmit power of an AP is computed by taking the maximum coverage of the AP as a fraction of the total grid area (which is 30% for our simulations). This prevents any single AP from using up the entire grid, since, due to power limitations, this is unlikely to happen in practice. For most of our results, we have also fixed the number of transmit-power levels to 15. The numbers of transmit-power levels are quite diverse across different vendors [3, 13] and we find that 15 power levels covers the space of most typical radios. For our

multi-channel results, we also fixed the number of available channels to three. This represents the most widely-deployed 802.11b systems.[3] For transmission rate, we adopt a conservative approach where APs always transmit at 1 Mbps uniformly across their entire coverage area. We defer the study of dynamic rate-adaptation schemes based on path loss to future work.

To compute the utility, we used Monte-Carlo sampling. That is, we randomly select some sample points within the coverage areas, to estimate the cumulative coverage of all the APs. We could have used an exact method for computing coverage areas by first computing the coverage of each AP and then subtracting from it any overlapping zones. However, computing overlaps exactly is a mathematically daunting task. Monte-Carlo sampling provides a quick, simple, and fairly accurate approximation of the coverage of the grid. In our computations, we used different sample sizes and compared the relative error in the computed result. When comparing sample sizes of 50,000 and 250,000 for example, we found that the error in the computed utility was less than $\pm 1.6\%$, which is acceptable. The area of the grid was 250,000.

### 4.5.2 Results

We repeated our simulation 30 times in order to minimize statistical variation in our results. For every run, we generate a set of randomized AP locations to prevent placement biases that could affect any of our algorithms. For RIA, in each run, we also ran the algorithm 10 times on the same set of AP locations, and took the maximum of the computed utilities. An example output of our simulator is shown in Figure 4.5. We now discuss our results further.

Figure 4.6 presents the mean coverage area for each of our power-control algorithms and the optimal solution (using only a single channel and five power levels). For these low-density deployments, we see that the VOA, IOA, and DOA algorithms perform quite close to the optimal solution, which was computed using exhaustive search of all possible configurations. For high-density deployments, we are not able to provide a quantitative comparison, since the search space for the optimal solution increases exponentially fast with increasing AP densities. In general, since we need to assign both power levels and channels to APs, the size of the search space effectively becomes $P^N * C^N$, where P = number of transmit powers, C = number of channels, and N is the number of APs. For P = 5, C = 3, and N =

[3]Our multi-channel results only present the benefits of using multiple channels and their effect on power control. We defer a study of the effect of varying the number of available channels on the performance of the algorithms to future work.

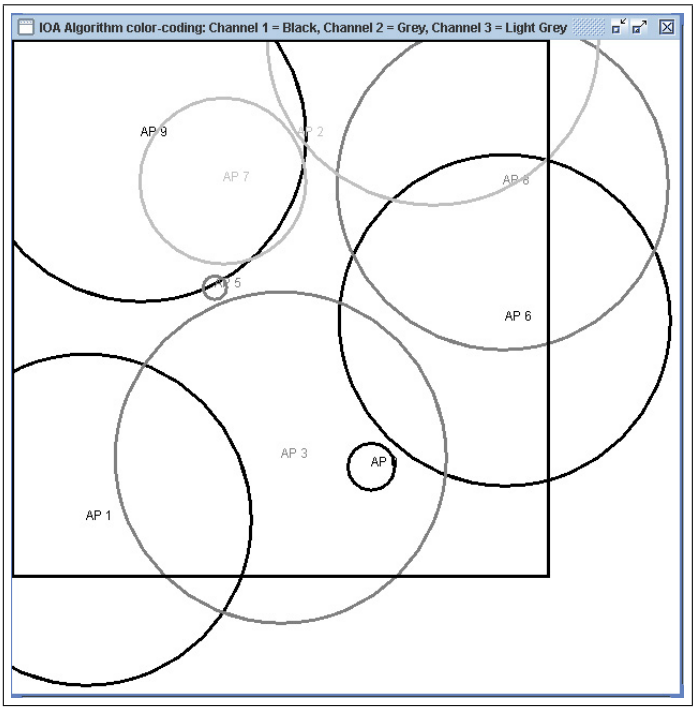


Figure 4.5: The square represents the area on which the APs are placed. Each shade represents a separate channel.

10, we have $\approx 576$ billion possible configurations! However, as we discuss later, we do obtain evidence from our multi-channel results of near-optimal behavior even for high density deployments. These results show that our algorithms cover almost 100% of the grid, clearly indicating that our algorithms are near optimal.

Figure 4.7 presents a comparison of VOA, IOA, DOA, and RIA. We observe that RIA always performs worse than VOA, IOA and DOA, especially in high-density environments. This is probably because RIA does not bias power-level increases towards APs that are in less congested areas, causing it to perform poorly in high-density environments where opportunities for interference increase. Another reason may be that RIA increments power levels at *all* APs. In contrast, VOA, DOA, and IOA maximize their coverage greedily at each step of the algorithm. Consequently, with these algorithms, at high AP densities, APs that are close to an AP transmitting at high power may effectively be blocked from communicating at all. Although this may be thought of as negative behavior, it is actually beneficial since it serves to reduce the overall interference in the system. Figure 4.8 illustrates this by showing that the number of APs can be reduced by as much as 60% with

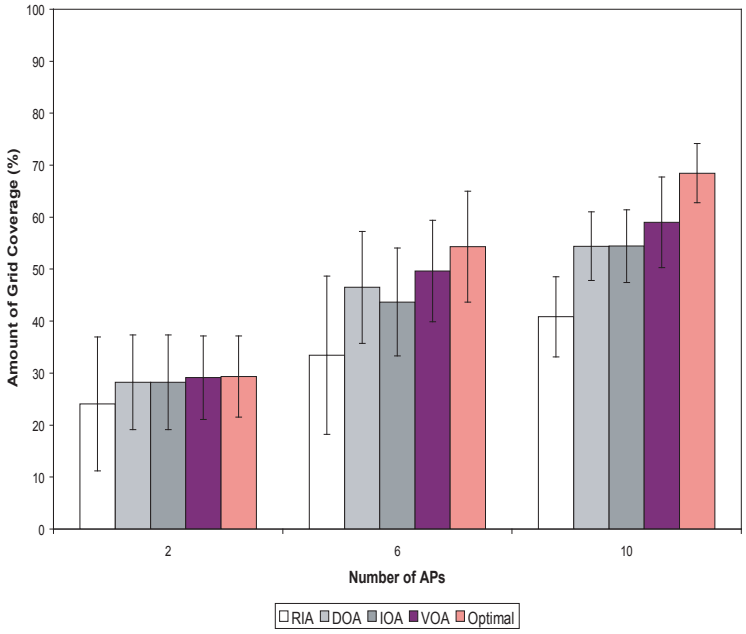


Figure 4.6: Histogram of performance of power-control algorithms against optimal configuration using single channel and five power levels.

the IOA algorithm while RIA uses almost all of the available APs. This result also gives us an intuition as to the optimal number of APs that would be required to cover a grid of a given size.

We now turn our attention to the multi-channel case. Figure 4.9 presents results for our power-control algorithms using random channel assignment and Figure 4.10 shows the performance of the power control algorithms using our two-phase channel-assignment algorithm. These figures demonstrate the benefits of using multiple channels over a single channel. For dense environments, we see an almost 37% increase in the cumulative coverage area as compared to a single channel. Moreover, the percentage grid coverage of the VOA/IOA/DOA algorithms increases to approximately 93%, from about 70% for the single-channel case. Since the interference region is effectively partitioned among the three channels, the coverage area increases. In addition, we also observe that the gap between RIA and VOA/IOA/DOA has also decreased. Since the interference per channel has been reduced, RIAs blindness to interference does not hurt it as much.

When we compare the performance of the random channel-assignment algorithm

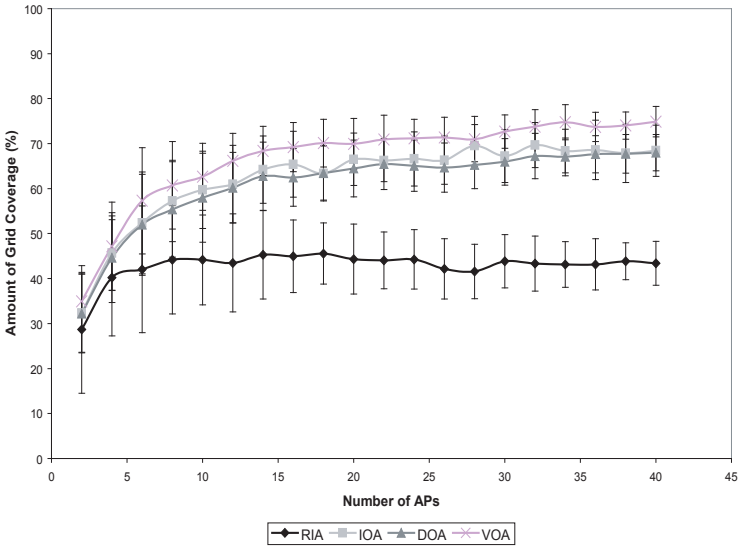


Figure 4.7: Performance of power-control algorithms using a single channel and 15 power-levels

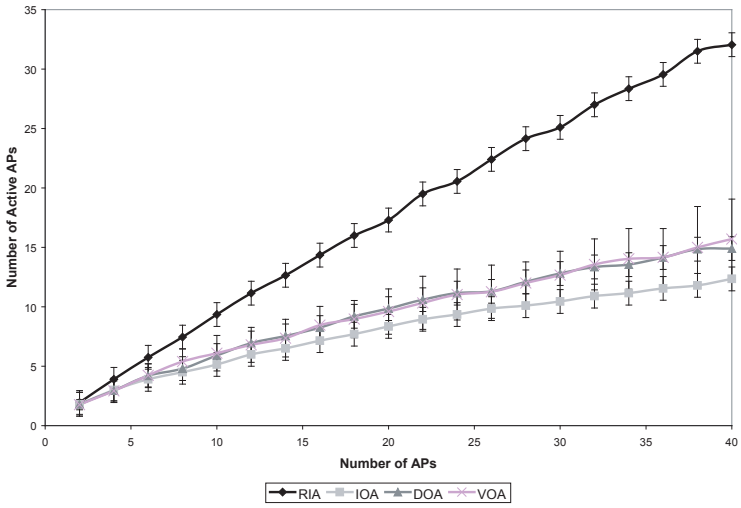


Figure 4.8: Comparison of power-control algorithms based on the number of APs used with a single channel and 15 power-levels

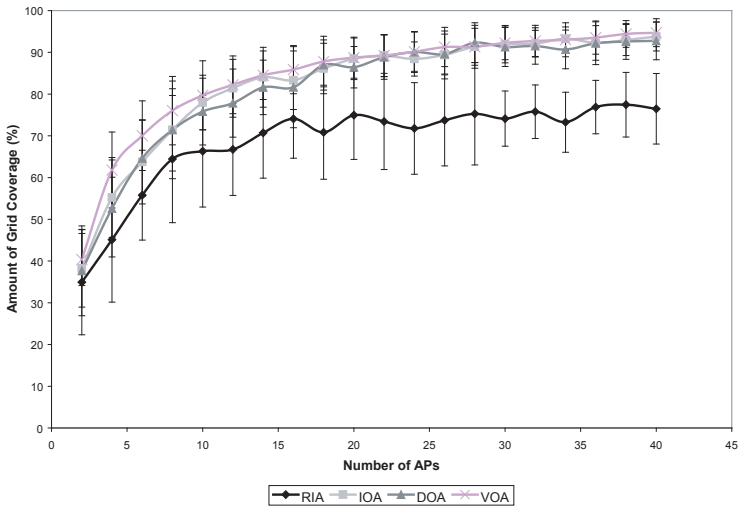


Figure 4.9: Performance of power-control algorithms using random channel assignment for 3 channels and 15 power levels

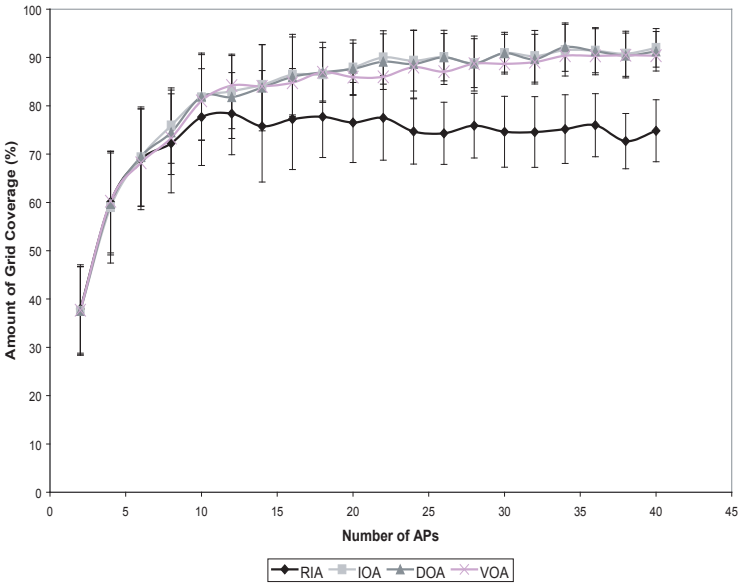


Figure 4.10: Performance of power-control algorithms using two-phase channel assignment for 3 channels and 15 power levels

and our two-phase algorithm, we see that both algorithms perform roughly similarly across the board. At low AP densities, the two-phase algorithm does perform better because it places APs farthest away from each other (i.e., with least interference), onto a common channel. This gain is maximized for the first channel, but decreases for later channels, so that overall, the gain from this strategy is not too great, especially at higher AP densities. Intuitively, having multiple channels simply partitions the problem space into three. At high enough density, this reduction in problem size does not reduce the overall interference level, and a random channel assignment works just as well as a more complex channel allocation strategy. In such situations, performance depends more on the power-allocation algorithm than the channel-allocation algorithm, as illustrated by the better performance of VOA, IOA, and DOA over RIA for both schemes. This leads us to advocate the (far simpler) random channel-allocation strategy as a pragmatic solution in real-world deployments.

## 4.6 Discussion

Deploying a wireless infrastructure network requires us to balance several conflicting requirements. For this problem, we have taken the first step towards an ideal world, where an installer can set up a network quickly and simply walk away. We propose a successive-refinement approach to deployment. We argue that this approach is better suited for real-world wireless deployments. We also present a mathematical and geometric model that crisply describes the solution space and identifies the characteristics of an optimal configuration. We design and evaluate heuristics that yield near-optimal configurations. We find that the choice of heuristics for transmit-power control of access points is a crucial factor in determining the quality of the solution. We also find that a random channel-assignment approach is effective for assigning channels as the deployment density increases.

We hasten to point out that our results are preliminary because they do not capture several aspects of the real-world problem. For example, our interference model is very simplistic and does not capture irregularity in the coverage of the APs. This affects IOA since it relies on the underlying geometric model. Also, our utility function assigns the same utility to each point on the covered grid. In reality, this utility is dependent upon many factors: uplink/downlink channel conditions, transmission rate, traffic load, etc. Finally, although our algorithms do perform well in simulation, we still need to test them on a real testbed.

Nevertheless, our results do allow us to develop some intuition about the form

of the final solution.

- Although finding the optimal configuration even for our simple problem is hard, to our surprise, we find that simple heuristics approach this optimal configuration closely.
- In general, careful power control appears to be more important than careful channel allocation. This result should hold even in more general conditions.
- Surprisingly, VOA, IOA, and DOA all perform almost identically even though IOA more accurately models the degree of interference and was thus expected to be superior to VOA/DOA. Incidently, VOA/DOA are also insensitive to the underlying geometric model, making them suitable for non-circular coverage areas as well.
- We also observe that naive random channel assignment is able to perform similarly to interference-aware two-phase channel assignment.

Based on these observations, we conjecture the following: For the coverage maximization problem, in dense deployments, an effective configuration strategy would be to first perform a random assignment of channels to APs, and then to use a greedy power-allocation algorithm that is the same as or similar in spirit to VOA or DOA. Since channel assignment is performed at random, coordination is only needed for power allocation.

In this chapter, we presented an approach to meet the objectives for static optimization, to managing wireless LANs. We extend some of these ideas to the self-managed architecture we present in the next chapter, to meet the objectives for dynamic optimization.

# Chapter 5

# Dynamic Optimization

We now present the design of an architecture for dynamic optimization. Recall, our design goals arise from the need to build a solution that is incrementally deployable and backwards-compatible. The techniques we use build on the utility model presented in the previous chapter. We use a conflict graph [66] for representing utility, and extend it to capture additional information. We also present an experimental approach to computing utility that is amenable to real-world deployments. Finally, we also present techniques for optimizing utility. The first part of the chapter describes the high-level concepts that comprise the architecture, while the second part discusses its implementation details.

## 5.1 Towards an Infrastructure-based Solution

Wireless LANs are comprised of wireless access points (or infrastructure) that are connected to a wired network, and wireless clients (or end-points) that connect to these access points. An infrastructure-based solution for these wireless LANs only requires modifications to the access points and/or connected infrastructure. Infrastructure-based solutions are easier to deploy than solutions requiring client modifications, as the network is usually managed by a single administrative entity (or provider). On the other hand, client devices are owned and operated by individual users. Take the example of an airport, where a self-managed wireless network is deployed. It is unreasonable to expect passengers/users to modify or upgrade their devices just to use the airport's network. If Internet access is not absolutely required, users may simply choose not to use the network, which is undesirable both

for the network provider and end users. Therefore, for a solution to be practically realizable, it must be backwards-compatible.

In contrast, many existing self-managing systems require modifying clients to allow them to collect information on the RF environment, on behalf of the network. Aside from lacking features for backwards-compatibility, other concerns with this approach are as follows.

- *Performance:* Clients operating realtime applications can suffer from poor performance if they are required to go off-line to collect information on the RF environment. This problem is already evident in existing IEEE 802.11x standards that mandate periodic scanning of channels by clients, to find better access points to associate with. Scanning overheads have been shown to be over 250ms [74], causing unacceptable delays for interactive applications such as voice telephony.

- *Energy Consumption:* Clients collecting observations of the RF environment also have to generate site reports for them. These reports are subsequently forwarded to one or more neighbouring access points. Both these operations (i.e., report generation and transmission) consume energy and can reduce the battery life of client devices. This also raises issues of incentive compatibility for clients that do not utilize the network as much. We are not aware of any existing system that addresses these concerns.

- *Network View:* A client's view of the network is highly dependent on its spatial location in the environment. Ideally, to obtain the best view of the network, clients would be uniformly distributed over the entire deployment space. In reality, this is not true, and in campus-wide deployments, clients are typically grouped in clusters [55]. An example of a scenario that suffers from poor client distribution is depicted in Figure 5.1. Because clients do not reside between APs 1 and 2, they are not able to identify inter-AP interference that is being experienced by the APs.

- *Site-Report Correctness:* Site-report correctness refers to the accuracy of the information provided by the clients. Because access points utilize this information for configuration purposes, clients can use this fact to game the system. As there is no way to verify the correctness of the received reports, clients can lie by indicating regions of high interference, causing the infrastructure to provision more resources (e.g. better channels) to their regions. For this reason, it is important to reduce the dependency of the infrastructure on the clients.

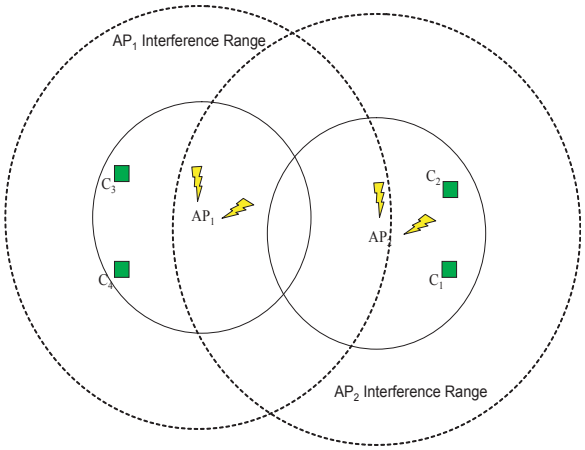


Figure 5.1: Clients cannot detect the presence of inter-access point interference due to poor distribution. Access points cannot detect it either because they are in each others interference range.

We also identify an important fact regarding client reports. Clients can only report observations of neighbouring clients and access points if they lie within the transmission range of these devices. If clients are in their interference range instead, they cannot correctly identify the interfering devices. As we discussed earlier, the interference range is typically twice the transmission range. Therefore, there is a high probability of having clients in the interference range.

## 5.2 Architecture Overview

Having identified the need for an infrastructure-based solution, we now present an overview of our architecture, shown in Figure 5.2. The architecture is based on the centralized thin-AP paradigm. The central controller assumes the responsibility of the master that coordinates the channels and power levels of all access points. The choices of channels and power levels are decided based on utility information, which is computed using statistics collected by the access points. We first illustrate the high-level procedure that is followed in computing these channels and power levels, and then discuss each step in greater detail. We point out that the core functionality of the architecture manifests itself in the optimizer module of the central controller. Therefore, all of the steps (except the first two) that we outline next describe the sequence of steps followed in the execution of the optimizer. These steps are also

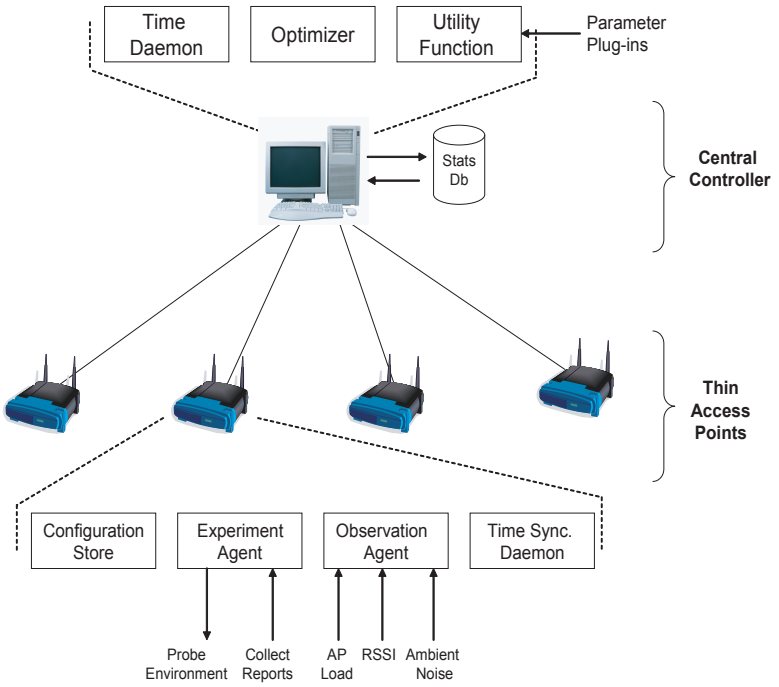


Figure 5.2: High-level illustration of self-management architecture

illustrated in Figure 5.3.

1. *Utility Parameters:* At startup, the central controller obtains input on the performance parameters that are to be captured by the system. This information is obtained by means of an API that is exposed to the network administrator. The administrator specifies each parameter along with a preference value (or weight) that she associates with it. The preference indicates the importance of the parameter in the optimization process.

2. *Utility Function:* The parameters and their weights are used to generate a utility function that identifies the performance objectives of the system. These objectives may differ between deployments and the utility function allows us to adapt to different operating environments.

3. *Conflict Graph:* A conflict graph (or CG), as described in [66], is used to model interference between access points. Each access point is represented by a vertex in the CG. An edge is added from a vertex A to a vertex B if the access point corresponding to vertex A causes interference at the access point

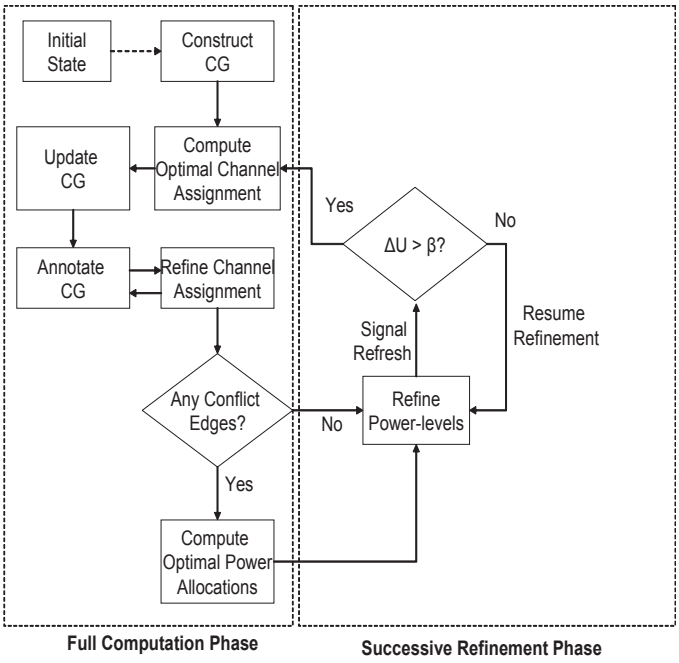


Figure 5.3: State diagram of steps followed in configuration optimization procedure.

corresponding to vertex B. The weight of the edge represents the amount of interference that A causes at B. The edges of the CG only incorporate direct access-point interference, and not interference that is experienced by their clients. The conflict graph serves as a building block for many features supported by our architecture, one of which is channel assignment.

4. *Optimal Channel Assignment:* Using the conflict graph from the previous step, channel assignment performs a vertex colouring of the CG. Each colour corresponds to a channel and therefore each vertex is assigned a particular colour in the graph. The algorithm discussed in Section 5.3.4 is used for coloring the CG. The CG is then updated to reflect the channels assigned to the vertices. Therefore, if the vertices connected by an edge in the original CG are coloured differently, the weight of the edge is zero. This process is repeated for all access points to generate the updated CG.

5. *Annotated Conflict Graph:* The updated conflict graph generated in the previous step is then augmented further, to generate the Annotated Conflict Graph, or ACG. The updated CG is annotated by adding clients to it one by one. When adding a client, edges to/from access points and the client may be

added that represent either client association to an access point or interference between them. Note, clients do not have edges to all access points, and vice versa. Association edges have weights that capture the positive utility being obtained by the client from the access point. Interference edges have weights that represent the disutility due to interference between them. When a client is inserted into the conflict graph, a refinement of the access point's channel is also done to minimize the interference experienced by the client from neighbouring access points and vice versa. We elaborate on this idea further in Section 5.3.4.

6. *Optimal Power-Level Assignment:* The ACG is then used to perform power control for the access points. Due to the nature of the ACG, the power-control problem is essentially a maximization of the sum of the edge weights in the ACG. This implicitly maximizes the utility function. The algorithm we use for this problem is described in Section 5.3.4. Note, however, that performing power control causes the underlying ACG to change. This is because it changes the amount of interference that is being experienced by access points and clients. Therefore, the ACG may need to be recomputed (or refined) each time the power level of an access point is changed.

7. *Incremental Power Refinement:* The final step in the configuration process is incremental power refinement. At this point, the system has entered the successive-refinement phase where it remains, until optimal configurations need to be recomputed. Incremental power refinement is a continuous process where an access point is selected at random and its power level is either increased or decreased. If the change improves the overall utility, it is kept, otherwise it is discarded. The goal of this step is to allow the system to gradually adjust to the changing environment. It also keeps the underlying ACG up-to-date. However, there may be circumstances where a large change in utility is measured by the system, or the previously computed optimal configuration becomes stale (e.g., due to client migrations, access point failures, etc.). This could require re-computation of power and channel assignments. In our system, this refresh is triggered periodically (e.g., every 30 mins) where the system measures the difference between the current utility and that which was computed the last time optimal channel and power-level assignment was performed. If the change is greater than a threshold ($\beta$), the system discards the current ACG and repeats the optimization process, as shown in Figure 5.3. Otherwise, it returns to incremental power refinement.

We now discuss the conceptual framework on which the architecture is based.

## 5.3 Architecture Components

### 5.3.1 Utility Function

We build on the utility framework described in Section 4.2, to construct more sophisticated utility functions that capture multiple objectives. The strength of the approach is the ability to specify more than one objective using a single optimization function. The utility parameters and weights specified by the network administrator are used to generate different types of objective functions. There may be multiple definitions for a parameter and these need to be supplied for the utility function as well. For example, access point load could be defined simply as the number of clients associated to the access point, or by more sophisticated load definitions that take into account transmission bit-rates of clients and the measured traffic demand. The utility function can cater to any of these definitions. To our knowledge, no other approach provides this flexibility in specifying objectives using a single unified framework.

The utility (or objective) function consists of parameters that reflect not only the network's performance, but also that of wireless clients and applications. For example, an FTP application is interested in maximizing throughput, whereas a streaming application is more interested in minimizing end-to-end delay, given an acceptable amount of bandwidth. This information can be supplied by clients when they associate to the network. The system can then support individual user needs as well.

We now examine the structure of the utility function in more detail. The general utility function for the network can be expressed as follows:

$$U_{total} = \Sigma_k U_k \tag{5.1}$$

where,

$$U_k = w_1 p_1 + w_2 p_2 + ...w_n p_n \tag{5.2}$$

Equation 5.1 represents the total utility of the system, and Equation 5.2 represents the utility obtained by each access point. Note, this definition of access point utility assumes that the utility is represented as a weighted sum of the utility obtained from each of the parameters. This is just for illustration purposes, as the utility can be represented using any other utility function definition as well. In Equation 5.2, $p_1$ to $p_n$ represent the performance parameters that are to be captured, and $w_1$ to $w_n$ are their respective normalized weights. Below, we discuss some parameters for this utility function. Note, the definitions we supply for these

parameters are by no means restrictive and other definitions may also be used. These parameters serve as examples of how to specify objectives for the utility function. We present two parameters for modeling client performance, one that models network performance, and one that is applicable to both.

1. *End-User Utility*: For simplicity, we assume that a single application is being run at each client. Our approach can easily be extended to multi-application scenarios as well. We first consider end users that are running non-realtime applications, where the goal is to maximize client throughput. In order to provide structure to the problem, consider a two-client scenario. Both clients are associated with the same access point. Client $c_1$ is communicating at a rate $r_1$, while client $c_2$ is communicating at $r_2$. We assume that each client continuously sends at a fixed rate and does not implement dynamic rate adaptation (e.g. Auto-Rate Fallback). In this scenario, the throughput of each client is constrained by the wireless rate of the slower client. Because the slower client occupies the medium for a longer period of time when transmitting than does the faster one, the effective throughput of both is the same. This performance anomaly for IEEE 802.11 networks was first pointed out in [52]. Therefore, the throughput can be mathematically expressed as follows. Suppose each client is transmitting $b$ bits of information to the access point, assuming symmetric channel conditions. Then,

$$b = r_1 t_1$$
$$b = r_2 t_2$$

where, $t_1$ is the time taken by client $c_1$ to transfer $b$ bits and $t_2$ is the time taken by client $c_2$. Therefore, the total time taken to transfer $2b$ bits of information is $t_1 + t_2$. So, the average throughput (i.e. bits/time) is:

$$\begin{aligned} R &= \frac{2b}{t_1 + t_2} \\ &= \frac{2b}{b(\frac{1}{r_1} + \frac{1}{r_2})} \end{aligned}$$

Assuming utility is proportional to the average throughput, the simplified equation for average throughput can be stated as:

$$R = \frac{1}{(\frac{1}{r_1} + \frac{1}{r_2})}$$

Furthermore, we generalize the above equation for more than two clients,

$$R = \frac{1}{\sum \frac{1}{r_i}} \tag{5.3}$$

where, $r_i$ is the wireless rate of client $i$. Because 802.11 supports short-term fairness, $R$ represents the average throughput of each client associated to the access point. Because non-realtime clients are interested in maximizing throughput, their utility is exactly the value specified using Equation 5.3. We now consider how the utility of clients that are running realtime applications can be specified using Equation 5.3. Suppose there are $n$ non-realtime clients and $m$ realtime clients associated to the same access point. Suppose each realtime client requires a minimum throughput of $\bar{r}$ to satisfy application demands. Then, to function properly, the following condition must hold:

$$\frac{1}{\sum \frac{1}{r_i}} \geq \bar{r}$$

If all realtime clients have identical throughput requirements, then each clients utility (in terms of the average throughput) can be expressed as follows:

$$R_{realtime} = \begin{array}{rcl} \frac{1}{\sum \frac{1}{r_i}} & , & if \frac{1}{\sum \frac{1}{r_i}} \geq \bar{r} \\ 0 & , & Otherwise \end{array}$$

Because each of these throughput equations is simply a measure of the utility obtained by each of the clients, their aggregate utility can be expressed as follows:

$$\begin{aligned} U_{clients} &= U_{nonrealtime} + U_{realtime} \\ &= \frac{n}{\sum \frac{1}{r_i}} + m * (R_{realtime}) \end{aligned} \tag{5.4}$$

2. *Network Load*: Network load is a parameter that caters to network performance and is defined as the total load experienced by all access points in the network. Access-point load is defined as the sum of the individual loads contributed by its clients. This can be expressed as:

$$load_i = \Sigma_k l_{ik}$$

where $load_i$ is the aggregate load experienced at access point $i$, and $l_{ik}$ is the load contributed by client $k$. The goal of specifying this parameter in the utility function is to support load balancing across access points. Note, this parameter only captures the effects of network load and it must be used in conjunction with a tuning parameter to actually provide load-balancing functionality. For a given total load in the system, we model the load-balancing problem using the following utility function:

$$U_{load} = \Sigma_i U_{load_i} \tag{5.5}$$

$$U_{load_i} = -(|load_i - \frac{\Sigma_k load_k}{N}|) \tag{5.6}$$

Equation 5.6 assigns a negative utility to the difference between an AP's load and the average load, which is what it should have in a perfectly balanced system.

3. *Interference*: Interference affects the performance of both the network and the clients. Therefore, its disutility incorporates components for each, i.e., inter-access-point interference and access-point-client interference. Note, client-client interference may also exist between clients. However, such interference constitutes a very small fraction of the total networks interference because clients typically download more data than they upload. In addition, because clients usually do not continuously send data, unlike access points, they are less likely to experience interference from each other. Thus, we ignore this case in the utility function we discuss here. In Section 5.4.1, we discuss an approach to actually computing this utility. The total interference in the network can be represented as the sum of inter-access-point interferences, and access-point-client interferences. This is expressed as follows:

$$Int_{total} = \Sigma_{i=1}^{N}\Sigma_{j=1}^{N} Int_{ij} + \Sigma_{i=1}^{N}\Sigma_{v=1}^{K} Int_{iv} + \Sigma_{i=1}^{N}\Sigma_{v=1}^{K} Int_{vi}$$

where, $Int_{ij}$ is the interference that access point $i$ causes on access point $j$, $Int_{iv}$ is the interference access point $i$ causes on client $v$, and $Int_{vi}$ is the interference client $v$ causes on access point $i$. $N$ and $K$ are the total number of access points and clients, respectively. As discussed in Chapter 4, interference also depends on the load of the interfering entity. Adding this to the previous equation gives us:

$$\begin{aligned} U_{interference} = -(&\Sigma_{i=1}^{N}\Sigma_{j=1}^{N} Int_{ij} load_j \\ &+ \Sigma_{i=1}^{N}\Sigma_{v=1}^{K} Int_{iv} load_v \\ &+ \Sigma_{i=1}^{N}\Sigma_{v=1}^{K} Int_{vi} load_i) \end{aligned} \tag{5.7}$$

where $load_i$ is the load at the access point or client. Because interference impacts the performance of both the access points and clients negatively, it is represented as a disutility. Though this discussion presents the total disutility from interference, we later show how each component of this total utility is incorporated separately into the conflict graphs for optimization.

The utility functions and parameters we described also have a time component associated with them, illustrating the utility at different points in time. However, for simplicity, in this and later discussions, we have ignored this dimension of the utility function.

### 5.3.2 Conflict Graph

The conflict graph represents the interference (or conflicts) experienced at each access point due to one or more neighbouring access points. The conflict graph we define for our system only contains edges between access points if they interfere directly with each other. In contrast, Mishra et al. [66] define edges as both a function of inter-access-point interference and access-point-client interference. Moreover, they use undirected edges to model interference as symmetric. These techniques have limitations, as we describe later. We formalize the definition of the conflict graph we use as follows. Consider a graph $G = (V, E)$, where $V$ is the set of vertices and $E$ the set of edges. Then, $G$ is a conflict graph if the following relations hold.

1. $V = \{ap_1, ap_2, ap_3, \ldots, ap_n\}$, where $ap_i$ is access point $i$.
2. $E = \{(v, u) | f(ap_v, ap_u) < 0\}$
3. $f(i, j) = -(Int_{ij} load_i)$,
   where, $Int_{ij}$ is the interference caused by access point $i$ on access point $j$ and $load_i$ is the load on access point $i$.

A conflict graph is therefore a *directed graph* where each edge represents interference caused by an access point at which the edge originates on an access point at which the edge terminates. Due to the nature of the wireless channel, interference between access points may not be symmetric. Therefore, an undirected edge cannot accurately model the degree of interference between two entities. An example conflict graph is depicted in Figure 5.4. The goal in constructing the conflict graph is to use it during channel assignment to reduce the amount of conflict that occurs between access points.

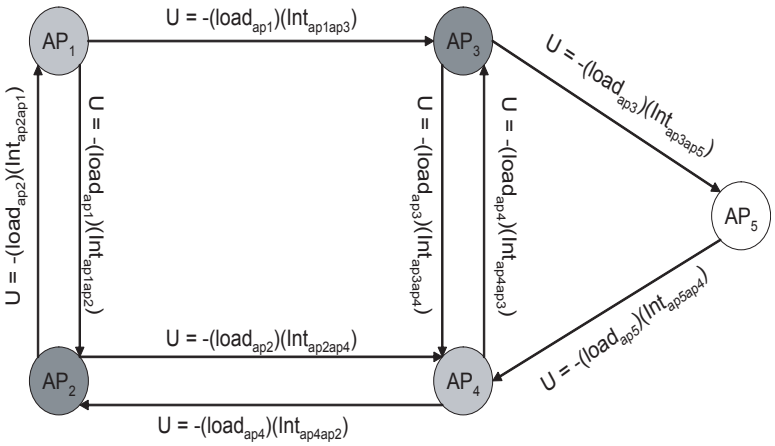


Figure 5.4: An example conflict graph, annotated with interference/conflict information.

The channel-assignment problem can be modeled as a vertex-colouring problem where each colour corresponds to a different channel. The goal is to assign colours to vertices such that no two vertices that have an edge between them are assigned the same colour. In this situation, the edge weight between the vertices is assigned a value of zero. An example colouring is illustrated in Figure 5.4, where adjacent vertices are assigned different shades. Because $AP_1$ and $AP_4$ are not adjacent to each other, they can be assigned the same colour. We also present the idea of a *k-colourable* graph. A graph is k-colourable if we can colour it perfectly (i.e., with no adjacent vertices having a common colour) using just $k$ colours. Therefore, if we have $k$ channels and we can colour the graph perfectly, no access point will have any conflicts with any other access point. Unfortunately, the graph-colouring problem has been shown to be NP-hard [51]. However, there are a variety of heuristics (discussed in Chapter 4) that can be used to approximate an optimal colouring in polynomial time. We discuss one such heuristic that we use for channel assignment in Section 5.3.4.

A closer examination reveals an additional difficulty. The number of colours (or channels) for our problem is also limited and is often restricted to three for the most common wireless LAN devices (e.g. IEEE 802.11b). Therefore, even if there were an algorithm that could colour a graph optimally in polynomial time, it may not be possible to do so if the graph is not *k-colourable*. These difficulties point to the need for additional mechanisms that can reduce the number conflicts in the graph. Power control provides this mechanism, as we describe later.

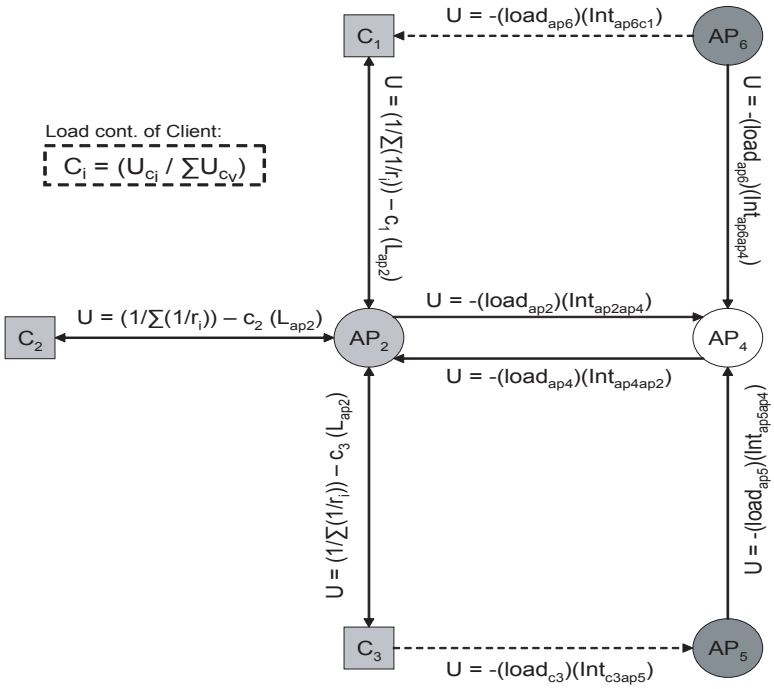


Figure 5.5: An example of an annotated conflict graph that contains client vertices and edges between clients and access points.

### 5.3.3 Annotated Conflict Graph

The annotated conflict graph augments the conflict graph with additional information to represent clients. Clients are added to the graph to reflect their utility in the optimization process. Each client is represented by a vertex in the conflict graph, and edges are inserted between clients and access points. There are two types of edges that can exist between a client and an access point. If a client is associated to an access point, an *association edge* is added between them. We assume that association edges are undirected. If a client interferes with an access point, or an access point interferes with a client, an *interference edge* is added between them whose direction represents the source and destination of the interference. An illustration of the ACG is shown in Figure 5.4.

The weights of the interference edges are derived using techniques similar to those used for the basic conflict graph. The weights of the association edges correspond to the utility that clients receive from their access points. For clients running non-realtime applications, their utility can be modeled using Equation 5.3. In addi-

tion, we may also want to capture the effects of network load in our representation. Using the load definition given by Equation 5.6, the unbalanced load ($L_i$) that an access point $i$ carries is:

$$L_i = -(|load_i - \frac{\Sigma_k load_k}{N}|)$$

Each client associated to the access point contributes a certain amount to this imbalance. This contribution is proportional to the positive utility that a client obtains from the access point. This is expressed as:

$$L_{c_v} = (L_i * \frac{U_{c_v}}{\Sigma_j U_{c_j}})$$

Where, $L_{c_v}$ represents the contribution of client $v$ to the imbalance. For overloaded access points, this negative value is added to the edge between the access point and the client. Therefore, if we were to migrate a client to a neighbouring access point, the load reduction on the current access point would be precisely what this equation captures. We add this load factor to the association edge between the client and the access point because we want to minimize weights on association edges that correspond to highly loaded access points. The load balancing algorithm can then identify these as low-utility access points, and subsequently migrate their clients to neighbouring access points for better load balancing across the network. The tradeoff between client throughput maximization and load balancing is well known [38]. Our utility function captures this tradeoff using weights that are assigned to each parameter. In the current representation, we have only considered client throughput, network load, and interference. However, it is relatively straightforward to add any other objective that we may also want to maximize. For instance, end-user objectives can be incorporated by further annotating the association edges.

Explicitly representing clients proves to be a better model for capturing utility and leads to better configurations of the network. Consider if we did not explicitly represent clients, and instead modeled their conflicts as edges between their associated access points. An example is shown in Figure 5.6. In this case, as a result of client $c_5$, the conflict graph contains an edge from each access point to all other access points. Clearly, this graph cannot be 3-coloured. However, if the ACG were used for this purpose, it would be possible to perform a 3-colouring of the graph. In the ACG, the clients would be assigned the same colour as their associated access point. Therefore, explicit client representation in the conflict graph is critical to obtaining good configurations.

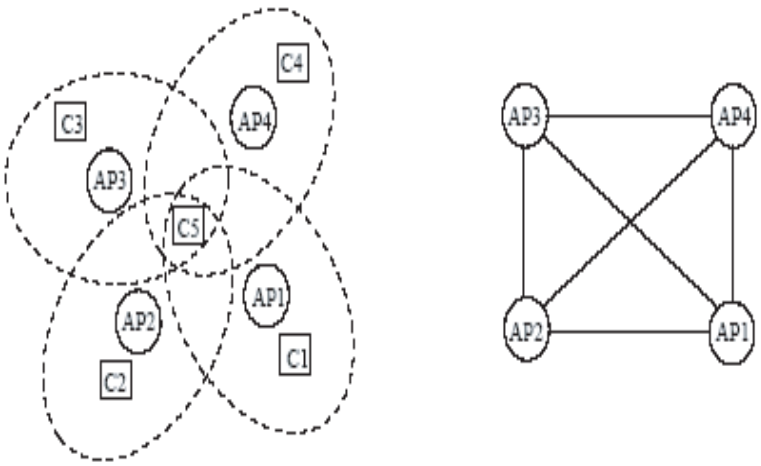


Figure 5.6: The left picture represents a scenario where client $C_5$ is being interfered with by neighbouring access points. The right picture shows the resulting non-client conflict graph (from [67]).

The channel refinement step we described in Section 5.2 is performed during the construction of the ACG. Once the ACG is constructed, all channel assignments are complete. The ACG is then used to perform power control for each access point. Construction of the conflict graph and annotated conflict graph is based on actual conflict (or interference) information detected in the environment. During the optimization process, the initial construction of the conflict graph is done with access points transmitting at maximum power. We refer to this graph as the Maximum Power Single Channel (MPSC) graph. The MPSC graph represents the worst-case number of conflicts that can exist between all access points in the deployment. This graph is useful in the optimization step, as we describe later.

Conflicts are computed by conducting a series of experimental tests that determine whether access points and clients interfere with each other. An experiment is defined as a *controlled* test that is performed to identify a particular scenario of RF interference. The controlled environment is provided by means of a separate signalling plane that we describe later. Experiments not only capture conflicts that can arise as a result of clients/access points residing within transmission range of the interfering source, but also those that reside within interference range of the source. The experiments provide information for annotating conflict edges between access points, as well as edges between access points and clients (i.e. in the ACG). Details of these tests are discussed in Section 5.4.1. The annotated conflict

graph also requires other information that is specified in the utility function (e.g., AP/Client load). This additional information is captured through passive collection of network statistics. Details on how this is done are discussed in Section 5.4.1.

### 5.3.4 Utility Optimization

Utility optimization is achieved by means of two parameters: channel assignment, and power control. As was discussed in Chapter 4, we assume separability between the parameters and optimize each independently. In our architecture, channel assignment is performed before power control. This approach makes our system more stable as changes in channels can effect service for many users. We elaborate on this idea in the next section. We first discuss our approach to channel assignment, and then discuss the details of power control.

#### Channel Assignment

To re-iterate, the goal of channel assignment is to reduce the number of conflicts between interfering nodes in the conflict graph. Therefore, channel assignment attempts to allocate orthogonal channels to nodes that have an edge between them. Once this is done, re-assignment of channels to access points should be done sparingly since this disrupts service for clients associated to the access point. Furthermore, our channel-assignment optimization step only incorporates access-point conflicts, not client conflicts. This is motivated by the need to provide stability in the assignment of channels. By also adding client conflicts to the optimization step, we substantially increase the likelihood of channels being reassigned during each run of the optimization algorithm. This fact motivates a channel-assignment procedure that occurs in two steps. In the first step, channels are assigned optimally using only the conflict graph presented in Section 5.3.2. In the second step, channels assignments are refined during construction of the annotated conflict graph.

Recall, the conflict graph that we construct is based on the access points transmitting at maximum power (or MPSC). Because access points locations are fixed and maximum conflict information is considered, inter-access point interference is not likely to increase, unless a new AP is installed in the network. This approach provides stability to channel assignment, because power alterations cannot worsen the number of conflicts seen by the channel-assignment process. Although this approach may be too conservative, performing power control after channel assignment yields a solution that is both stable and also minimizes the number of conflicts. The algorithm we use for optimal channel assignment is called one-point optimization

**Algorithm 3** One-Point Optimization

```
1: A = {a_1, a_2, ..., a_i} /* set of access points */
2: Let each access point in A be randomly assigned a channel
3: while true do
4:    cnc, nnc = ComputeTotalConflicts(A)
5:    bstColour = a_0.ch /* Assign any colour as the best colour */
6:    bstAccessPoint = a_0 /* Assign any access point as the best access point */
7:    for i = 1 ... | A | do
8:       newColour = FindBestColour(a_i)
9:       oldColour = a_i.ch
10:      a_i.ch = newColour
11:      if nnc > ComputeTotalConflicts(A) then
12:         bstColour = newColour
13:         bstAccessPoint = a_i
14:         nnc = ComputeTotalConflicts(A)
15:      end if
16:      a_i.ch = oldColour
17:   end for
18:   if nnc == cnc then
19:      Terminate.
20:   else
21:      bstAccessPoint.ch = bstColour
22:   end if
23: end while
```

(OPO). The algorithm is based on Tabu search optimization, discussed in Chapter 4. The OPO algorithm is shown in Algorithm 3.

First, the algorithm assigns a channel randomly to each access point and computes the current total number of conflicts (*cnc*) in the system. It then performs iterative refinement of the assigned channels. In each iteration, the algorithm takes each access point ($a_i$) in turn and computes the gain in utility, in terms of reducing the total number of access-point conflicts, that can be made by switching the access point to a different channel. It computes the utility gain for the access point on all channels (using the $FindBestColour(a_i)$ method). After finding the channel (*newColour*) that yields the greatest gain for $a_i$, it checks whether changing $a_i$ to *newColour* yields an improvement in utility that is larger than the best utility gain seen in the iteration so far (*nnc*). If so, ($a_i$, *newColour*) is considered the best improvement seen so far. Because the algorithm performs this operation across all

access points, it selects the access point and channel (one-point) that yields the largest gain in overall utility. The rest of the assignments that were examined in this step are placed in a Tabu-list as configurations that do not improve the utility beyond that of the configuration that was selected in this step. This prevents the algorithm from following bad paths in the search space. If a configuration present in the Tabu list is seen in the future, the algorithm does not consider it when choosing between configurations in a given step. Tabu list functionality is implemented as part of the *FindBestColour* method, and is not explicitly presented in Algorithm 3. This iterative process repeats until we reach a configuration where any further one-point alterations do not yield a gain in utility (i.e. $nnc == cnc$). Because the solution of the algorithm may depend on the channels that were assigned initially to the access points, we perform multiple runs of the algorithm and choose the best solution (in terms of utility) among them. This increases the chances of reaching better optima and possibly the global optima.

In the second step of the channel-assignment process, channel refinement is performed to channels computed in the first step. We support this functionality because we also want to incorporate client conflicts *unobtrusively* in the channel assignment. What this means is that the refinement procedure improves the channel assignment of the access points based on client conflicts, while preventing the number of inter-access point conflicts from increasing. This is achieved by performing *local channel search* at the access points as clients are added to the conflict graph, during ACG construction. Every time a client is inserted into the conflict graph, the algorithm checks to see if the client has any conflict edges to other access points. If so, it searches locally for a channel that simultaneously reduces the total number of conflicts for all the clients that are associated to the access point, while at the same time not increasing the number of inter-access point conflicts. If such a channel is found, the access point switches to that channel. If not, the access point does not change its channel. This process repeats every time a client is added to the conflict graph to generate the ACG.

The two-step approach for channel assignment provides greater resilience to channel changes, as changing an AP's channel disrupts service for many users. Therefore, changes to channel assignment occur at larger timescales than changes to transmit power, which is described next.

### Power Control

We outline the following goals for power control.

**Algorithm 4** Optimal Power Control

```
1: A = {a_1, a_2, ..., a_i} /* set of access points */
2: while true do
3:    u = ComputeTotalUtility(A)
4:    θ = MaxConflictAP(A)
5:    Z = {z_i | neighbour(θ, z_i) = true}
6:    for i = 1 ... | Z | do
7:       AdjustWeight(θ, z_i)
8:    end for
9:    γ = MaxConflictEdgeAP(θ, Z)
10:   ReducePowerLevel(γ)
11:   if u > ComputeTotalUtility(A) then
12:      IncreasePowerLevel(γ)
13:      Terminate.
14:   end if
15: end while
```

- *Minimizing Interference:* Power control further reduces internal interference that could not be accomplished simply by performing channel assignment. It minimizes the disutility due to interference.
- *Improving System Utility:* Power control improves the system's overall utility by choosing configurations that also simultaneously improve the utility of other parameters in the utility function.
- *Balancing Network Load:* Power control also enables load balancing across the network. This is achieved by migrating clients between access points during the power-control process. We currently do not support this feature, but plan to explore it in future work.

In our architecture, power control is performed using information available in the ACG. It can be performed more frequently than channel assignment because alterations to access-point power have a smaller impact on clients. Recent work shows that power control can also be done very efficiently, and even on a per-packet basis [30]. However, we hasten to present two constraints that make the power control problem challenging. First, power control needs to ensure that clients do not lose service, because this defeats the purpose of our system. Secondly, every alteration to access-point power causes the underlying ACG to change. Therefore, we may need to re-compute (or refine) the ACG for every change in access-point power. Thus,

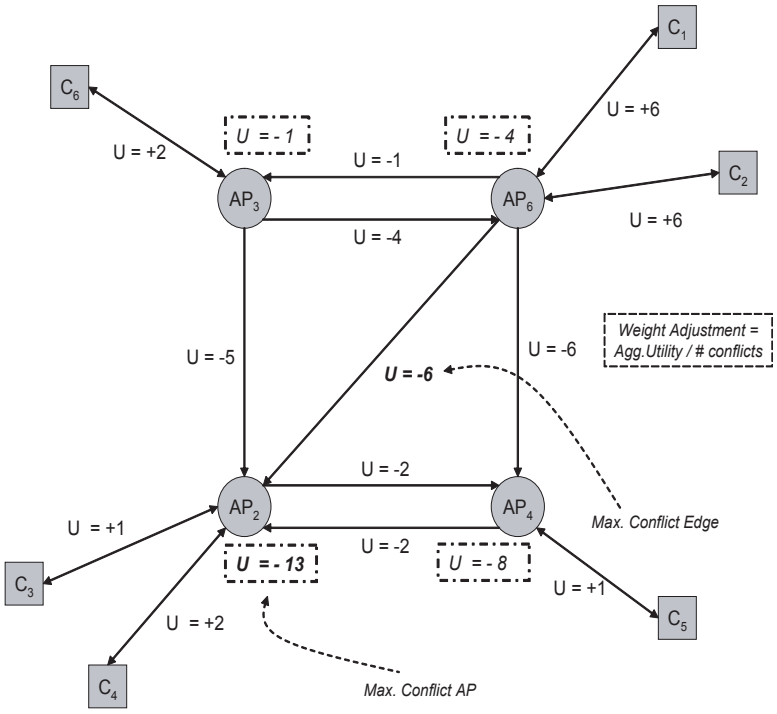


Figure 5.7: $AP_2$ is identified as the maximum conflict AP and the edge from $AP_6$ to $AP_2$ represents the maximum conflict edge, before edge re-weighting is done.

although power control can be performed more frequently than channel assignment, it must be performed conservatively to prevent these problems from exacerbating. The algorithm we propose proceeds in two steps. In the first step, we compute optimal power levels for all access points (full computation phase). In the second step, we refine access point powers to allow the system to adapt to changes in the environment (successive refinement phase).

The optimal power-control algorithm is illustrated in Algorithm 4. The algorithm starts with all access points transmitting at maximum power. Note, this is the configuration we have after channel assignment. In each iteration, it finds the access point that has the greatest number of conflicts ($MaxConflictAP(A)$). The greatest conflicts AP is the one whose sum total number of conflicts (measured as disutility) on all incoming edges from neighbouring APs is the largest. For this AP, the algorithm selects a neighbouring access point (in the conflict graph) from the set of all neighbouring access points ($Z$) that causes the greatest conflict on the maximum conflict AP ($MaxConflictEdgeAP(\theta, Z)$), after performing a *re-*

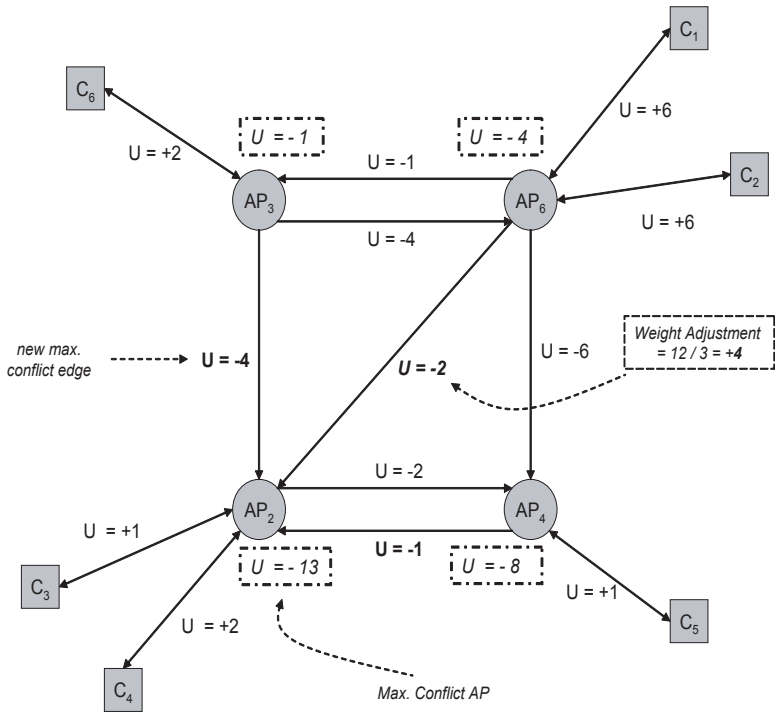


Figure 5.8: After edge re-weighting, $AP_3$ is identified as the AP that has the maximum conflict edge to $AP_2$. $AP_3$ and $AP_4$ edge weights to $AP_2$ only change slightly because these APs provide very little utility to their clients.

*weighting* step ($AdjustWeight(\theta, z_i)$). The re-weighting step does the following. For each AP that interferes with the maximum-conflict AP, a re-weighting of the incoming conflict edge is done that is proportional to the amount of utility that the interfering AP provides to its clients. APs that provide high utility to their clients have larger weights added to their outgoing conflict edges. Re-weighting allows the algorithm to prevent selecting high utility APs for the power reduction step. This is illustrated in Figures 5.7 and 5.8. We want to prevent power reduction of high utility APs as this is counter-productive to maximizing system utility. The neighbouring access point that is chosen after edge re-weighting ($\gamma$) is then told to reduce its power level by one ($ReducePowerLevel(\gamma)$). This process repeats in successive iterations until there is no further improvement that can be made and a decrease is detected in the overall utility, at which point the algorithm terminates (after reversing the last power alteration). The resulting configuration is assumed to be optimal. Recall, the goal of power control is to maximize the sum of the edge weights in the ACG. Our algorithm tries to achieve this by ensuring that APs with

high positive-utility edges are not affected as much as those with low positive-utility edges. This maximizes the utility of the system.

In the second step, once optimal power levels are assigned to access points, we switch to a successive refinement phase. In this phase, the system periodically selects an access point at random and asks it to either decrease or increase its power level. If the utility improves, the power change is applied. Otherwise, it is reversed. These incremental changes about the current power level prevent power control from causing large changes in utility and service disruption. Note, for both optimal power control and successive power refinement, the power of an AP is only decreased if it does not cause clients to loose connectivity in the process.

## 5.4 Implementation Details

We now discuss the implementation aspects of the architecture that we presented in Figure 5.2. We first discuss the software components that are required at the access points, and then provide details on the central controller. In further discussions, we use controller to refer to the central controller and TAP to refer to a thin access point.

### 5.4.1 Thin Access Point

There are four software components that comprise the design of the TAP. We discuss each of them in turn. We also illustrate the need for a signalling plane for performing experiments and outline how we can construct one for our system.

#### Configuration Store

TAPs maintain a small configuration store (e.g., in flash memory) for storing their configurations (channel and power level) that is periodically updated by the controller. The configuration store provides a means of updating TAP configurations without affecting the operation of the TAP. The TAP periodically reads the configuration store and makes the necessary changes to its configuration. To prevent race conditions in the reading/writing process, the TAP acquires a lock on the configuration store before reading from it. The frequency at which reads are performed to the configuration store can affect TAP performance. To avoid these expensive context switches, a daemon process can be run that triggers change notifications whenever an update is applied to the configuration store.

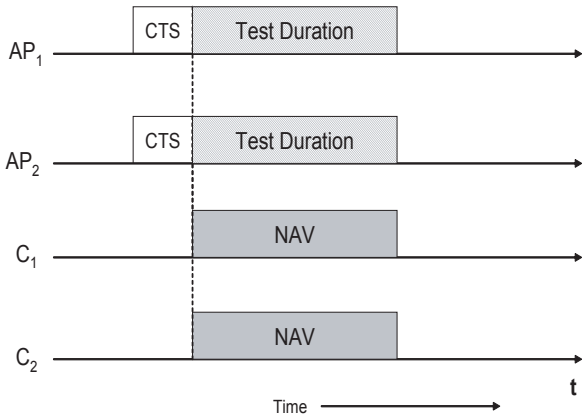


Figure 5.9: An example of how a controlled environment is setup for an experiment

### Signalling Plane

TAPs can utilize a separate signalling plane on which to perform experimental tests required for detecting conflicts (or interference). A separate signalling plane leads to a cleaner system design. The signalling plane can be provisioned by creating a pair of virtual channels on top of a single physical channel. This approach is common in cellular networks that operate multiple virtual channels over a few physical channels. These virtual channels are then time-multiplexed on the physical channels. In our case, one channel can be used for data, while the other can be used for signalling. The virtual channels are only relevant to the TAPs performing the experiments and are not visible to clients. Using these virtual channels, we can support our interference experiments without requiring any client modifications. Alternatively, if clients do have hardware or software-based support for multiple interfaces, we can also allot an independent channel to serve as the signalling plane in our architecture. Software-based support for multiple interfaces can be provided by means of software such as VirtualWiFi [43]. In this situation, the access point would then be provisioned with two radios, one for data and the other for signalling. This latter approach can provide improved performance as signalling traffic no longer needs to contend with data.

An important requirement for detecting interference is the need to set up a controlled environment for facilitating experiments. Unfortunately, the uncoordinated nature of wireless LANs makes providing such an environment difficult. In order to facilitate this requirement, we need to introduce mechanisms for nodes to back-off from transmitting when experiments are being conducted. For this purpose, we

introduce a short *back-off* phase before initiating an experiment. This involves the associated TAPs sending *unprovoked CTS* transmissions on the signalling plane to their clients. An unprovoked CTS is a CTS that is not preceded by an RTS. The CTS serves as a dummy packet used to set the NAVs of stations in range of the TAPs, as shown in Figure 5.9. The controller commands all TAPs whose clients may interfere with the experiment to send out an unprovoked CTS to their clients. The value in the NAV field of the CTS is set to the duration of the desired experiment (or experiments). However, due to their nature, some experiments may also involve clients sending feedback to the TAPs. Unfortunately, clients that have their NAV field set as a result of receiving the unprovoked CTS will not respond to any test packets during this period. Therefore, in such experiments, TAPs whose clients need to participate in the experiment can send an unprovoked CTS with a smaller NAV value. The value could be any value between 2*SIFS and the propagation delay of an RTS packet, as all test packets to clients are RTS packets. After sending the unprovoked CTS packet, the TAP waits a SIFS interval and then transmits the RTS test packet to the client. Because the client's NAV value will have expired by the time it completes reception of the test packet, it will be able to reply to it. Other clients also associated to the TAP will not be able to acquire the channel before the TAP because their NAV field will not have expired when the test packet is transmitted. All other TAPs that need to quieten their clients can send CTS packets with NAV values equal to the duration of the experiment. Therefore, using this approach, the TAPs can perform experiments in a controlled setting without having to worry about clients interfering in the process. Obviously, this setup (i.e., back-off phase) incurs a cost in terms of the additional time required to perform each experiment. We provide a worst-case analysis of this overhead for each experiment in Section 5.4.1. Note, the effect of this cost is only incurred in cases where clients do not have hardware or software support for multiple interfaces. With hardware/software-based support, the separate channel allocated for signalling prevents the signalling traffic from contending with data. Therefore the extra setup costs do not matter in that case. In subsequent sections, we assume that a separate signalling channel (whether virtual or physical) is available to us for performing experiments.

### Experiment Agent

The experiment agent provides the functionality in the architecture for performing experiments. We describe the experiments that the agent carries out to detect different interference scenarios. Before describing these experiments, we first identify different scenarios that can arise in a wireless LAN deployment. There are

many interference scenarios that can occur and it would be an arduous task to list all of them. Fortunately, many of these are similar, in terms of the relationship between the interfering and interfered nodes. Therefore, we first classify these interference scenarios, and later discuss experiments that can detect interference in each category. We focus only on detecting scenarios of internal interference.

### Interference Classification

Our classification describes interference scenarios in terms of their distance (in hops) from the infrastructure. The intuition is that as the interference scenario moves further away from the infrastructure, it becomes progressively harder to detect and resolve. The classification is not perfect but provides enough basis on which to define a variety of experiments to detect many scenarios of RF interference. For this classification, we assume that the carrier-sense and interference ranges of the nodes are equal. In other words, we do not address the exposed terminal problem.

- *Inter-Access-Point (Zero-Hop) Interference:*

  There are two types of inter-TAP interference scenarios and we briefly describe them further. These are 'zero hop' interference scenarios since the interference is experienced *zero hops* from the network infrastructure.

  1. *Single Overlap Interference (SOI):* If the interference range of one TAP covers a neighbouring TAP while the reverse is not the case, the overlapped TAP suffers interference from transmissions by the overlapping TAP. The overlapped TAP is a *hidden terminal* from the perspective of the overlapping TAP. This scenario can increase the packet loss rate dramatically at the interfered AP, because of collisions with packets that are both sent and received by the overlapped TAP.
  2. *Dual Overlap Interference (DOI):* In this case, both TAPs mutually contend for the medium and the probability of their packets colliding is low. However, packets that are being received may still experience collisions if one TAP is receiving data while the other is transmitting.

  Inter-TAP interference is relatively straightforward to address in our framework because we assume control of the TAPs. The experiment for these scenarios is described later.

- *Access-Point - Client (One-Hop) Interference:*

We now describe interference scenarios that include clients that are associated to the network. The scenarios we describe involve both TAPs and clients. This is referred to as *one hop* interference as the interference is experienced one hop from the network.

1. *Overlapping Access Point (OAP):* Consider the case where a TAP's interference range covers a client connected to a neighbouring TAP. In this situation, the client experiences interference from this TAP, from whom the client may or may not be hidden. In the worst case where the client is hidden from the interfering TAP, packets both sent and received by the client are susceptible to collision with packets transmitted from the interfering TAP. This is similar to the SOI interference scenario, described earlier. However, these two scenarios are presented separately because their detection mechanisms are different in our architecture.
2. *Overlapping Client (OC):* In this case, the client's interference range covers a TAP adjacent to the TAP to which the client is associated. In the worst case where the TAP is hidden from the client, packets both sent and received by the TAP are susceptible to collision with packets transmitted from the interfering client. Resolving this scenario is difficult because clients cannot be instructed to reduce their power. Therefore, if the client is causing a lot interference at the affected TAP, the only alternative is to re-assign the TAP's channel. The degree of interference depends upon client characteristics. For idle clients, OC interference may not be a critical concern.

In the discussions above, we have assumed the worst case where either TAP or client is hidden from the interfering source. For the case where the interfered entity is not hidden, both OAP and OC cases occur together. In our evaluation section, we show how in most cases, OAP and OC interference are likely to occur together. The existence of both scenarios can be detected by performing experiments individually for each.

- *Inter-Client (Two-Hop) Interference:*

  Clients can also interfere with each other. Here, we are interested in the case where the clients are associated to separate TAPs. If clients are associated to a common TAP, they can perform an RTS/CTS exchange before each data transmission to reduce the chances of interfering with each other. Alternatively, if the TAP implements PCF (Point Coordination Function) functionality, it can coordinate client access to the medium, thereby reducing

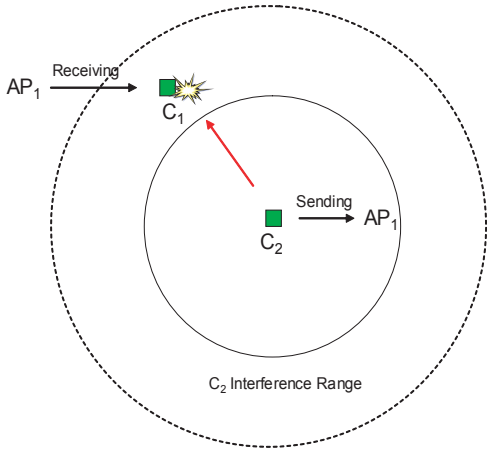


Figure 5.10: An example of an inter-client interference scenario where a client that is transmitting interferes with another client that is receiving.

inter-client interference for scenarios where clients are associated to the same TAP.

In scenarios where overlapping clients are associated to different TAPs, they interfere with each other if their respective TAPs use the same channel for communication. We describe two scenarios for inter-client interference. The first case is illustrated in Figure 5.10. In this case, the client $c_1$ experiences interference for traffic that it is receiving from its TAP. Client $c_2$ is hidden from $c_1$'s TAP and vice versa, thereby preventing $c_2$ from backing-off on its transmissions.

The second case is illustrated in Figure 5.11. Here, both clients are within interference range of each other, and send traffic to their respective TAPs. Therefore, for each packet transmission, they contend mutually for the medium. The probability of packet collisions is low in this scenario. However, the fact that the clients lie in each other's interference range prevents them from transmitting simultaneously, even though there is an opportunity to do so without causing collisions at the receivers. Once again, this case is difficult to handle because clients cannot be told to adjust their transmission power, unlike TAPs. Therefore, one possibility would be to have the clients' TAPs select different channels for communication. However, these scenarios are hard to detect by the infrastructure, because they are two hops away, In this thesis, we do not discuss mechanisms to address such scenarios.

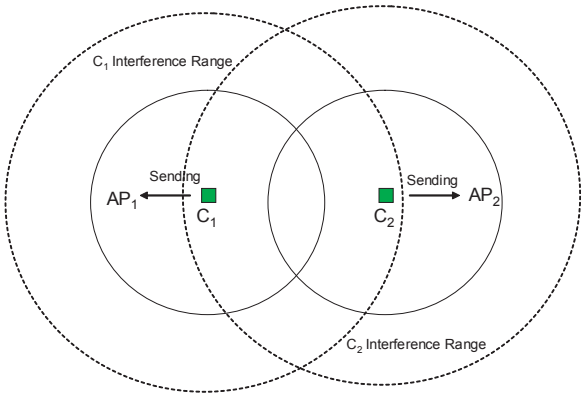


Figure 5.11: An example of an inter-client interference scenario where both clients contend mutually for the medium whenever either of them transmits data.

### Interference Detection Algorithms

We assume that the following interference experiments are performed in sequence, i.e., we first perform zero-hop experiments and then one-hop experiments. We use the following notation to describe nodes participating in an experiment. A node is a *Tester* if it initiates the experiment, to allow other nodes to observe interference. A tester may also observe interference for nodes that are not capable of doing so themselves, e.g. clients. A *Testee* is a node that checks to see if the *Tester* is interfering with it. Note, there is exactly one tester per experiment. However, we can have one or more testees in each experiment. Each experiment we describe is discussed both from the viewpoint of the tester and that of the testee(s).

- *Zero-hop Interference Experiment:*

  In the zero-hop experiment, a single TAP acts as the tester while all other TAPs act as testees. The tester transmits $m$ broadcast packets, where, for each packet, the testee(s) listens for interference. The testee(s) must synchronize with the tester to ensure that its observations are accurate. Synchronization is supported with the help of the time-synchronization daemon, described in Section 5.4.1. During a broadcast, the testee observes whether there is a change in the channel state, i.e., whether the channel transitions from *idle* to *busy*. If there is a change, then with high likelihood, the testee is in interference range of the tester. Moreover, if it is able to decode the broadcast packet correctly, then it is also in transmission range of the tester. We send out $m$ broadcast packets for this experiment to increase confidence

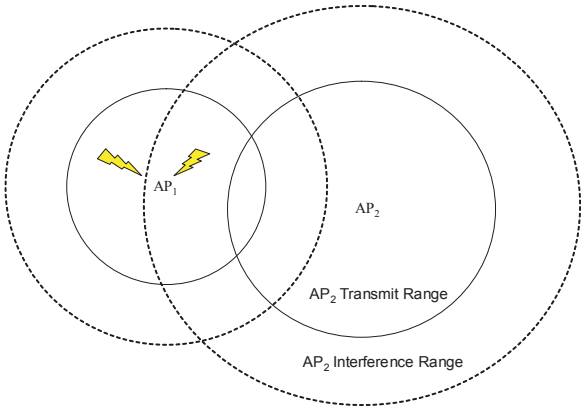


Figure 5.12: AP1 is interfered with by AP2. AP1's transmissions collide with transmissions initiated at AP2. The outer dotted rings outline the interference ranges of each of the APs.

in the results of the experiment. We later show that a relatively small value of $m$ suffices for this purpose.

Each TAP performs this experiment exactly once. Therefore, the number of experiments required in total to detect zero-hop interference is bounded by $O(N)$, where $N$ is the number of TAPs. An illustration of an inter-TAP interference scenario is shown in Figure 5.12.

- *One-hop Interference Experiments:*

  One-hop interference experiments are of two types: Those that test for OAP interference and those that test for OC interference. Recall, OAP interference occurs as a result of a TAP interfering with a client, while OC interference is caused by a client interfering with a TAP.

  - *Overlapping-Client (OC) Interference Experiment*

    A client interferes with a TAP when it is associated to another TAP, but when its interference range covers the TAP experiencing interference. This is depicted in Figure 5.13.

    For this experiment, the tester is the TAP to which the client is associated whereas the testee is the TAP that tests for interference. The tester transmits an RTS packet to the client. Upon receiving the RTS, the client responds with CTS. During the CTS transmission, the testee observes to see a change in channel state. If the testee detects a change,

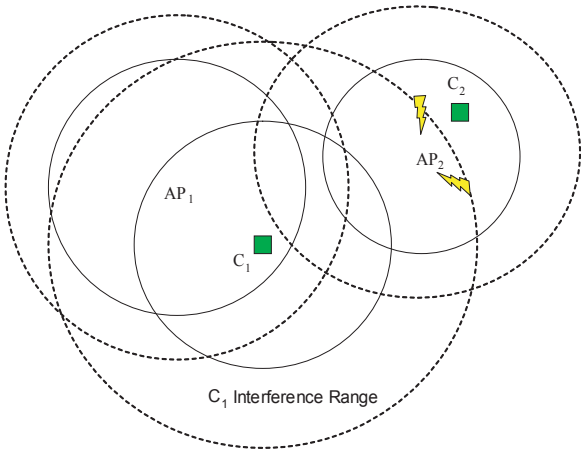


Figure 5.13: $C_1$ is associated to $AP_1$ but interferes with $AP_2$, to which $C_2$ is associated. $C_2$'s transmission to $AP_2$ collides with $C_1$'s transmission to $AP_1$.

then client-TAP interference exists between the testee and the client. After the tester receives the CTS packet from the client, the experiment is complete. This process is repeated $m$ times as well, to increase our confidence in the result.

A minor difficulty arises in this experiment. If the testee also experiences inter-TAP interference from the tester, then it must ignore channel state changes during the transmission of the RTS. This is achieved by having the testee ignore state changes for a duration equal to the propagation delay of the RTS packet, from the time at which the RTS transmission was initiated. This requires synchronization between the tester and testee, which is achieved by means of the time synchronization daemon. The propagation delay is fixed for RTS packets and can therefore be preset for this experiment.

We assumed for this experiment that the packet-delivery ratio on the link from tester to the client is high. In cases where the RTS packet transmitted by the tester gets lost, the testee could wrongly infer that there is no client-TAP interference between it and the client. To avoid this, the tester can notify the testee whether it received a CTS from the client. If not, the testee can choose to ignore the outcome of the experiment if it does not detect a change in channel state during its observation window. If the testee does observe a state change, then it can assume that the CTS packet was lost for the tester and thereby

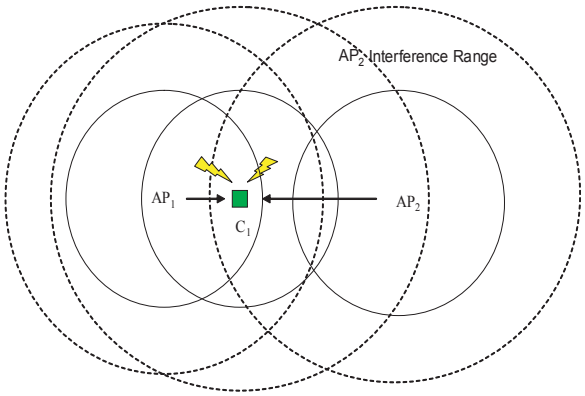


Figure 5.14: $C_1$ is associated with $AP_1$ but is interfered with by $AP_2$. $AP_1$'s transmission to $C_1$ collides with transmissions initiated at $AP_2$.

accept the results of the experiment. Using this approach, we can infer, with high probability, whether there is interference experienced by the testee, from the client.

This experiment needs to be performed for each client in the network. Therefore, for $C$ clients in total, the number of experiments required to detect OC interference is upper bounded by $O(C)$.

– *Overlapping Access-Point (OAP) Interference Experiment*

A TAP interferes with a client if the client lies in its interference range (see Figure 5.14). For this experiment, the tester is the TAP to which the client is associated. The testee is the TAP that may be causing interference for the client. The tester transmits an RTS packet to the client, while the testee simultaneously sends a broadcast packet. If the broadcast packet and the RTS packet collide at the client, the client will not receive the RTS transmission correctly. In this situation, it will not respond to the tester with a CTS, causing the tester to time out. If this occurs, the tester can assume that the RTS packet collided with the broadcast sent by the testee. The experiment completes after the either the tester receives a CTS from the client or times out in the process. This experiment is also repeated $m$ times.

This experiment also assumes a high packet-delivery ratio on the link from the tester to the client and vice versa. In cases where either the RTS transmitted from the tester or the CTS reply by the client gets lost, the tester may wrongly infer the existence of OAP interference. This assumption is harder to relax than for the OC experiments because

the result of the experiment depends on whether the tester receives the CTS. However, we argue that simply repeating the experiments can help alleviate the problem. If there is no OAP interference, then with high probability, after $m$ experiments the tester should receive a packet from the client. We show in the next section that a relatively small value of $m$ can provide good results.

OAP experiments are more cumbersome to perform than the previous experiments. This is because these experiments require TAPs to be pairwise synchronized for each client. Therefore, the total number of such experiments that need to be performed is bounded above by $O(CN)$, where $C$ is the total number of clients, and $N$ is the total number of TAPs. In subsequent sections, we discuss how this overhead can be reduced by utilizing information already known about the wireless deployment.

The experiments described above detect the presence of interference between two nodes. However, they do not indicate how much interference the nodes experience. The actual interference depends on how often the interfering node is transmitting, or its transmission throughput. For idle nodes, their interference is effectively zero, even if our interference experiments identify such nodes as interferers. For TAPs, this interference corresponds roughly to the total load that the TAP carries for its clients. For clients, this is approximated by the received throughput measured at their respective TAPs. These ideas are elaborated further in Section 5.4.1. However, both the TAP load and received throughput of a client can fluctuate considerably over time, depending on traffic characteristics. To smooth out these values, the TAP can maintain an Exponentially Weighted Moving Average (EWMA) of TAP load and received client throughput, and update these values periodically. Then, if our experiments detect an interferer, the actual interference that it causes on the interfered node corresponds to these average values we have computed.

### Analysis of Experiment Overhead

There is a tradeoff between the number of experiments that need to be performed and how accurately we capture the total number of conflicts in the system. In this section, we provide a worst-case analysis into how long it would take to perform each experiment as well as the overhead of performing all the experiments in sequence.

As we described earlier, before each experiment begins, a back-off phase needs to be initiated where TAPs send an unprovoked CTS to all clients. The back-off phase

is initiated for all experiments. The duration of the back-off phase is the amount of time it takes to transmit a CTS packet. RTS/CTS packets are transmitted at the lowest supported transmission rate, which is $1Mbps$ for IEEE 802.11b. Therefore, the worst case propagation delay of the CTS is $\frac{(14*8)}{(1.024*10^6)} \approx 107\mu s$, where 14 is the size in bytes of a CTS packet. We do not include in the back-off phase, the SIFS interval the TAPs need to wait before CTS transmission, as this interval would exist regardless of whether we were performing the experiment. We include the overhead for the back-off phase in each experiment. Further, we discuss in detail the overhead for each experiment. As described earlier, each experiment is repeated $m$ times. The numbers of TAPs and clients are denoted by $N$ and $C$, respectively.

- *Zero-hop Experiment:* The zero hop experiment involves a TAP sending out a broadcast packet for observation by other TAPs. The IEEE 802.11x standard mandates that broadcast packets be transmitted at the lowest supported rate. Therefore, assuming the worst case, each broadcast packet will suffer a propagation delay of $\frac{(28*8)}{(1.024*10^6)} \approx 214\mu s$, where 28 bytes is the smallest size for a broadcast packet, containing just the header of the packet. Moreover, each broadcast packet will be transmitted after a DIFS ($50\mu s$) interval. Therefore, the per-packet delay for the experiment is $214 + 50 = 264\mu s$. Because we transmit $m$ such packets, the total overhead is given as $(264 * m)\mu s$. Assuming each TAP performs this experiment in sequence, and we have a back-off phase before each experiment, the total overhead of the zero-hop experiments is $(371 * N * m)\mu s$.

- *One-hop OC Experiment:* One-hop OC experiments require the tester to perform an RTS/CTS exchange with the client. Both the RTS and CTS will need to wait for a SIFS interval before transmission. The propagation delay for an RTS packet is $\frac{(20*8)}{(1.024*10^6)} \approx 163\mu s$, where 20 bytes is the size of the RTS packet. We already showed that a CTS packet incurs a propagation delay of $107\mu s$. Therefore, the total overhead for the experiment is given by $(2 * 107 + 163 + 2 * (SIFS)) \approx 378\mu s$, including the additional time for the back-off phase. Note, this delay is a little more than that for the broadcast packet. This fact will be useful for the OAP analysis we discuss next. For OC, we perform the experiment for $C$ clients and each experiment is repeated $m$ times. Therefore, the total overhead for OC experiments is given as $(378 * C * m)\mu s$.

- *One-hop OAP Experiment:* The OAP experiment involves TAPs synchronizing to send packets to the client, and the client replying with a CTS. For this experiment, either a collision occurs at the client in which case the tester TAP

will timeout, or a CTS is received for the RTS. Because the timeout period defined by the standard is the same as the time it would take for the CTS to be received by the tester, the two delays are identical. Moreover, because the broadcast packet transmitted by the testee takes less time than the RTS/CTS exchange by the tester, it can be safely ignored in our calculations. Therefore, each OAP experiment takes $\approx (378 * m)\mu s$, including the back-off phase for each experiment. Because each client needs to be tested with $N - 1$ TAPs in total, and there are $C$ clients, the total overhead for OAP experiments is given as $(378 * (N - 1) * C * m)\mu s$.

Consolidating the total overhead for each type of experiment, and assuming each experiment is performed in sequence, the worst case time for completing all experiments is:

$$T = m * ((371 * N) + (378 * C) + (378 * (N - 1) * C)\mu s \tag{5.8}$$

*Considerations for m:* We now discuss how to choose a value for $m$. Assume the packet delivery ratio between a transmitter and receiver is $P$. $P$ represents the probability of successfully delivering a packet to the destination. When selecting a value for $m$, we are interested in a value where the probability of successfully delivering one out of $m$ packets to the destination is high. This probability, $Pb_m$, is:

$$\begin{aligned} Pb_m &= P + (1 - P) * P + ((1 - P)^2) * P + \ldots ((1 - P)^{m-1}) * P \\ &= P(\frac{1 - (1 - P)^m}{1 - (1 - P)}) \\ &= 1 - (1 - P)^m \end{aligned} \tag{5.9}$$

Therefore, either the first packet is received, or the second is received, or the third is received, and so on, up to the $m$th packet. Even for a poor link (e.g., where $P = 0.30$), a value of $m = 5$ suffices to yield a high probability of successfully delivering a packet to the destination ( $Pb_m = 0.84$ ). Therefore, we argue that even modest values of $m$ can provide high confidence in the result.

For a typical deployment, with values of $C = 200$, $N = 20$, and for $m = 5$, a value of $\approx 7.6s$ is obtained for $T$. This represents the worst case where all experiments are performed in sequence. In the next section, we discuss possibilities for reducing this overhead by performing experiments opportunistically and in parallel.

### Scheduling of Experiments

Performing each experiment in sequence is impractical and costly. From the previous discussion, we identify OAP experiments as being the dominant factor impacting the value of $T$. OAP experiments alone constitute 90% of the overhead. Therefore, we discuss techniques to reduce factors affecting the overhead of these experiments. The two factors are the number of testee TAPs with whom we must conduct experiments and the number of clients for which OAP experiments need to be performed.

First, we claim that OAP experiments do not need to be performed with all TAPs acting as testees. With high probability, clients are likely to suffer interference only from TAPs that are neighbours of TAPs to which they are associated. In most deployments, it is highly unlikely that all TAPs are neighbours of each other. Mishra et al [67] find average node degrees of 4 even for a dense deployment of 70 access points. Moreover, Akella et al [31] find that even for unplanned deployments, typical node degrees lie between 3-8. These insights can help us reduce the overhead of the OAP experiments by almost 60%. However, there may be certain cases where a TAP that is not a neighbour of another TAP in the underlying conflict graph can still cause interference on some client of the TAP. In such cases, we advocate using the results of the OC experiments for the client to determine if OAP experiments need to carried out with a particular TAP. Clients typically transmit at maximum power, so that they can communicate with access points if they happen to reside at the boundary of the cell. Because of this, access points that cause OAP interference on the client will also likely suffer from OC interference by the client. Thus, we can use the information from the OC experiments to determine whether OAP experiments need to be performed with a particular TAP. Note, this requires that OC experiments be performed before the OAP experiments. The schedule for performing OAP experiments would then involve assigning clients to time slots such that only clients that have mutually independent OC conflict sets are assigned to the same slot for conducting experiments. Based on arguments similar to those we stated for the TAPs, clients are also not likely to have OC conflict set sizes larger then 4, for dense deployments. We intend to explore these and other techniques for scheduling experiments as part of future work.

Next, we claim that for ongoing maintenance of conflict information, we are not required to perform experiments for all clients each time a change needs to be incorporated into the annotated conflict graph. If a client associates to an access point, only experiments for that client need to be performed. During the power control phase, because powers are only changed incrementally, experiments need to be performed only with the TAP whose power was altered. Only in very extreme

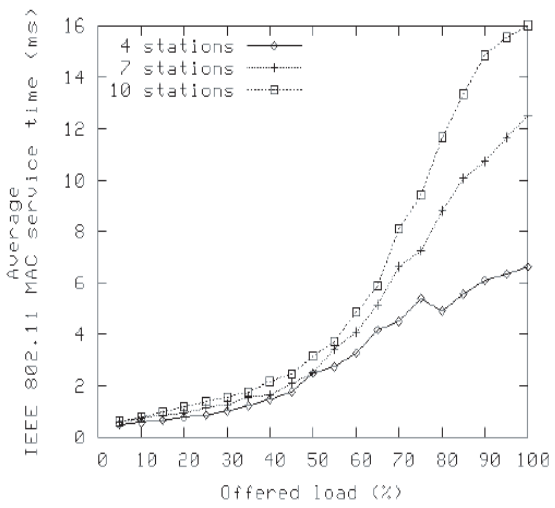


Figure 5.15: An example illustration (from [63]) of mean MAC service time for differing amounts of offered load. The x-axis represents the offered load by each client relative to the wireless channel's capacity. The three curves represent the number of stations associated to the access point.

cases, where all conflicts are discarded and need to be recomputed, will experiments need to occur for all clients. These situations are not likely to occur often and are based on the $\beta$ threshold we presented earlier. Therefore, we can use this threshold to tune the performance of the system.

### Observation Agent

The observation agent's function is to collect network statistics that are used for computing the utility of the system. The observation agent periodically sends observation reports to the controller, which computes the utility on behalf of the TAPs. The agent can be implemented as a daemon that passively collects and reports statistics from different layers of the network stack. There may be many parameters of interest and we illustrate some that are relevant to the discussions in previous sections.

1. *Load:* In our architecture, the load metric is relevant for load-balancing purposes and for quantifying interference. There are two types of load that we are interested in:

- TAP load is the total downlink load experienced by the TAP, as a result of servicing clients connected to it. The definition of load we consider here takes into account the traffic demand at the TAP. This can be approximated using mean *MAC Service time* [63]. MAC Service time is defined as the time that elapses between when the MAC layer receives a packet and when it is handed to the PHY layer for transmission. Although there is no correlation between the MAC service times of any two packets, the mean MAC service time stabilizes quickly and provides a good approximation to the load experienced at an access point, as is shown in Figure 5.15. This graph presents the MAC service time for an IEEE 802.11x-compliant access point for different amounts of offered load. An Exponentially Weighted Moving Average (EWMA) of the MAC service time can be computed and maintained by the observation agent. It is relatively straightforward to instrument the MAC layer to collect this type of information.
- Client load is defined as the transmission throughput of the client. This information is useful for annotating interference edges from clients to access points. We approximate this load by maintaining a running average of the number of packets received per second from the client. This is a rough approximation of the load at the client because it assumes no packet loss from the client to the TAP. More accurate estimates could be obtained if clients explicitly provided this information. However, this requires client modifications, which we do not advocate for our system.

2. *Ambient Noise:* The ambient noise of a wireless channel is the average level of background noise present in that channel. This information is relevant for assessing channel quality when deciding between different channels. This allows our system to not only choose channels where there is less internal interference, but also channels in which the noise from external interference is minimal. The agent can also measure this for the signalling channel, if it is physically separate from the data channel. Based on this information, if the controller detects too much background noise on the current channel, it may choose to switch to a quieter channel. This is important for signalling purposes because the outcome of the experiments depend on the amount of background noise present in the channel.

3. *Client Wireless Rate:* The wireless rate of the client defines the physical rate and modulation technique the client uses for sending data packets to the TAP. This information can be useful for load-balancing purposes as well as for estimating average client throughput in a cell. For load balancing, if the TAP

detects that a particular client is using a very low transmission rate relative to the other associated clients, it may choose to migrate that client to a neighbouring access point, to improve service for the others. Without information on the client's wireless rate, such decisions are hard to make, especially if clients are bursty in nature. The wireless rate being used by clients is piggybacked on the PLCP header, as was discussed in Chapter 2. However, the PLCP header is not exposed to the MAC. Therefore, cross-layer mechanisms may be required to obtain this information. Alternatively, average Received Signal Strength (RSS) information can also be used to obtain an estimate of the wireless rate of the client.

### Time-Synchronization Daemon/Time Daemon

In this section, we provide a description of both the time-synchronization daemon that runs at the TAP and the time daemon that runs at the controller. The time-synchronization daemon is responsible for synchronizing the TAP with the controller's time daemon. This allows each TAP to maintain an accurate view of the controller's clock. The time daemon at the controller ensures that the TAPs are correctly synchronized during each experiment. As we illustrated earlier, our interference experiments rely on strong synchronization primitives between TAPs. For example, in the one-hop OC interference experiment, it is important for both TAPs to send packets simultaneously in order to affect a collision at the client. If the TAPs are not correctly synchronized, the results obtained from this experiment would be incorrect.

The implementation of the daemons can be supported with the help of the popular Network Time Protocol (NTP) [64]. The base implementation of NTP only provides accuracies of within a millisecond on LANs. However, cheap and relatively straightforward hardware/software upgrades for pulse-per-second (PPS) support, can provide accuracies of upto a nanosecond, which is more than sufficient for our purposes. Details of such techniques are available in [65].

An example of the synchronization that occurs between the TAPs is illustrated in Figure 5.16. For the inter-TAP interference experiment, at time $X$, the controller sends a "Run $Experiment_1$" command to TAP 1, which is the tester. At the same instant, it also sends an "Observe $Experiment_1$" command to TAP 2, which is the testee. Tap 2 observes the environment for the duration of the experiment and then sends back the results, called "$Experiment_1$ Report", to the controller at time $Y$.

In addition to the experiments, TAPs also periodically send observations of the environment (using the observation agent) to the controller, at fixed time intervals.

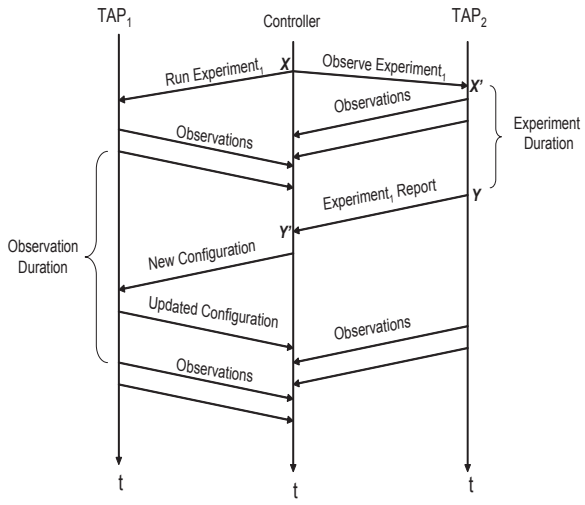


Figure 5.16: Timing Diagram Illustrating communication between TAPs and Controller

Depending on how the signalling plane is implemented (i.e. virtually or physically), observations may or may not overlap periods at which experiments are performed.

### 5.4.2 Central Controller

#### Utility Function

The utility function module is responsible for computing individual component utilities that are used for graph construction by the optimizer. It exposes an API to the network administrator to obtain information on the specifics of each parameter. The information that needs to be specified for each parameter is: parameter name, relationship to utility, parameter definition, and parameter weight. Example parameters that can be captured are shown in Table 5.1.

The utility-function module is also responsible for collecting conflict information from the environment. Therefore, it is the utility module that controls the running of experiments at each of the TAPs. Scheduling and other information that is required for the experiments is maintained by this module. The utility values that are thus computed, are then stored in the statistics database. Note, the utility-function module stores conflict information separately in the database as well. This is utilized by the optimizer, as we describe next.

Table 5.1: Example Utility Function Parameters

| Utility Parameter | Relationship to Utility |
|---|---|
| Throughput | Positive |
| RTT Delay | Negative |
| Packet Loss Rate | Negative |
| Access Point Load | Negative |
| Access Point (Client) Interference | Negative |
| Client Energy Consumption | Negative |

The strength of our architecture is that it allows the network administrator to tune the utility function. If a change occurs to the utility parameters or their weights, the utility-function module can notify the optimizer to use the new utility values that it has computed based on latest parameter information, once these utility changes have been inserted into the statistics database. Note, the interference experiments will probably not need to be performed again because conflict information will likely remain unchanged as a result.

### Optimizer

We described the high-level functionality provided by the optimizer module in Section 5.2. We now discuss its implementation details.

1. *Conflict Graph Construction:* The utility-function module notifies the optimizer once it has collected and stored conflict information in the statistics database. The optimizer uses this information to construct the conflict graph. Note, although the edges of the conflict graph discussed earlier also contain load information, this is not directly represented in the conflict graph when it is constructed. This is because this information is likely to change considerably over the lifetime of the conflict graph, causing the assigned load values to become stale. During the running of the channel-assignment and power-control algorithms, this information is retrieved from the statistics database and matched to the edges of the graph to obtain actual interference information. Recall, the conflict graph we construct is the maximum power single channel (MPSC) graph. Conflict information for this graph rarely changes,

unless an access point fails or a new one is installed. Therefore, we can compute this conflict graph once and maintain it over a long period of time, e.g. an entire day. This graph may be updated during periods when the network is idle, e.g. overnight. Thus, construction of the conflict graph does not impose any significant overheads in our system.

2. *Optimal Channel Assignment:* Intuitively, it seems likely that optimal channel assignments that are computed based on conflict-graph information would also not change often. However, because optimal channel assignment and the construction of the updated conflict graph occur based on load information of the interfering nodes, these can change on shorter timescales. This separation between the MPSC graph computed in the previous step and the updated conflict graph computed in this step allows our system to cater to changes in the environment, while at same time keeping the overhead of computing conflicts low. Note, load information may not be immediately available when the system is installed. In such cases, it can be supplied initially by the network administrator, to reflect the capacity requirements at different points in the deployment. Our system can then refine this information over time.

3. *Annotated Conflict-Graph Construction and Channel Refinement:* Recall, the annotated conflict graph incorporates client information as well. If no clients are present, which may occur at system startup, the annotated conflict graph is simply the conflict graph that was computed in the previous step and no channel refinement needs to be done. As clients associate to the network, experiments for these clients are performed and the ACG is updated incrementally. For simplicity, Figure 5.3 does not illustrate incremental updates to the ACG.

   Clients typically exhibit quasi-static mobility patterns [55]. Therefore, in most cases, the annotated conflict graph will maintain an accurate view of the conflicts of the clients. Nevertheless, for changes that do occur, these can be applied incrementally to the ACG as well. Based on arguments similar to those for the CG, the ACG also does not contain load information directly. Load information is retrieved during the channel-refinement and power-control phases of the optimization.

4. *Optimal Power-Level Assignment:* Optimal power assignment utilizes the ACG for computing power levels. As mentioned earlier, the ACG can change as the powers of the access points are adjusted. Therefore, for each step in the power-control algorithm, we need to update the ACG. The optimizer achieves this by notifying the utility function module of the change, which

then re-initiates some experiments at the TAPs. Note, as we indicated earlier, because at one most power change occurs in the power control algorithm per step, updates to the ACG are localized to the TAP whose power was updated. Therefore, in the worst case, at each step of the algorithm, we are required to perform experiments for the TAP whose power was adjusted. This requirement, based on Equation 5.8 and the deployment parameters we specified earlier, amounts to an overhead of approximately $378ms$, which includes the zero-hop and OAP interference experiments. Therefore, ACG updates do not impose a significant overhead in our system.

Another important requirement for the optimizer is determining whether a decrease in power during the power-control process causes a client disconnection. Identifying whether a client disconnected can be done by means of a simple test performed on the signalling channel. After the power is reduced by one step, the TAP (to which the client is associated) can perform one or more RTS/CTS exchanges with the client. If the TAP notices a decrease in the number of CTS replies from the client, it can assume that the client is in danger of being disconnected. A decrease to that particular power level can then be avoided.

5. *Incremental Power Refinement:* The functionality required for incremental power refinement is similar to that required for optimal power assignment, i.e., supporting updates to the ACG and detecting client disconnections. The frequency at which power refinement is done represents a tradeoff between system agility and the overhead incurred in doing the refinement. This tradeoff can be modeled by means of a tuning parameter supplied to the optimizer, set based on the specifics of the deployment environment.

   Channel assignment and power parameters eventually become out of date due to changes in the environment. Therefore, a re-computation step may need to be performed after some time. Re-computation is achieved by discarding the current ACG and sending an experiment re-initiation message to the utility-function module. In most cases, unless the original conflict graph also needs to be updated, only experiments involving clients need to be performed. The re-computation of the ACG is controlled by the threshold parameter ($\beta$), as described earlier.

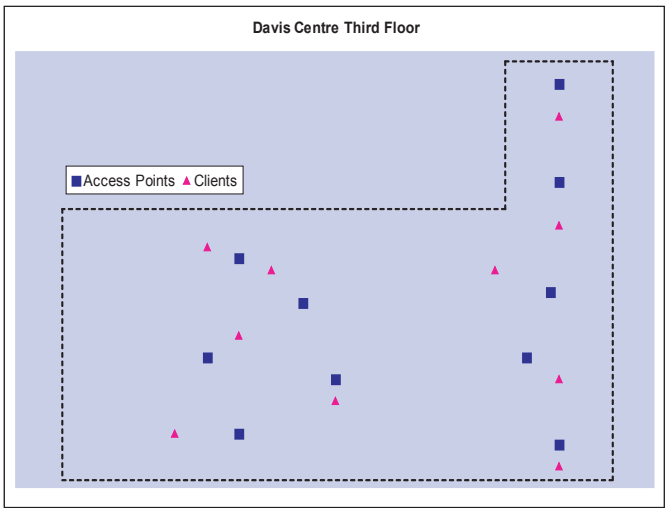


Figure 5.17: Access Point Layout for Davis Centre Third Floor. Clients locations are relevant for the results discussed later.

## 5.5 Preliminary Evaluation

We are currently in the process of validating the design and scalability aspects of our architecture. In this section, we present preliminary results that were obtained through simulation. The components that we implemented for these results are the experiment framework for detecting interference, the channel-assignment and refinement algorithm, and finally our proposed power-control algorithm.

### 5.5.1 Evaluation Methodology

We used QualNet, a commercially available network simulator (from Scalable Networks), that precisely simulates the MAC and PHY layers for the IEEE 802.11b/a/g standards. For implementing our proposed techniques, we instrumented the IEEE 802.11 MAC layer (common to all IEEE 802.11x standards). In our simulations, we used the access-point layout of the third floor of Davis Centre (DC) to evaluate our algorithms. Information about access-point layout and their corresponding configurations were made available to us by the Information Systems and Technology (IST) service organization, at the University of Waterloo. An illustration of this topology is shown in Figure 5.17. The topology contains 10 access points that are spread out across the building to provide complete coverage. In our simulations,

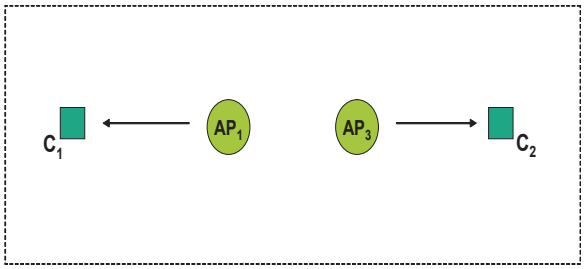


Figure 5.18: The four node topology used for the inter-access point interference experiment. The modified topology for the other two interference experiments swaps the locations of $AP_1$ and $C_1$ from their current positions.

each access point had one client associated to it, and which established a constant bit rate (CBR) traffic flow at rates high enough to saturate the medium. The packet size for all traffic was set to 512 bytes with a fixed transmission rate of 1Mbps. We did not examine the effects of dynamic rate-adaptation algorithms in this study. We also evaluated our algorithms on a denser version of the DC topology, where we randomly placed 10 additional access points and clients on the map. Note, in our simulations, clients were associated to the access point with the strongest signal strength. We currently do not provide load balancing across access points using power control, leaving its implementation to future work.

The utility function that we considered for our simulations captures two performance metrics, namely, client throughput, and RF interference. Both metrics were assigned equal weights, (i.e. 0.5). More sophisticated utility function evaluations are left for future work. For comparison purposes, we evaluated two forms of our proposed algorithms; one that only performs channel assignment (OPO), and the other that also performs power control (OPO /w TPC). These were evaluated against the manually optimized channel and static power configuration used for the DC building. We present this evaluation to illustrate the benefits of using dynamic optimization over static optimization. In addition, to illustrate the benefits of our proposed dynamic optimization algorithms, we compared them against a well-known Least Congested Channel Search (LCCS) algorithm [66]. LCCS is considered the current state-of-the-art algorithm for channel assignment. It works as follows: Each AP periodically observes data transmissions from other access points and clients on its assigned channel. If the number of transmissions it hears exceeds a pre-specified threshold, it tries to move to a channel that is less congested. LCCS serves as an illustration of how well local tuning algorithms can perform in comparison with our centralized channel assignment and power control algorithms. The

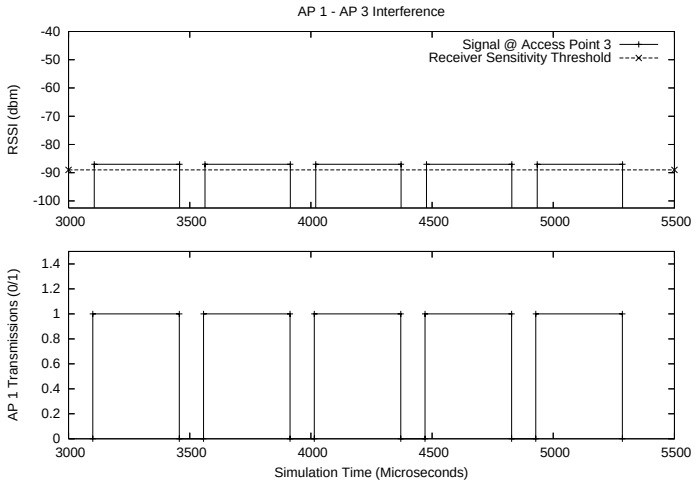


Figure 5.19: $AP_1$ causes interference on $AP_3$, represented as an increase in received signal strength at $AP_3$ during $AP_1$'s transmissions

LCCS implementation that we used did not observe data transmissions explicitly, but instead was supplied with interference information, computed through experiments. This improved the accuracy of the algorithm as it is now able to accurately pinpoint neighbouring access points it is likely to interfere with, without having to infer this through observation. We called this enhanced version of the algorithm 'optimized LCCS' (or oLCCS).

### 5.5.2 Evaluation Results

#### Validation of Experiments

We first present proof-of-concept results for the interference-detection experiments we discussed earlier. For these results, we used a small four-node topology. The layout of the topology is presented as we discuss results for each experiment. Note, the value of the noise floor in our results is $-102.5dbm$, represented as the origin on the y-axis of all graphs that plot Received Signal Strength information.

- *Inter-AP interference:* The topology we used for this experiment is shown in Figure 5.18. In this scenario, both APs send data to their clients and each lies

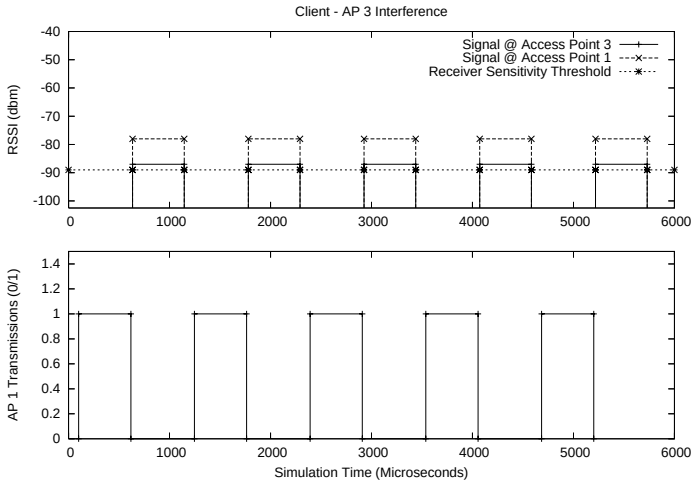


Figure 5.20: $AP_3$ experiences interference from $AP_1$'s client, represented as an increase in received signal strength at $AP_3$, during the client's transmissions

in the other's interference range. The result of $AP_3$ detecting this interference scenario is illustrated in Figure 5.19. The graph at the bottom plots the time at which broadcasts are sent by $AP_1$, during the experiment. The top graph plots the signal strength received by $AP_3$. From the figure, we see that as soon as $AP_1$ transmits a broadcast packet, an increase in signal strength beyond the receiver sensitivity threshold is observed at $AP_3$. This increase persists for the duration of the broadcast. Assuming $AP_3$ is aware of when $AP_1$ sends the broadcast, this result indicates that inter-AP interference can be detected by $AP_3$. The receiver sensitivity threshold used in our simulations is $-89dbm$, which is the minimum signal strength at which $AP_3$ can observe a change in the state of the channel. The results in which $AP_1$ detects interference from $AP_3$ are symmetric.

- *Client-AP interference:* For the Client-AP interference scenario, we modify the previous topology by inter-changing the locations of $AP_1$ and its client. Therefore, we expect $AP_3$ to suffer from Client-AP interference in this scenario. This is illustrated in Figure 5.20. The graph on top shows the received signal strength at both APs. The graph on the bottom indicates the time at which RTS packets are transmitted by $AP_1$. Because we do not have access

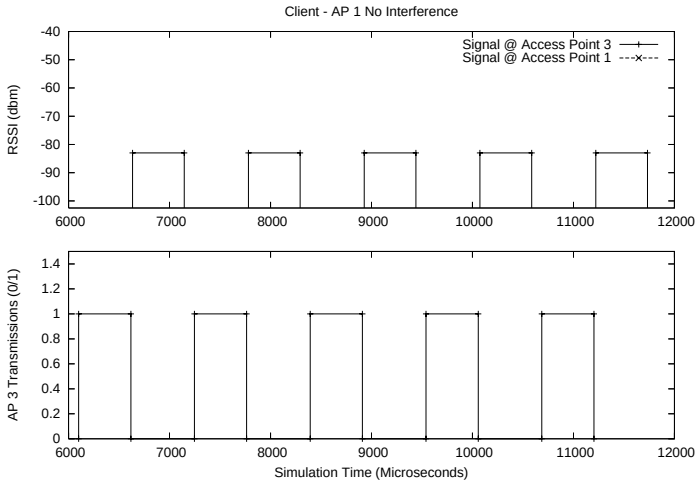


Figure 5.21: $AP_3$'s client does not cause interference on $AP_1$, since there is no increase in received signal strength seen at $AP_1$ when the client is transmitting

to received signal-strength information at the client-end in our architecture, this information is not illustrated. We see from these results that once an RTS packet transmission by $AP_1$ is complete, both APs observe an increase in signal strength on the wireless channel. This increase in signal strength occurs, with high probability, due to the CTS packet transmitted by the client. In reality, the client waits a SIFS interval (approx. $10\mu$s) before transmitting the CTS packet. However, because of the timescales shown on the graph, this is not visually evident. Nevertheless, this result illustrates that $AP_3$ is able to detect interfering signals from the client, thereby validating the correctness of our proposed Client-AP interference experiment. We also present results to show that $AP_1$ does not suffer from Client-AP interference as a result of $AP_3$'s client. This is shown in Figure 5.21. In this experiment, no signals are picked up by $AP_1$ as a result of CTS transmissions by $AP_3$'s client. Therefore, because of the differences in the outcome of each experiment, our experiment can clearly identify scenarios of Client-AP interference.

- *AP-Client interference:* The topology used for AP-Client interference is identical to the one used for Client-AP interference. In this case, $AP_1$ detects interference on the client's behalf. The result of this experiment is illustrated

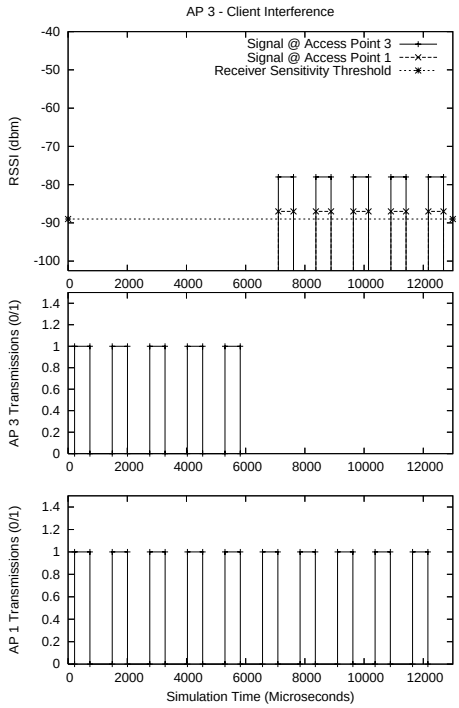


Figure 5.22: $AP_1$'s client does not respond during the first five transmissions due to packet collisions. Packets are returned for the second set of transmissions because $AP_3$ is no longer transmitting.

in Figure 5.22. For the first five transmissions, when $AP_3$ sends broadcasts concurrently with $AP_1$'s RTS transmissions, no corresponding CTS packets are received by $AP_1$. However, for the next sequence of transmissions when only $AP_1$ sends out RTS packets, it receives back CTS packets from its client. In fact, even $AP_3$ can hear the CTS signals from $AP_1$'s client, indicating the presence of Client-AP interference. Therefore, by identifying the loss of CTS packets for the first five transmissions, $AP_1$ is able to detect the presence of AP-Client interference, caused by $AP_3$. To further illustrate that the reverse case is not true, i.e. AP-Client interference caused by $AP_1$ on $AP_3$'s client, we perform a similar experiment with $AP_3$'s client. This is shown in Figure 5.23. In this case, $AP_3$ receives CTS signals from its client irrespective of whether $AP_1$ is transmitting. Therefore, $AP_1$ does not cause AP-Client interference on $AP_3$'s client.

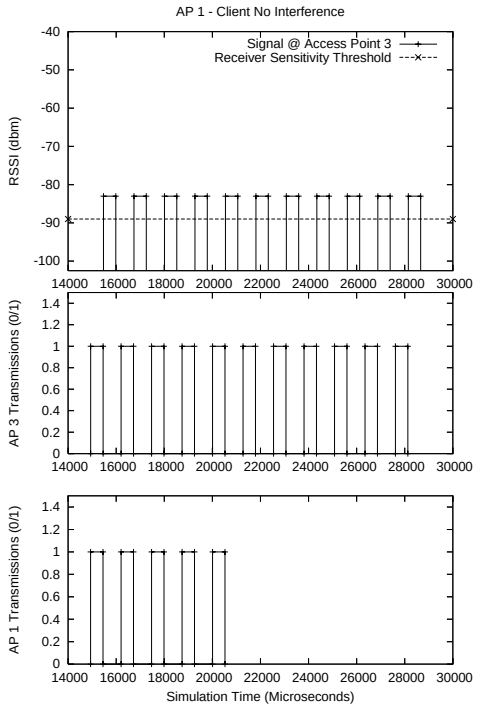


Figure 5.23: $AP_3$ receives responses from it's client, irrespective of whether $AP_1$ is transmitting or not.

These results serve to validate our experimental approach to detecting interference. For the next set of results, we use this information as part of our channel-assignment and power-control algorithms, which are evaluated further.

### Evaluating Channel Assignment and Power Control

We first present results for our algorithms on the original DC topology, which we call SparseDC, because the topology features a sparse deployment of access points. The modified denser DC topology is called DenseDC. Our performance analysis uses the mean throughput per user as the utility obtained by a configuration. The throughput distribution across clients is also analyzed in this section. Single-channel results that use a static transmit power are illustrated to serve as a baseline for comparison. Note, in our simulations, all the dynamic algorithms first generate and apply the configurations to the access points, before any traffic flows are initiated.

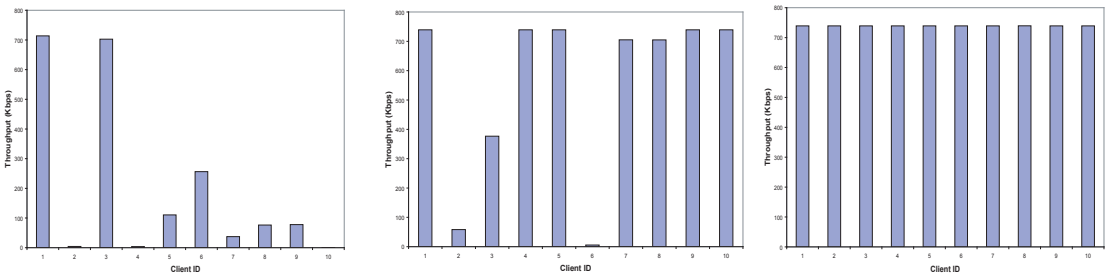


Figure 5.24: Client Through Distribution (Left: Single Channel, Centre: DC Configuration, Right: Dynamic Algorithms

These results, therefore, examine the quality of the configurations generated by the algorithms, instead of their ability to adapt to changes in the environment, which is a subject of future work.

- *Client Throughput Distribution* We first present results of the client throughput distribution for each algorithm, for a single simulation run. The transmit power of all access points and clients is set to $20dbm$, which represents a low-interference scenario. We present results for the SparseDC topology.

  Figures 5.24 compare the throughput distribution across each of the 10 clients that are connected to the access points. Starting with the leftmost chart, the single channel approach yields the worst throughput distribution across clients. Two clients obtain maximum throughput (which is approximately 740 Kbps in our simulations) while all others did not even receive half of the maximum. In fact, 3 out of 10 clients obtain almost no throughput at all. Clearly, this is unacceptable in terms of utility maximization. When we observe the throughput distribution for the hand-optimized multi-channel DC configuration (middle chart), we see a substantial improvement in throughput. Approximately 70% of the clients obtain maximum throughput. This points to the benefits of using multiple channels in improving the aggregate throughput of the network. However, not all clients receive this throughput and we still have at least two clients that receive little to no throughput. Finally, by observing the rightmost chart, which is the same for all the dynamic algorithms (i.e. oLCCS, OPO, and OPO with power control), we observe that all users now get the maximum throughput from the network. This illustrates the benefits of dynamic optimization in its ability to detect interference and subsequently mitigate it.

- *Mean User Throughput vs. Number of Channels:* We now present results to

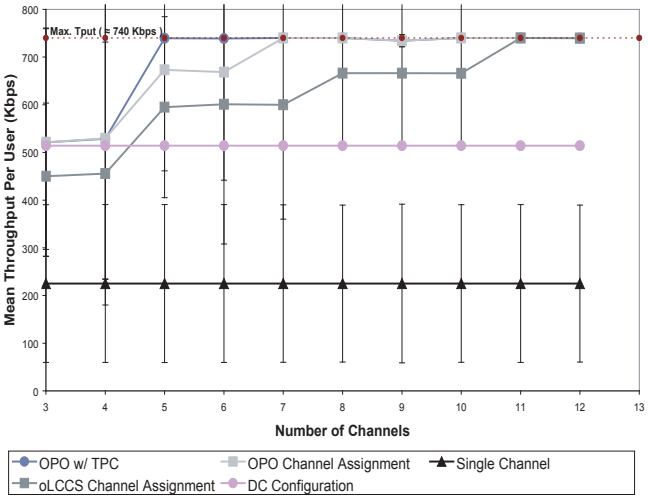


Figure 5.25: Performance of algorithms in Sparse (Original) DC High-Interference Scenario

analyze performance as we vary the number of channels. The performance metric that we use is the mean throughput per user, i.e., the average throughput obtained across all clients. Intuitively, the greater the number of channels, the greater the mean throughput per user. We present two sets of results for this case; those obtained on the SparseDC topology, and those obtained on the DenseDC topology. For each set of results, we compare the performance of the algorithms in high and low-interference scenarios. A low-interference scenario corresponds to a transmit power of $20dbm$, whereas a high interference scenario corresponds to a transmit power of $30dbm$.

For the SparseDC low-interference scenario, all dynamic schemes performed alike and provided the maximum mean throughput per user, irrespective of the number of channels. Therefore, we do not present results for this scenario. Instead, Figure 5.25 presents results for the SparseDC high-interference scenario. For this result, we see that the DC configuration provides only twice as much throughput over the single-channel case, even though we use three times as many channels. This result illustrates that although using multiple channels does improve the average throughput of each user, it does not increase linearly as a function of the number of channels. When we compare the performance of the dynamic schemes with the DC configuration, we find little improvement for a limited number of channels. This is because of the

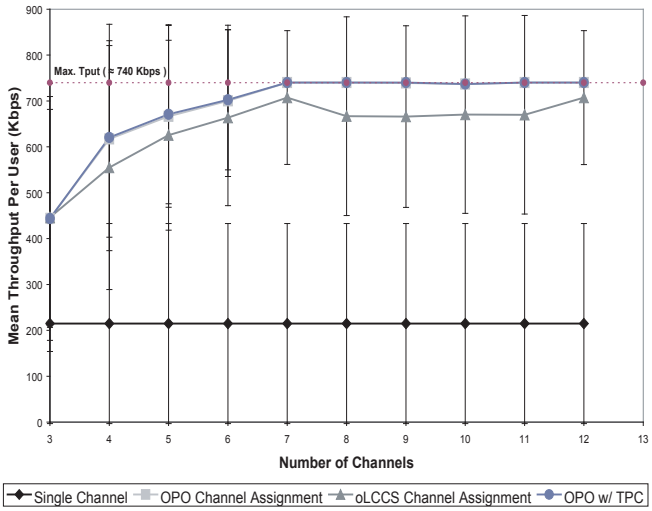


Figure 5.26: Performance of dynamic optimization algorithms in Dense DC Low-Interference Scenario

large degree of interference that is present in this scenario. As a result, even a very good channel-allocation algorithm will not be able to resolve all conflicts to provide maximum throughput to all users. However, as the number of channels is increased, the dynamic schemes exploit this additional flexibility and quickly approach the maximum throughput mark. In particular, OPO with power control converges quickly to the maximum throughput because it also adjusts the transmit power of access points to further reduce conflicts. oLCCS takes longer to converge because it is not able to utilize the additional channels optimally, using just local tuning. This result therefore highlights the benefits of a coordinated approach to channel assignment that yields better results in scenarios characterized by high interference.

For the DenseDC low-interference scenario, we observe the results illustrated in Figure 5.26. The dynamic schemes no longer provide maximum throughput for all clients, irrespective of the number of channels. In this plot, we observe up to a 40% decrease in the mean throughput per user, when compared with the SparseDC case (for 3 channels). This is attributed to the larger degree of interference that is present due to additional access points and clients. Nevertheless, even with this reduction in throughput, the dynamic schemes that use multiple channels can still provide at least a factor of 2 improvement over using just a single channel. When comparing the dynamic schemes with

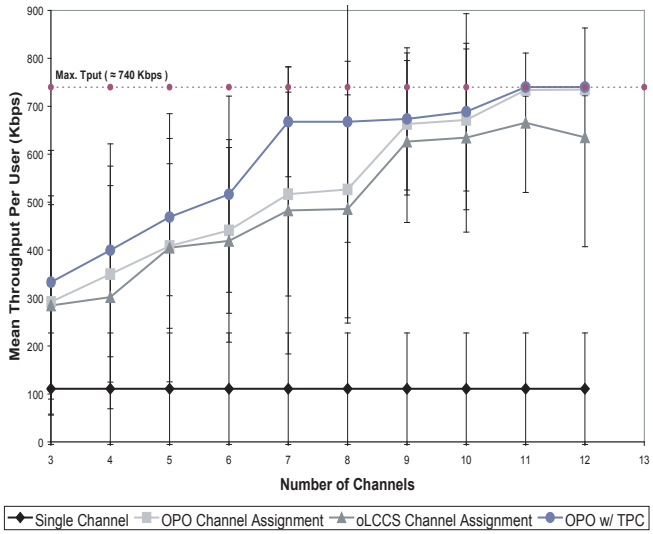


Figure 5.27: Performance of dynamic optimization algorithms in Dense DC High-Interference Scenario

each other, we observe that oLCCS no longer matches the performance of the other dynamic schemes. Because of access-point coordination that is exploited by the other two schemes, they are able to perform better than oLCCS.

Finally, we present results for the DenseDC high-interference scenario, shown in Figure 5.27. In this scenario, we observe a further decrease in the throughput for each scheme. The interference caused due to both a denser topology and higher transmit power prevents any of the schemes from providing a high mean throughput for users, with only a limited number of channels. As expected, the throughput does improve gradually with an increase in the number of channels. More interestingly, however, we observe that both optimal channel assignment and oLCCS perform closer to each other. We conjecture that in scenarios characterized by a large number of conflicts, even an optimal solution to channel assignment may not be good as all solutions yield poor performance. This is precisely what we observe in this result, where the benefits of global coordination (used in OPO) over local optimization (used in oLCCS) are only marginal. However, OPO with power control performs better than both OPO and oLCCS. This is because it is able to reduce the number of conflicts that could not be resolved simply by performing channel assignment. However, these improvements are not very significant, primarily

because our simple power-control algorithm only takes into account inter-access-point conflicts during the power-control process. A more sophisticated algorithm that also incorporates other types of conflicts (e.g., client-to-access-point conflicts) is expected to perform even better.

## 5.6 Discussion

Building a self-managing infrastructure for wireless LANs is a challenging task because of the complexity and unpredictability of wireless technology. These problems are exacerbated by the continuously evolving wireless landscape. Today, a plethora of wireless devices utilizes the same unlicensed RF spectrum as wireless LANs. This creates an interference nightmare in wireless networks and results in severely degraded network performance. Studies have shown throughput reduction factors of up to four times in field measurements [4]. Self-management infrastructure is thus a key requirement for Wireless LANs.

Existing solutions covered in this thesis are lacking in three major respects: modeling of the network, backwards compatibility, and incremental deployment. Unrealistic assumptions regarding the network model that are used in existing approaches makes their practical application questionable. The dynamic-optimization solution we outlined in this chapter addresses this concern, allowing it to operate despite irregular and dynamic wireless coverage areas. In our approach, we also isolated the need for an infrastructure-based solution as a critical requirement for backwards compatibility, and proposed techniques to support this requirement. To support incremental deployment, we showed how utility theory can be applied to cater to any set of performance objectives. We believe that these three features fulfill the necessary requirements to building rapidly deployable real-world infrastructure networks that provide good performance.

To address RF interference in real-world deployments, we proposed an experimental approach that serves to provide root-cause analysis into the true nature of RF interference. This technique provides the following key features. It does not assume uniform/static coverage areas for interference detection. It captures interference scenarios where the interfering source may not be identifiable. It does not require any client-side modifications. Finally, given appropriate network support, it also does not incur significant overheads for detecting interference. In our evaluation, we proved the viability of this technique in accurately detecting different scenarios of RF interference.

For our dynamic-optimization solution, we extract two key ideas from the solution proposed for static optimization. First is the use of utility theory to capture system objectives. Second is the idea of separation when it comes to performing channel assignment and power control, as was illustrated in the geometric model presented in Chapter 4.

Many existing solutions we analyzed approached the self-management problem with distinctly different objectives. As a result, each solution catered to a particular performance requirement, e.g., providing realtime support, non-realtime support, load balancing, etc. [18] We view all of these objectives in a single unified utility model. Utility functions provide an abstraction of system performance that can capture any type of objective, and moreover, multiple objectives. This encourages heterogeneous network deployment, where for different regions in the deployment, we can specify different optimization objectives. We envision that our architecture provide an interface to the network administrator for specifying high-level objectives. This interface could also present a visual floor plan of the deployment space, which the administrator can annotate using different parameters and preference weights, depending on deployment requirements. Our architecture can then compute the correct set of objectives for each sub-region and optimize them independently. This approach encourages system evolution where catering to problems such corporate restructuring is a simple matter of re-annotating regions of the deployment.

Utility functions can also support a variety of optimization techniques. Better optimization techniques in the future can replace existing ones in the architecture. Because the general goal of optimization is to maximize utility, any technique that fulfills this requirement can be applied. Additionally, we argue that although our architecture supports both local tuning and global coordination, coordination techniques yield better solutions as they do not get stuck in local minima. In this thesis, we provide examples of two simple coordinated optimization techniques for channel assignment and power control. We show, through evaluation, that for many scenarios, coordinated techniques provide a better solution than local tuning.

The second key idea we extract from static optimization is the notion of separability in optimization. We use this idea in our architecture based on two important insights. First, as pointed out earlier, both channel assignment and power control are tightly coupled. A particular solution to channel assignment can affect the solution for power control and vice versa. This leads us to believe that there likely will not be an optimal solution if both parameters are optimized in parallel. Take the simple example where we first perform power control, given a particular channel configuration. Once power control is complete, the channel configuration will

no longer be optimal with respect to the current power configuration. Thus, we should consider optimizing channel assignment. Once that is complete, the reverse case is true for power control and this process repeats forever. Based on this trivial analysis of the problem, it is unlikely that we will converge to an optimal solution if both parameters are altered in parallel.

Our second consideration in the optimization procedure is the idea of performing channel assignment first, and then power control. We consider this order in the optimization process because channel assignment is a more disruptive process than power control. Power control can be performed more frequently because unlike channel assignment, clients do not need to re-tune their radios to cater to power alterations at the access point. We first perform channel assignment based on a conservative maximum-power conflict graph (MPSC), and then perform power control on top of that. This isolates the channel-assignment algorithm from the technique used for power control as power control can only improve utility by reducing the number of conflicts. This induces separability as well as stability for the channel assignment. Our evaluation indicates that even this conservative approach to channel assignment performs well in comparison to local tuning.

# Chapter 6

# Conclusions

This thesis proposes an architecture for self-management of wireless infrastructure networks. Self-management is a key requirement for modern-day wireless networks that suffer from poor performance. We show that traditional *static optimization* methods are insufficient to meet the performance goals of these networks. We also show that existing *dynamic optimization* systems lack the essentials required to construct rapidly deployable self-managing wireless infrastructure. Our contribution is a solution that is incrementally deployable, backwards compatible, and amenable to real-world deployments. We summarize our contributions as follows.

1. We propose an infrastructure-based solution that does not require client-side modifications. This allows our architecture to support backwards compatibility.

2. We employ a utility model that provides a unified framework for capturing any performance objective, and even multiple objectives. This supports incremental deployment.

3. We illustrate an extension of the general conflict graph (called the Annotated Conflict Graph or ACG) to conveniently represent system utility.

4. We propose a novel experimental approach for detecting different scenarios of RF interference accurately. This method works despite irregular and dynamic coverage areas.

5. We make use of a centralized architecture to support global optimization and access-point coordination, which provides better results than local tuning.

As a result, we propose some heuristics for optimal channel assignment and power control.

6. We propose a dynamically re-configurable framework that can refine configurations in response to the changing RF environment. Techniques for inferring and detecting change are proposed.

7. We present preliminary simulation results to validate some design and performance aspects of our solution.

# Chapter 7

# Future Work

The architecture we propose in this thesis provides a basis upon which many avenues for future work can be explored. We briefly discuss some of them further.

## 7.1 Evaluation

- *Extensive Simulations:* Our preliminary results examine only one aspect of the architecture, i.e., its optimization structure. For a better appreciation of the benefits and scalability of our methods, we must analyze the behavior of the system in dynamic environments. We are currently constructing scenarios for this purpose and plan to present the results of our findings in the near future. Additionally, for our results, we explored only a few basic metrics for performance analysis, i.e., client throughput distribution, and mean client throughput. Other metrics of interest may be the percentage reduction in the number of conflicts, interference reduction measured as a reduction in the number of packet collisions, etc. More sophisticated topologies and the effect of different traffic types also require consideration.

- *Prototype Implementation:* For a thorough evaluation of the system, we are also in the process of building an actual prototype on which to conduct tests. In particular, we are interested in exploring how well our interference-detection experiments work in practice. For this purpose, we are setting up a testbed in the Tetherless Computing Lab, at the University of Waterloo. This testbed consists of four access points and a few iPAQ PDAs. We are using a variety of methods to create a controlled environment for conducting

tests, including restricting access-point coverage areas to within a few meters of the access point. For instance, we are using copper tapping on access point antennas to increase signal attenuation and reduce coverage. The next steps for this testbed involve actually implementing the different pieces of our architecture, for the access points as well as the central controller. We hope to report findings on our implementation and experiences in the near future.

## 7.2 Extensions

There are numerous extensions that can also be applied to our basic architecture.

- *Scheduling Experiments:* Scheduling interference experiments is critical in reducing the overhead of the system. We proposed preliminary techniques for this that involved using information already known about the conflict graph. Nevertheless, more sophisticated scheduling mechanisms may be worth exploring to further reduce the number of experiments that need to be performed, both in cases when conflicts need to be computed during a re-computation, and also when incremental updates need to be made to the annotated conflict graph.

- *Sophisticated Algorithms:* Our initial efforts in developing channel-assignment and power-control algorithms focused on relatively straightforward optimization techniques. This was done to illustrate the benefits of our architecture and show that even simple algorithms yield good solutions. However, we have yet to explore more sophisticated solutions that may be more applicable to our problem. We are currently collaborating with different research groups towards this effort.

- *Additional Tuning Knobs:* Other tuning knobs are also worth exploring to improve performance. For instance, time-slotted access-point activation, described in Chapter 2, could be explored in conjunction with channel assignment. This combination has the capability of constructing a zero-interference solution. In this case, channel assignment can be performed first as is currently proposed in our solution. Once that is complete, we can analyze the resulting conflict graph to generate a schedule for activating access points such that no two access points or their clients that conflict are activated at the same time. This effectively reduces the number of conflicts to zero. Power control could also be used with this solution to generate a more relaxed AP activation schedule. We intend to explore this technique in the future.

- *Additional Functionality:* We also have not currently explored all the techniques we proposed for our architecture. For instance, the utility function that we evaluated did not take into account load balancing across access points. Furthermore, load balancing using power control also requires that the algorithm ensure that clients are not disconnected in the process. Therefore, determination of neighbouring APs to which clients can be migrated needs to be supported. We have devised preliminary protocols to do this and intend to incorporate them in our architecture.

- *Incorporating Client Feedback:* The focus of our work has been on developing an infrastructure-based solution. However, this does not restrict us from receiving feedback from clients, if they support this feature. By receiving such information, we would be able to relax some requirements for the system and also improve its accuracy. For instance, if clients report information on neighbouring access points they can hear, this could reduce the number of experiments that need to be performed for these clients. Other client statistics such as client load can also aid in computing utilities more accurately. By combining feedback mechanisms with our current architecture, we would then be able to support both upgraded and legacy clients.

- *Additional Self-Management Capabilities:* This thesis has focused on enhancing wireless network performance. However, there may be other mechanisms that we also desire from a self-managing system. For instance, fault diagnosis is also an important requirement, to support reliability in a wireless deployment. Some recent work has explored this problem in the context of wireless LANs and these ideas can be extended to our architecture [28].

# Bibliography


[1] Control And Provisioning of Wireless Access Points (CAPWAP). http://www.ietf.org/html.charters/capwap-charter.html.

[2] Engim Delivers Simultaneous Multi-Channel WLAN Switching Engine Silicon. http://www.tmcnet.com/enews/041103j.htm.

[3] LinksysInfo.org - The #1 Source in Linksys Information. http://www.linksysinfo.org/modules.php?name=Forums&file=viewtopic&t=769.

[4] Propagate Inc., AutoCell - The Self-Organizing WLAN. http://www.propagatenet.com/resources/index.html.

[5] Radio Resource Measurement Enchancements, IEEE 802.11k Meeting Update. http://grouper.ieee.org/groups/802/11/Reports/tgk_update.htm.

[6] Ruckus Wireless Inc., MIMO and Smart Antenna Techniques for 802.11a/b/g Networks. http://www.ruckuswireless.com/technology/whitepapers/mimo/.

[7] Spectrum Managed 802.11a, IEEE 802.11h Meeting Update. http://grouper.ieee.org/groups/802/11/Reports/tgh_update.htm.

[8] Taipei's Mobile City Project Selects Nortel Wireless Mesh Network Solution. http://www.nortel.com/corporate/news/newsreleases/2004d/11_17_04_taipei_city.html.

[9] Wireless Network Management, IEEE 802.11v Meeting Update. http://grouper.ieee.org/groups/802/11/Reports/tgv_update.htm.

[10] Wireless Philadelphia Executive Committee. http://www.phila.gov/wireless/.

[11] Airespace Inc., The Evolution of the Enterprise-Class Wireless LAN Access Point. White Paper, 2004.
http://www.airespace.com/pdf/WP_Enterprise_Class_WLAN_AP.pdf.

[12] Aruba Networks Inc., Advanced RF Management for Wireless Grids. Technical Brief, 2004.
http://www.arubanetworks.com/pdf/rf-for-grids.pdf.

[13] Cisco Aeronet Access Point Hardware Installation Guide (for 340 and 350 Series), 2004.
http://www.cisco.com/application/pdf/en/us/guest/products/ps447/c2001/ccmigration_09186a00800eec5f.pdf.

[14] AirMagnet Inc., AirMagnet Enterprise 6.0 DataSheet, 2005.
http://www.airmagnet.com/products/assets/Enterprise6_DataSheet.pdf.

[15] Anritsu MS2687B Spectrum Analyzer, product catalog, 2005.
http://www.us.anritsu.com/downloads/files/MS2687B_E1400.pdf.

[16] D-Link Wireless 108G MIMO Router (DI-634M), 2005.
http://www.dlink.com/products/?pid=458.

[17] Extricom Inc., Wireless LAN Switch DataSheet, 2005.
http://www.extricom.com/imgs/Uploads/PDF/SWDataSheet.pdf.

[18] Introducing 802.11v - A hope for Wi-Fi management, 2005.
http://www.techworld.com/networking/features/index.cfm?featureid=1135&Page=1&pagePos=14.

[19] Meru Networks Inc., Virtual Cells: The Only Scalable MultiChannel Deployment. White Paper, 2005.
http://www.merunetworks.com/pdf/Virtual_Cells_WP4_0705.pdf.

[20] Motorola Inc., LANPlanner Suite, 2005.
http://www.motorola.com/Enterprise/contentdir/en_US/Enterprise/Files/LANPlanner.pdf.

[21] Star Wars NASCAR? Get Ready for rocket racing, 2005.
http://www.cnn.com/2005/TECH/space/10/04/rocket.race/.

[22] Trapeze Networks Inc., RingMaster Datasheet, 2005.
http://www.trapezenetworks.com/products/datasheets/rm/rm.asp.

[23] Vivato Inc., Vivato Indoor WiFi Base Stations, Product Catalog, 2005. http://www.vivato.net/.

[24] WildPackets Inc., Omni Wireless Sensor Datasheet, 2005. http://www.wildpackets.com/elements/omni_wireless_sensor/Omni_Wireless Sensor.pdf.

[25] WildPackets Inc., RF Grabber Datasheet, 2005. http://www.wildpackets.com/elements/rfgrabber/RFGrabber.pdf.

[26] Wireless-G Access Point with SRX, 2005. http://www.linksys.com/.

[27] Xirrus Inc, WLAN Array Architecture. 2005. http://www.xirrus.com/public/pdf/Xirrus_WLAN_Array_Architecture.pdf.

[28] Atul Adya, Paramvir Bahl, Ranveer Chandra, and Lili Qiu. Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom)*, pages 30–44, Philadelphia, PA, USA, 2004.

[29] Nabeel Ahmed and Srinivasan Keshav. A successive refinement approach to wireless infrastructure network deployment. *To Appear in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2006).*

[30] R. Ahola, A. Aktas, J. Wilson, K.R. Rao, F. Jonsson, I. Hyyrylainen, A. Brolin, T. Hakala, A. Friman, T. Makiniemi, J. Hanze, M. Sanden, D. Wallner, Yuxin Guo, T. Lagerstam, L. Noguer, T. Knuuttila, P. Olofsson, and M. Ismail. A single-chip CMOS transceiver for 802.11a/b/g wireless LANs. *IEEE Journal of Solid-State Circuits*, 39:2250–2258, 2004.

[31] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-management in chaotic wireless deployments. In *Proceedings of the 11th annual international conference on Mobile computing and networking (MobiCom)*, pages 185–199, Cologne, Germany, 2005.

[32] Devin Akin. *Certified Wireless Network Administrator (CWNA) Official Study Guide (Exam PW0-100).* McGrawHill/Osborne, Berkeley, California, 2003.

[33] Anmol Sheth and Richard Han. An implementation of transmit power control in 802.11b networks. Technical Report CU-CS-934-02, Department of Computer Science, University of Colorado, Boulder, CO, USA, 2002.

[34] Francois Baccelli, Bartomiej Blaszczyszyn, and Florent Tournois. Spatial averages of coverage characteristics in large cdma networks. *Wireless Networks*, 8(6):569–586, 2002.

[35] Victor Bahl, Jitendra Padhye, Lenin Ravnindranath, Manpreet Singh, Alec Wolman, and Brian Zill. DAIR: Managing enterprise wireless networks using desktop infrastructure. In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, MD, USA, 2005.

[36] Anand Balachandran, Geoffrey M. Voelker, and Paramvir Bahl. Wireless hotspots: Current challenges and future directions. In *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and services on WLAN Hotspots (WMASH)*, pages 1–9, San Diego, CA, USA, 2003.

[37] Yigal Bejerano and Randeep Bhatia. MiFi: A Framework for Fairness and QoS Assurance in Current IEEE 802.11 Networks with Multiple Access Points. In *23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1229–1240, 2004.

[38] Yigal Bejerano and Seung-Jae Han. Cell breathing techniques for balancing the access point load in wireless LANs. 2006. *To Appear in Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*.

[39] John Bicket. Bit-rate selection in wireless networks. Technical report, Cambridge, MA, USA, 2005. Master's Thesis, Massachusettes Institute of Technology (MIT).

[40] Ralf Borndorfer, Andreas Eisenblatter, Martin Grotschel, and Alexander Martin. Frequency assignment in cellular phone networks. In *Annals of Operations Research*, volume 76, pages 73–93, 1998.

[41] A. Bouju, J. F. Boyce, C. H. D. Dimitropoulos, G. Vom Scheidt, and J. G. Taylor. Tabu search for the radio links frequency assignment problem. In *Applied Decision Technologies (ADT)*, London, United Kingdom, 1995.

[42] D. J. Castelino, S. Hurley, and N. M. Stephens. A tabu search algorithm for frequency assignment. In *Annals of Operations Research*, volume 63, pages 301–319, 1996.

[43] Ranveer Chandra, Victor Bahl, Pradeep Bahl, and Ken Birman. VirtualWiFi: Connecting to mutliple IEEE 802.11 networks with one WiFi card. http://research.microsoft.com/netres/projects/virtualwifi/default.htm.

[44] David D. Clark, Craig Partridge, J. Christopher Ramming, and John T. Wroclawski. A knowledge plane for the internet. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pages 3–10, Karlsruhe, Germany, 2003.

[45] Mark de Berg, Otfried Schwarzkopf, Marc Van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications (Second Edition)*. Springer-Verlag New York, 2000.

[46] John Doyle, Bruce Francis, and Allen Tannenbaum. *Feedback Control Theory*. MacMillan Publishing Co., 1990.

[47] Andreas Eisenblatter, Martin Grotschel, and Arie M.C.A. Koster. Frequency planning and ramifications of coloring. In *Discussiones Mathematicae Graph Theory*, volume 22, pages 51–88, 2002.

[48] Steven J. Fortune, David M. Gay, Brian W. Kernighan, Orlando Landron, Reinaldo A. Valenzuela, and Margaret H. Wright. WISE design of indoor wireless systems: Practical computation and optimization. In *IEEE Computer Society - IEEE Computational Science and Engineering*, pages 58–68, 1995.

[49] Andrew Glassner. *An Introduction to Ray Tracing*. San Diego, California, 1989. Academic Press.

[50] Piyush Gupta and P. R. Kumar. Capacity of wireless networks. Technical report, University of Illinois, Urbana-Champaign, 1999.

[51] Magnus M. Halldarsson, Joseph Y. Halpern, Li (Erran) Li, and Vahab S. Mirrokni. On spectrum sharing games. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing (PODC)*, pages 107–114, St. John's, Newfoundland, Canada, 2004.

[52] Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Performance anomaly of 802.11b. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, San Francisco, USA, March-April 2003.

[53] Alex Hills and Bob Friday. Radio resource management in wireless LANs. *IEEE Communications Magazine*, 42(12):9–14, 2004.

[54] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of*

*the 9th annual international conference on Mobile computing and networking (MobiCom)*, pages 66–80, San Diego, CA, USA, 2003.

[55] Ravi Jain, Dan Lelescu, and Mahadevan Balakrishnan. Model T: An empirical model for user registration patterns in a campus wireless lan. In *Proceedings of the 11th annual international conference on Mobile computing and networking (MobiCom)*, pages 170–184, Cologne, Germany, 2005.

[56] Jeannette M. Janssen. *Channel Assignment and Graph Labelling*. John Wiley and Sons, 2002.

[57] B. Jaumard, O. Marcotte, and C. Meyer. Mathematical models and exact methods for channel assignment in cellular networks. pages 239–255, 1996.

[58] I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. In *IEEE Personal Communications*, volume 3, pages 10–31, 1996.

[59] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, pages 126–134, Venice, Italy, 2004.

[60] W. K. Lai and George G. Coghill. Channel assignment through evolutionary optimization. In *IEEE Transactions on Vehicular Technology*, volume 45, pages 91–96, 1996.

[61] K. K. Leung and B. "J" Kim. Frequency assignment for IEEE 802.11 wireless networks. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pages 1422–1426, Beijing, China, 2003.

[62] D. Maniezzo, M. Cesana, and M. Gerla. IA-MAC, Interference Aware MAC. Technical Report 020037, University of California Los Angeles (UCLA), 2002.

[63] Ignacio Mas, Hector Velayos, and Gunnar Karlsson. Distributed admission control for wireless LANs. In *Winternet Grand Finale Workshop*. KTH, Royal Institute of Technology, Stockholm Sweden, 2005.

[64] Dave Mills. Network Time Synchronization Project. http://www.eecis.udel.edu/ mills/ntp.html.

[65] Dave Mills. Pulse-Per-Second (PPS) Signal Interfacing. http://www.eecis.udel.edu/ mills/ntp/html/pps.html.

[66] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for WLANs. *SIGMOBILE Mobile Computing Communications Review*, 9(3):19–31, 2005.

[67] Arunesh Mishra, Vladimir Brik, Suman Banerjee, Aravind Srinivasan, and William Arbaugh. A client driven approach for channel management in wireless LANs. 2006. *To Appear in 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM).*

[68] Allen Miu, Godfrey Tan, Hari Balakrishnan, and John Apostolopoulos. Divert: Fine-grained path selection for wireless LANs. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys)*, pages 203–216, Boston, MA, USA, 2004.

[69] Lata Narayanan. *Channel Assignment and Graph Multicoloring.* John Wiley and Sons, 2002.

[70] Stefanie Olson. Wi-Fi Mosquito Killer coming to a porch near you, 2005. http://news.com.com/Wi-Fi+mosquito+killer+coming+to+a+porch+near+you/2100-11395_3-5961535.html.

[71] Jitendra Padhye, Sharad Agarwal, Venkata N. Padmanabhan, Lili Qiu, Ananth Rao, and Brian Zill. Estimation of link interference in static multi-hop wireless networks. In *USENIX - Internet Measurement Conference (IMC)*, pages 305–310, Berkeley, CA, USA, 2005.

[72] D. Qiao, S. Choi, A. Jain, and K. Shin. Adaptive transmit power control in IEEE 802.11a wireless LANs. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pages 433–437, Beijing, China, 2003.

[73] Lili Qiu, Paramvir Bahl, Ananth Rao, and Lidong Zhou. Fault detection, isolation, and diagnosis in multi-hop wireless networks. Technical Report TR-2004-11, Microsoft, 2003.

[74] Ishwar Ramani and Stefan Savage. SyncScan: Practical fast handoff for 802.11 infrastructure networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, San Diego, CA, USA, 2005.

[75] Ashish Raniwala and Tzicker Chiueh. Deployment issues in enterprise wireless LANs. Research Proficiency Report, Department of Computer Science, Stony Brook University, 2003.

[76] Ashish Raniwala, Kartik Gopalan, and Tzicker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mobile Computing Communications Review*, 8(2):50–65, 2004.

[77] Dipankar Raychaudhuri and Xiangpeng Jing. A spectrum etiquette protocol for efficient coordination of radio devices in unlicensed bands. In *Proceedings of 14th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 172–176, Beijing, China, 2003.

[78] H. G. Sandalidis and P. Stavroulakis. *Heuristics for Solving Fixed-Channel Assignment Problems*. John Wiley and Sons, 2002.

[79] Anmol Sheth and Richard Han. SHUSH: Reactive transmit power control for wireless MAC protocols. In *First IEEE International Conference on the Wireless Internet (WICON)*, pages 18–25, 2005.

[80] Steve Thomas. Detect interference in wireless nets, 2005. Anritsu Corp.

[81] Arunchandar Vasan, Ramachandran Ramjee, and Thomas Woo. ECHOS - Enhanced capacity 802.11 hotspots. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1–9, San Diego, CA, USA, 2005.

[82] C. Voudouris and E.P.K. Tsang. Solving the radio link frequency assignment problem using guided local search. In *Proceedings, NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Aalborg, Denmark, 1998.

[83] Y. Wang, L. Cuthbert, and J. Bigham. Intelligent radio resource management for IEEE 802.11 WLAN. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1365–1370, Atlanta, Georgia, USA, 2004.

[84] David Wetherall. Management chaotic wireless networks. Talk Delivered at Microsoft Research, Self-Managing Networks Summit, 2005.

[85] Jason White. Wireless technology changing work and play, 2005. http://www.cnn.com/2005/TECH/10/17/wireless.overview/index.html.

[86] G. Wolfle, F. M. Landstorfer, R. Gahleitner, and E. Bonek. Extensions to the field strength prediction technique based on dominant paths between transmitter and receiver in indoor wireless communications. In *2nd European Personal*

*and Mobile Communications Conference (EPMCC)*, pages 26–36, Bonn, Germany, 1997.

[87] Jihwang Yeo, Moustafa Youssef, Tristan Henderson, and Ashok Agrawala. Characterizing IEEE 802.11 traffic: The wireless side. Under review, Department of Computer Science, University of Maryland, 2005.

[88] J. A. Zoellner and C. L. Beall. A breakthrough in spectrum conserving frequency assignment technology. In *IEEE Transactions on Electromagnetic Compatiblity*, volume 19, pages 313–319, 1977.

[89] A. Zomaya, A. Smith, and F. Seredynski. On the use of the simulated annealing algorithm for channel allocation in mobile computing. *Wireless Communications and Mobile Computing*, 3(2):239–253, 2003.