Interference Management in Dense 802.11 Networks

by

Nabeel Ahmed

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2009

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 networks are growing at a phenomenal rate. This growth is causing an overcrowding of the unlicensed RF spectrum, leading to increased interference between co-located devices. Existing decentralized medium access control (MAC) protocols (e.g. IEEE 802.11a/b/g standards) are poorly designed to handle interference in such dense wireless environments. This is resulting in networks with poor and unpredictable performance, especially for delay-sensitive applications such as voice and video.

This dissertation presents a practical *conflict-graph* (CG) based approach to designing self-organizing enterprise wireless networks (or WLANs) where interference is centrally managed by the network infrastructure. The key idea is to use potential interference information (available in the CG) as an input to algorithms that optimize the parameters of the WLAN. We demonstrate this idea in three ways. First, we design a self-organizing enterprise WLAN and show how the system enhances performance over non-CG based schemes, in a high fidelity network simulator. Second, we build a practical system for conflict graph measurement that can precisely measure interference (for a given network configuration) in dense wireless environments. Finally, we demonstrate the practical benefits of the conflict graph system by using it in an optimization framework that manages associations and traffic for mobile VoIP clients in the enterprise.

There are a number of contributions of this dissertation. First, we show the practical application of conflict graphs for infrastructure-based interference management in dense wireless networks. A prototype design exhibits throughput gains of up to 50% over traditional approaches. Second, we develop novel schemes for designing a conflict graph measurement system for enterprise WLANs that can detect interference at microsecond-level timescales and with little network overhead. This allows us to compute the conflict graph up to 400 times faster as compared to the current best practice proposed in the literature. The system does not require any modifications to clients or any specialized hardware for its operation. Although the system is designed for enterprise WLANs, the proposed techniques and corresponding results are applicable to other wireless systems as well (e.g. wireless mesh networks). Third, our work opens up the space for designing novel fine-grained interference-aware protocols/algorithms that exploit the ability to compute the conflict graph at small timescales. We demonstrate an instance of such a system with the design and implementation of an architecture that dynamically manages client associations and traffic in an enterprise WLAN. We show how mobile clients sustain uninterrupted and consistent VoIP call quality in the presence of background interference for the duration of their VoIP sessions.

Acknowledgements

This dissertation and all my accomplishments draw upon the contributions of many whom I wish to thank. First and foremost I thank Almighty God, for giving me the strength, courage, and ability to successfully accomplish this task. Second, I am deeply grateful to my family for their tireless efforts and continuous support in whatever path I chose to take in life. To my father, for motivating me to excel in my work and always ensuring I have the necessary resources at my disposal to successfully accomplish my goals. To my mother, for instilling in me a strong character and helping me lead a righteous life. Finally, I am deeply thankful to my sister and brother-in-law for being there for me through difficult times. My life has been truly enriched by their presence.

I came to Waterloo almost five years ago without much of a clue on what I was going to do. I am deeply grateful to my supervisor, Keshav, for helping mold me into the strong researcher I am today. Keshav not only guided me in technical matters but also helped me to develop the right philosophy for doing research, which was instrumental in helping me establish a strong research vision. I will always cherish his amusing anecdotes, that have coloured not only my professional but also my personal perceptions on life. I must also thank him for allowing me to pursue my own problems of interest, while at the same time providing me with the necessary resources to ensure I succeed in my work. Throughout this journey, he has kept me focused and always reminded me to be mindful of the bigger picture in my work.

Dina Papagiannaki, my mentor at Intel Research, has for all practical purposes, been my second advisor. She has been a constant source of inspiration and has not only provided strong research advice but also helped me overcome the technical challenges in my work. I am also deeply grateful to her for increasing the visibility of my work beyond research circles and into the industry.

I would also like to thank my collaborators at the University of Wisconsin-Madison, including Suman, Vivek, Shravan, Sharad, and Arunesh for giving me an opportunity to share my research experience with them. Suman Banerjee, my mentor at UW Madison, has been a constant source of motivation during difficult times, which has allowed the group to succeed in various ways. I am also grateful to Vivek for putting up with me during the frustrating late night hours preceding paper submissions. We have shared in many such experiences throughout our graduate student lives.

I would like to thank Tim Brecht, Paul Ward, Sherman Shen, and Jitendra Padhye for serving on my thesis committee. Their valuable feedback and suggestions have helped to greatly improve the quality of this dissertation. Tim and Paul have been on the committee from the beginning and have been a constant source of guidance in helping me

align my research goals. Tamer Ozsu also deserves special thanks for believing in me as a researcher and preparing numerous recommendation letters in support of various scholarship and job applications.

My accomplishments would not have been possible without the support of my office mates, including Adi, David, Sumair, Mohammed, Shimin, Earl, Hossein, Usman, Darcy, and Majid. Adi has been with me from the beginning. As such, he has been both a dear friend and inspiring colleague with whom I have shared innumerable grad life experiences. Our stimulating late night discussions on life, random surge of ideas, and numerous consolation sessions during difficult times have forged a friendship and bond that will stand the test of time. I have yet to find a person as motivated, energetic and selfless as Adi. I am glad to have had the opportunity to share these experiences with him.

I also would like to thank Trevor, Ronaldo, Lawrence, and others at CSCF for their constant cooperation and willingness to help in both technical matters as well as assist in the smooth deployment of the wireless testbed. My thanks also go out to Margaret, Gail, Helen, and Jessica for helping in administrative matters and requests, especially those that were made on short notice. Without their support and cooperation, none of this work would have been possible.

Life is not just about work. There are a number of friends with whom I have shared this experience as well. I will only be able to name a few. These friends have not only provided for me a departure from work, but also made my little over five years in Waterloo very memorable. Umar Farooq and I have shared many experiences together, from watching crazy movies at the theater to jumping off planes at 10,500 feet! More importantly, I could always count on Umar to be there for me during difficult times (especially impending deadlines). He has acted not only as a friend but also a brother to me. Nomair and Mariam, my friends with whom I share a long history have also enriched my life in numerous ways. Their warm, affectionate, and ‘go the extra mile’ attitude, coupled with the fun moments I have enjoyed in their company will forever be cherished. They are a testament to what friendship is really all about. Omar Zia (a.k.a. Guru) and Kamran have been with me from the very beginning. As such, I share with them a special bond and friendship that I will cherish for many years to come. I cannot go without mentioning the wonderful parties Omar and Farheen have hosted and the delectable food that I thoroughly enjoyed, followed by late night rants and discussions on politics. Omar, Farheen, and Kamran are part of a larger ‘gang’ (including Usman, Nabeel Butt, Sajjad, Javeria, Raqib, Mohsin and Uzma) with whom I’ve enjoyed numerous camping and hiking trips as well as other unforgettable experiences. This gang has played an integral role in helping me accomplish my tasks. I cannot go without mentioning another dear friend, Omer

Beg, who has been around with me since my undergraduate days. Aside from sharing in academics, Omer Beg has played a vital role in the spiritual aspects of my life. As a friend and brother, I am truly grateful to him for keeping me in good company and always reminding me of the importance of keeping the ultimate goal in mind. I also wish to thank the many friends (including roommates) I've known through the Udai student organization, including Raiju, Bala, Srinath, Naveen, Sarvagya, Akhilesh, and Maheedhar. They have influenced my life in more ways than one and I will forever remember the good times we've shared together. The list of friends goes on and I wish to thank all of those who I have and have not mentioned for sharing in this wonderful journey that has finally come to an end.

In the end, I would also like to extend my deepest gratitude to my undergraduate advisor, Mike Dahlin, for introducing me to the wonderful world of research. Mike's encouragement, sincerity, and willingness to help was the right spark I needed to land an enjoyable career in research. I am blessed to have had an opportunity to work with him as an undergrad at UT Austin.

Dedication

This thesis is dedicated to my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networks are experiencing unprecedented growth and gradually becoming the dominant means by which people access the Internet. Last year alone, there were over 387 million WiFi devices sold around the world and this number is expected to increase in the future [16]. Moreover, WiFi technology is being used in a variety of different settings, from home and enterprise networks to city-wide wireless mesh networks (WMNs). This ubiquitous use of WiFi is also spurring the growth of the smartphone market, which are phones that are typically equipped with multiple interfaces such as Bluetooth, WiFi, and GSM, to name a few [106].

Despite this growth, the amount of available unlicensed RF spectrum has remained unchanged [1] . As we can see in Figure 1.1 (which shows the FCC's spectrum allocation in the US), the unlicensed bands (marked in the figure) constitute a very small fraction of the entire RF spectrum, where unlicensed WiFi devices must operate. This fixed allocation has lead to a scarcity of the RF spectrum, where more and more devices must share these unlicensed frequency bands for communication. Without properly designing protocols that facilitate sharing of the RF spectrum, WiFi devices can potentially experience poor performance due to *RF interference*. RF interference occurs when two or more devices simultaneously transmit on the wireless channel, causing collisions between wireless signals at the receiver. This makes it difficult for the receiver to correctly recover the bits transmitted by the sender. With the projected growth of WiFi technology in the upcoming years, RF interference is likely to become a major barrier to the performance of wireless networks that tout broadband speeds for wireless users, especially as the density increases.

In addition to the growth of the WiFi market, emerging applications such as voice

[1] Due to demand, the FCC only recently allocated the 60 GHz frequency for unlicensed use

Figure 1.1: Allocation of RF Spectrum by the Federal Communications Commission (FCC) in the US

and video are also placing additional demands on such wireless networks. The ability to stream high definition video at home and on-the-go while simultaneously transferring large files over the network requires an abundance of bandwidth that existing WiFi networks fail to provide. In addition, the delay-sensitive nature of voice and video applications makes the delivery of such content even more challenging for such WiFi networks. To enable such applications, RF interference must be systematically addressed.

The first IEEE 802.11 (WiFi) standard was drafted in 1999 and since then, it has been implemented universally by WiFi chip manufacturers. The designers of the IEEE 802.11 standard likely never expected the exponential growth of WiFi technology that is being seen today. As a result, the original design of the standard made many simplifying assumptions with respect to medium access control. In particular, all devices contend independently for channel access, without any explicit coordination amongst each other. Such decentralized techniques work well for a small number of users, but fail in dense networking environments that contain hundreds of users [2] [73]. Having realized these shortcomings, WiFi architects are moving towards more *managed* and *coordinated* designs. In addition, IEEE standards bodies are also playing their part by devising standards such as IEEE 802.11v and 802.11k to facilitate better management and coordination between WiFi devices.

[2] A common occurrence in many enterprise wireless networks

WLANs	HP Labs	Seoul National University	Our Testbed
Exposed Terminals	39%	9%	39%
Hidden Terminals	43%	70%	35%

Table 1.1: Percentage of links that experience interference across different research/industrial WLAN testbeds

While WiFi technology has been used in many different settings (e.g., home, enterprise, and metro-scale wireless mesh networks), among its more popular applications are enterprise networks. In an enterprise network, access points (APs) are deployed throughout an office (or campus) to provide blanket coverage for wireless access. Enterprise networks (or WLANs) embody a unique set of challenges because of user density and the dynamics of indoor environments (for example, due to people moving about in the building). Moreover, use of such networks in meeting room and libraries create pockets of heavy usage where traffic load can also impact user experience. Moreover, emerging applications such as voice and video require uninterrupted service despite the presence of radio interference from other WiFi devices. Table 1.1 shows the percentage of links that suffer from hidden and exposed terminal interference (discussed in greater detail in Chapter 2) for different enterprise-scale wireless testbeds. Finally, non-802.11 devices transmitting on the same frequency also cause interference. These challenges make combating RF interference in enterprise WLANs a difficult challenge.

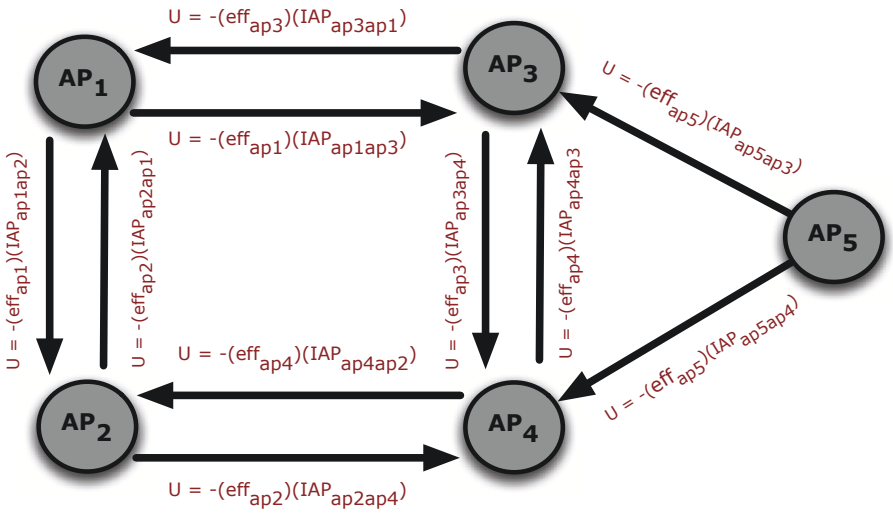

Figure 1.2: An example of a conflict graph

Motivated by these challenges, this dissertation proposes techniques to *systematically* address RF interference in centralized enterprise WLANs. Centralized enterprise

WLANs (described in greater detail in Chapter 3) are networks where the APs relinquish management functionality to a central controller that manages the configuration of the APs [3]. We propose a *conflict graph* based approach to model interference, develop novel techniques to measure the conflict graph, and apply it to optimize the performance of enterprise networks. Conflict graphs (first proposed in [121]) encode interference information between wireless links. An example conflict graph is shown in Figure 1.2, where the nodes in the graph represent APs and an edge exists between two nodes if the AP from which the edge emanates interferes with the AP at which the edge terminates. The values on the edges of the conflict graph describe the impact of interference on the AP experiencing interference (described in greater detail in Chapter 4). The conflict graph provides a way to *globally* model interference, allowing the design of centralized algorithms that can potentially drive the network configuration to the global optimum of the network. In contrast, decentralized algorithms do not require network coordination and optimize configurations based on local information at each individual AP. We show in Chapter 4 that such techniques lead to sub-optimal performance for wireless clients.

1.1 Scope and Goals

This dissertation bridges the gap between the *theory* of conflict graphs and their *practical* application to enterprise WLANs. It has the following goals:

- **Rapid Detection of RF Interference:** In the future, WiFi clients will be mobile while using the network (for example, users making VoIP calls while on-the-go). In such scenarios, clients may encounter intermittent interference as they move around, causing application performance to degrade. In such cases, it is imperative to rapidly detect interference (at timescales of a few seconds) to allow the network to re-tune it's parameters and ensure reliable service delivery to mobile users.
- **Online Estimation of RF Interference:** In an enterprise WLAN, interference is primarily measured between AP-client links. To perform these measurements, clients must be associated with the network. In real-world deployments, clients come and go and the network continuously undergoes changes in traffic and topology. This necessitates an online approach to estimating RF interference [4].

[3] Note that while we tackle interference in enterprise WLANs, the underlying principles of our work are equally applicable to other managed WiFi networks as well, such as wireless mesh networks (WMNs)

[4] In other systems such as wireless mesh networks, interference may be measured offline or overnight

- **Free from RF Propagation Models:** Many RF propagation models have been proposed to approximate signal propagation in different environments. While these models have been used to estimate interference, they are known to be inaccurate in practical settings [79]. Therefore, techniques that use these models have limited application in real-world deployments. This motivates an approach that is free from RF propagation modeling and estimates interference through (active or passive) measurements.

- **No Client Modifications:** Enterprise WLANs are found in a variety of different settings, from corporate offices to university campuses. To allow widespread deployment in such settings, we require that clients are not modified for interference estimation purposes. This allows the solution to be both incrementally deployable as well as backwards compatible with existing IEEE 802.11 standards.

This dissertation is divided into three parts. In the first part, we *design* an enterprise WLAN architecture (dubbed 'SMARTA') that measures, annotates, and maintains the conflict graph for the network. Using the conflict graph, SMARTA performs frequency selection and power control to maximum network performance (given a particular objective function). This is achieved under the constraints outlined above. The algorithms for channel assignment and power control are shown to provide significant gains in throughput as compared to existing schemes. SMARTA is implemented and tested on QualNet [13], a high-fidelity network simulator.

In the second part, we *implement* SMARTA to gauge its real-world application to enterprise WLANs. To allow for this, we deploy a 38 node centralized wireless testbed. The testbed is deployed in the William Davis Centre (DC) building at the University of Waterloo. We implement SMARTA's interference measurement framework (dubbed 'Micro-Probing') in the driver/firmware of the Intel 2915ABG (Centrino) card. Micro-probing is evaluated against the current state-of-the-art approach for interference estimation [98]. We show that micro-probing achieves the same level of accuracy as the current best approach with two orders of magnitude reduction in measurement overhead.

In the third part of this dissertation, we *apply* micro-probing to the problem of supporting mobile VoIP clients in interference-limited enterprise environments. The system (dubbed 'Overcast') requires that the conflict graph be continuously measured and updated as clients move about in building. Micro-probing is well-suited to this application and we show how using this framework, Overcast provides dependable service to VoIP users even in the presence of co-located backlogged interferers. In other work, we have applied micro-probing to optimize centralized scheduling of data traffic in an enterprise WLAN. The details of this scheme are covered elsewhere [109].

We summarize the four key parts of the thesis next.

1.2 SMARTA

In Chapter 4, we describe the design of a Self-Managing ARchitecture for Thin Access points (SMARTA). This architecture prescribes a set of techniques for measuring a conflict graph for an enterprise WLAN. Using the conflict graph, SMARTA dynamically adjusts both access point channel assignments and power levels to optimize arbitrary objective functions, while taking into account the irregular nature of RF propagation, and working with unmodified legacy clients. We evaluate the SMARTA architecture and show that it is able to provide significant improvements over existing approaches. For example, in a typical scenario, SMARTA can provide 50% more throughput and 40% lower mean per-packet delay than a hand-optimized configuration. Moreover, SMARTA automatically reconfigures channels and power levels in response to both small and large changes in the RF environment due to client movement.

1.3 Testbed Design

In Chapter 5, we describe the details of the testbed platform we designed and built to test our algorithms for centralized control. Centralized enterprise WLANs have a unique set of requirements that prior testbed designs fail to provide. We highlight these requirements and describe the hardware and software design of our testbed. We also benchmark the testbed to ensure that it meets the requirements for centralized control. Finally, we also briefly describe our experiences with using the testbed during the last two years.

1.4 Micro-Probing

In Chapter 6, we present the Micro-Probing interference measurement system. Micro-Probing implements SMARTA's interference measurement framework and addresses the engineering challenges not met by the 'paper design' proposed in Chapter 4. For instance, SMARTA makes assumptions such as: (1) Synchronization between pairwise transmitters during an interference test, (2) Clearing of the air to perform interference tests, and (3) The ability to measure RF spectral energy to detect interference. Micro-probing addresses these requirements and demonstrates the real-world application of SMARTA's interference measurement system. Note that, like SMARTA, Micro-probing

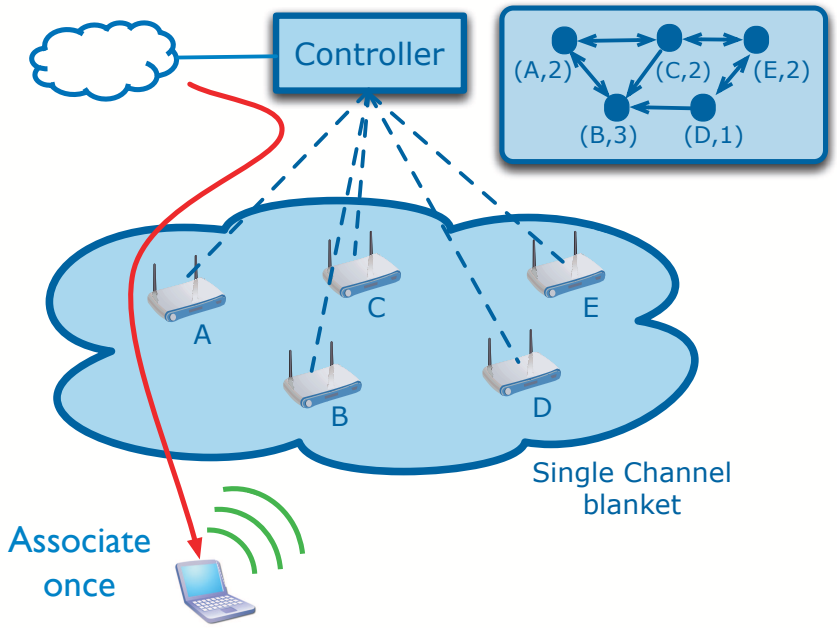

Figure 1.3: High-level view of the Overcast architecture. The client associates only once to the network (through AP A) and the controller seamlessly manages the AP-client link thereafter.

is also fully compliant with existing 802.11 standards and can measure interference for legacy clients.

1.5 Overcast

In Chapter 7, we present the Overcast system that uses conflict graphs to mitigate interference and ensure reliable service for mobile clients operating VoIP sessions. Overcast illustrates an application where real-time computation of the conflict graph is necessary to ensure that the decisions taken by the optimization framework do not degrade network performance. The salient features of Overcast are: (1) A virtual AP architecture to support seamless mobility for VoIP clients, (2) Centralized selection of APs for each client, and (3) Coarse-grained scheduling of conflicting APs. A high-level picture of the Overcast system is shown in Figure 1.3. All APs are configured on the same channel and appear to the client as a single virtual AP. The client associates to the network only once and the controller subsequently decides which AP the client will communicate with (using appropriate selection metrics). The conflict graph (shown on the top right hand side of the figure) plays an integral role in managing interference in this single-channel

WLAN system. Additional details on the Overcast systems are presented in Chapter 7. Overcast is evaluated on the 38 node wireless testbed (described in Chapter 5) and is shown to provide good quality of service (QoS) to mobile VoIP clients. It increases the number of clients supported by the network by a factor of two to three in the presence of background traffic.

1.6 Contributions

The principal contribution of this dissertation is that it bridges the gap between the theory of conflict graphs and their practical application in real-world wireless deployments. The ability to compute conflict graphs at significantly smaller timescales facilitates new innovations in algorithm design and network optimization. This is a clear departure from existing work that assumes conflict graphs require lengthy measurements, making it difficult to re-measure them in an online network. The specific contributions of this thesis are:

- **Novel Interference Measurement Techniques**: We develop novel techniques to *practically* measure the conflict graph for enterprise WLANs [30]. Aside from measuring conflicts, these techniques have broader application beyond conflict graph measurement, such as controlling client transmissions for uplink traffic [71].

- **Application to Network Optimization**: We apply the conflict graph framework to a challenging optimization problem where fine-grained interference information is necessary to meet the performance objectives of the system. We show how the resulting system, with the help of the conflict graph, gracefully manages interference even as the number of contenders increases in the network. Prior techniques cannot be applied either because of their inability to compute the conflict graph on short timescales or their need for client modifications.

- **Evaluation on an Enterprise-scale Wireless Testbed**: We design and deploy an enterprise-scale WLAN testbed (consisting of 38 nodes) in which we implement and evaluate our proposed protocols/algorithms. Because we focus on centralized WLANs, the testbed mimics centralized control and we show how this design ultimately influences our hardware and software choices for the wireless platform [29].

- **Practical Conflict Graphs**: Our work enables the practical application of conflict graphs. Furthermore, by carefully choosing our design constraints, we propose

techniques that can be rapidly deployed in existing WLAN designs [32]. We believe that our work provides opportunities for new and exciting research on enterprise WLAN optimization.

Chapter 2

Background

In this chapter, we provide background material relevant to this dissertation. In Section 2.1, we cover Radio Frequency (RF) basics and then briefly discuss the (WiFi) IEEE 802.11 standard. In Section 2.2, we outline the performance challenges for 802.11 networks, followed by a discussion of two commonly used interference models in Section 2.3. Finally, in Section 2.4, we provide some background on enterprise WLAN design over the past decade.

2.1 IEEE 802.11 Networks Primer

In this section, we first describe some RF basics and then briefly cover parts of the IEEE 802.11 standard that are relevant to this dissertation.

2.1.1 RF Basics

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 cannot be achieved 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 an RF receiver must receive to detect the transmission of a wireless signal. Most wireless-card manufacturers conservatively set this threshold to a low value (e.g. -85 dbm) to prevent

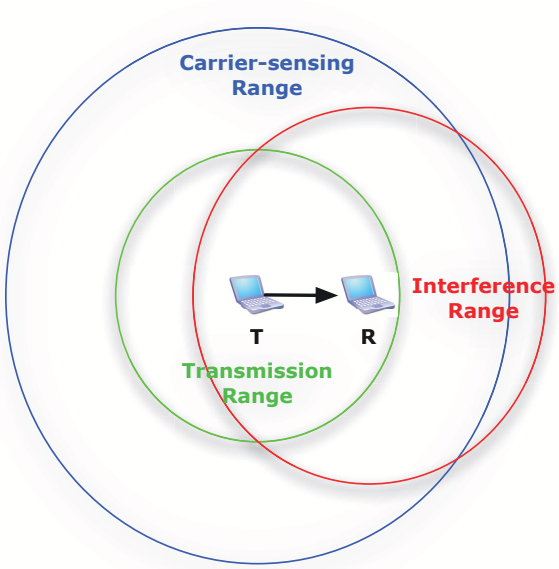

Figure 2.1: The transmission and carrier-sense ranges are defined by the transmitter (T), and the interference range is defined by the receiver (R).

interference with neighbouring devices. The effect of signal attenuation can be captured with the help of ranges as defined by the transmitter and receiver (as shown in Figure 2.1). These are described in greater detail below:

- *Transmission Range:* The transmission range is the range within which the receiver of a signal can decode the transmission correctly. This is typically smaller than the carrier-sensing range of the transmitter (for example, it is typically considered half the interference range in some RF propagation models).
- *Carrier Sense Range:* The carrier-sense range is the range within which the transmitter's signal exceeds the CST of the receiver. The receiver detects the medium to be busy and does not transmit at this time. The receiver can choose to de-sensitize itself to such signals by raising its carrier sensitivity threshold.
- *Interference Range:* The interference range (defined by the receiver) is the range within which any signal transmitted by another source interferes with the transmission of the intended source, thereby causing a loss at the receiver.

The three ranges shown above are affected by the power of the transmitter. The greater the transmission power, the more co-located nodes can receive the transmission,

and also the more nodes whose communication with other nodes will be affected by this transmission. The transmission range is also affected by the data (or coding) rate used by the transmitter. The higher the coding rate, the shorter the range, and vice versa. Note that while the ranges are shown to be circular, in reality, they can be arbitrary and depend on effects such as multi-path fading, scattering, etc.

2.1.2 IEEE 802.11 Overview

IEEE 802.11 (WiFi) is the most popular standard used for providing short-range wireless connectivity to users. It is designed to be simple yet able to adapt to changing environmental conditions. An 802.11 network can operate in one of two modes: infrastructure mode and ad hoc mode. In infrastructure mode, a device known as an Access Point (AP) acts as a bridge between the wired and wireless network and centralizes all wireless traffic. A second device known as the client, *Associates* (or connects) to the AP in order to gain access to the network. Clients can only communicate with the APs and not with other clients. In IEEE 802.11, a single AP's coverage cell is known as a *Basic Service Set (or BSS)*. When multiple APs are deployed in an enterprise, they form an Extended Service Set (or ESS). In this dissertation, we focus on these types of wireless deployments.

In ad hoc mode, there is no centralization and clients are able to directly connect to each other. Clients may forward traffic for each other to transfer data between hosts that are not in direct communication range. Ad hoc mode is uncommon and not used in this dissertation

The IEEE standards bodies have defined multiple communication modes for the 802.11 standard. The two most common modes that operate on the 2.4 GHz frequency band are 802.11b and 802.11g. IEEE 802.11b predates 802.11g and supports the following communication data rates: 1, 2, 5.5 and 11 Mbps. By contrast, 802.11g uses OFDM technology to sustain higher data rates. The data rates supported by 802.11g are: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. Furthermore, 802.11g is also backwards compatible with 802.11b and is intended to replace it as the de-facto 802.11 standard for the 2.4 GHz band. More recently, the IEEE 802.11n standard has started gaining momentum and is touted to support data rates of up to 500 Mbps.

Another common standard used in practice is IEEE 802.11a. 802.11a operates in the 5.8 GHz frequency band and uses OFDM technology to achieve the same data rates as 802.11g. 802.11a predates 802.11g but is gradually losing momentum as more WiFi devices are now being shipped with only 802.11b/g support. While the reasons for this

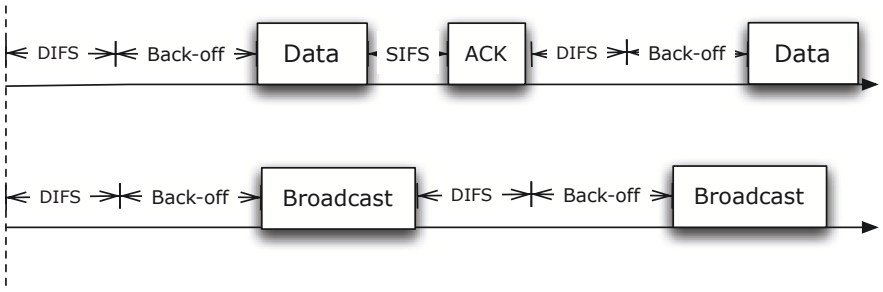

Figure 2.2: The top figure illustrates transmissions of unicast (Data) frames using 802.11. The bottom figure illustrates transmissions of Broadcast frames

are unclear, we believe that it is likely because the 2.4 GHz band has better propagation properties than the 5.8 GHz band.

2.1.3 IEEE 802.11 MAC Layer

The properties of the IEEE 802.11 MAC layer that we discuss here are the channel access mechanism (CSMA/CA) along with a discussion of how data packets are transmitted, the virtual carrier sensing (VCS) mechanism, and the implementation of broadcast and CTS-to-self packets.

Channel Access

The IEEE 802.11 MAC layer uses the Distributed Coordination Function (DCF) to independently allow each device to access the channel[1]. The basic idea is that devices first sense the channel and if it is idle, only then do they initiate a transmission. Channel sensing is done with the help of the physical carrier-sensing mechanism called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The fundamental difference between wired and wireless networks is the mechanism for detecting collisions on the medium. For wireless networks, it is practically impossible to detect collisions on-the-air. Hence, the protocol uses a collision avoidance mechanism, as well as positive acknowledgments (or ACKs) to know whether a packet was successfully transmitted. The procedure for exchanging frames using DCF is shown on the top half of Figure 2.2. Whenever a device wishes to transmit a frame, it must contend for the channel. It does

[1]The Point Coordination Function (PCF) channel access mechanism was also proposed in early versions of the 802.11 standard. However, for reasons that are unclear, it was abandoned in favor of the DCF approach discussed in this section

this by first waiting a fixed length of time (called DIFS). Once complete, it chooses a random number, upper bounded by a certain amount (called the *contention window size*) and counts down these number of slots before transmitting on the channel. The slot length is fixed for the 802.11 standard. Once the Data frame is successfully transmitted, the receiver waits a fixed period of time (called SIFS) and sends an ACK response to the transmitter. The ACK assures the sender that the data packet was correctly decoded by the receiver. This process then repeats for successive frames.

The random backoff period (also called *binary exponential backoff*) prevents co-located devices from picking the same number of slots to countdown and thus prevents them from simultaneously transmitting on-the-air, resulting in a collision. IFS intervals provide a way to synchronize transmission events in the network and also prioritize different types of traffic. Because control traffic (e.g. an ACK) has higher priority, it uses the smallest SIFS interval when contending for the medium. By contrast, data traffic uses the longer DIFS interval during contention. Two other contention periods, namely AIFS and EIFS are also defined by the standard. However, they are not relevant to this dissertation and are not discussed any further.

Broadcast Packets

Broadcast (and multicast) frames are intended for all nodes in the transmitter's neighbourhood. The bottom half of Figure 2.2 shows the frame exchange procedure for Broadcast packets. This procedure is identical to the procedure used for transmitting Data frames, but differs only in that ACKs are not sent back to the transmitter. Broadcasts are useful for diagnostics or when we want to measure a certain property of the transmitter. In later chapters, we show how we use broadcasts as part of our interference measurement framework.

Virtual Carrier Sensing (VCS)

The virtual carrier sensing (VCS) mechanism in 802.11 is used to allow a sender to reserve the channel before transmitting a data packet (see Figure 2.3). The procedure for transmitting Data frames with the help of RTS-CTS packet is shown in Figure 2.3.

The procedure begins by the sender transmitting an RTS frame. Once RTS transmission is complete, the receiver waits a SIFS period and responds with a CTS frame, at which point the medium is reserved for the Data transmission. The rest of the procedure is similar to that shown in the top half of Figure 2.2, except that the sender no longer

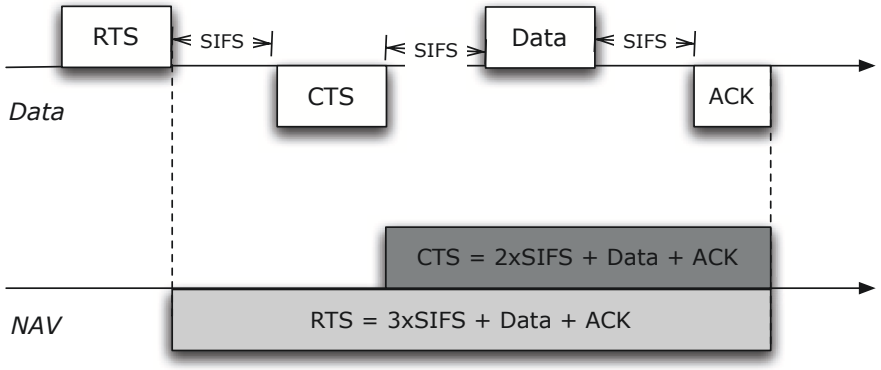

Figure 2.3: An illustration of the virtual carrier sensing mechanism of the 802.11 standard

needs to perform randomized backoff before transmitting the Data frame. Instead, it only waits a SIFS interval and proceeds with the Data transmission.

RTS-CTS control packets are used to set a Network Allocation Vector (NAV) field at neighbouring nodes that forces them to count down an additional time (defined by the NAV) before accessing the medium. When a neighbouring node receives an RTS packet, it sets its NAV field to the amount of time it would take to transmit the CTS, Data, and ACK frame, plus three times the SIFS interval, which is the amount of time that elapses between these frame transmissions. When a neighbouring node receives a CTS packet, it sets its field in the same way but discounts the time to contend and transmit the CTS frame. Effectively, if the RTS-CTS is successful, a sender is able to successfully complete a data transmission without the possibility of collisions from co-located nodes. This mechanism is commonly used to handle hidden terminal interference, described in Section 2.2.2 of this chapter.

CTS-to-Self

There may be cases when a single AP is serving both 802.11b as well as 802.11g clients (this is termed *mixed mode*). 802.11g clients transmit OFDM-modulated signals that 802.11b clients cannot detect. Therefore, 802.11b clients sense the medium to be idle and may begin transmission. To prevent this from occurring, 802.11g clients implement *protection mode*. In this mode, an 802.11g client transmits an unsolicited CTS packet (i.e. one not preceded with an RTS) that is addressed to itself (i.e. it places it's own MAC address as the destination). This *CTS-to-self* frame is modulated to allow 802.11b clients to decode it. Upon receiving the CTS-to-self, 802.11b clients set their NAV field

and allow the 802.11g client to transmit collision-free on the air. Protection mode is commonly used by the AP to reserve the medium before transmitting a frame to an 802.11g client. In our work, we use this mechanism to support interference measurements in an online network.

2.2 Performance Challenges for IEEE 802.11 Networks

When designing and deploying IEEE 802.11 networks, a number of challenges must be addressed to ensure consistent and dependable performance for wireless clients. We subdivide these challenges into those that are the result of RF-based communication and those that arise due to design flaws in the IEEE 802.11 standard.

2.2.1 RF Challenges

Electromagnetic signals are the primary means of carrying information in modern-day wireless networks. RF signals undergo a variety of transformations (termed "fading") as they propagate through the wireless medium (or *channel*), which can distort the signal and lead to data corruption. Though there are many such transformations [104], in this section, we focus on those that are common in indoor environments.

RF fading refers to the attenuation and transformation a signal undergoes as it traverses the wireless medium. Attenuation is the natural decay in the signal power that occurs as the signal moves further away from the transmitter. In addition, other effects such as *reflection*, *diffraction*, and *scattering* can also occur. Coupled together, these effects cause the signal to degrade (or fade) in a variety of ways. There are two major types of fading relevant to indoor networks. These are *Large-scale Fading* and *Small-scale Fading*.

- **Large-Scale Fading:** is defined as the pattern of variation in signal strength over large transmitter-receiver distances. Large-scale fading has been studied extensively and there are well-known propagation models that capture its effects. Model complexity can vary from incorporating only free-space path loss to augmenting the model with specific environmental properties (such as building material used). In general, little can be done to counter the effects of such fading, except to increase the power at the transmitter [104].

- **Small-Scale Fading:** is defined as the pattern of variation in signal strength over very short distances and represents rapid fluctuations that occur as the signal propagates through the air. The most common small-scale fading effect in indoor environments is termed *multi-path*. Multi-path is the result of multiple reflections of the same signal arriving out-of-phase at the receiver. This either amplifies the signal, or degrade its power. Multi-path fading is experienced over short distances in space and time. Multi-path has the potential to drop the signal power down to a null, but techniques such as antenna and receiver diversity have been used to counteract these effects [92]. Recently, Multiple-Input Multiple-Output (MIMO) systems have also been proposed to address the problem of multi-path fading. Vendors are already shipping MIMO-based APs based on the IEEE 802.11n standard [21, 18].

2.2.2 IEEE 802.11 Challenges

We now briefly cover the key performance challenges with respect to the 802.11 standard.

RF Interference

RF interference is the inability of a transmitter to correctly transmit information to a receiver because of the simultaneous transmission by one or more transmitters co-located in the neighbourhood of the transmitter-receiver link. Interference can be of two types: 802.11 interference and non-802.11 interference. These are discussed separately.

802.11 Interference

The IEEE 802.11 standard uses the Distributed Coordination Function to allow independent channel access, as discussed in Section 2.1.3. This mode of channel access can bring about two (independent) problems, first highlighted in seminal papers by Karn et al [75] and Bhargavan et al [43]. These are the *hidden terminal* and *exposed terminal* problems.

Hidden Terminals occur when two senders that cannot carrier-sense one another (i.e. they are *hidden* from each other) simultaneously transmit on the medium. In this case, the intended receiver of one (or both) of the senders receives transmitted signals from both senders. In effect, a collision occurs, and the receiver is unable to decode the signal from its intended sender. This is illustrated in Figure 2.4. Note that there may be cases where a collision occurs but it does not lead to signal corruption, allowing the

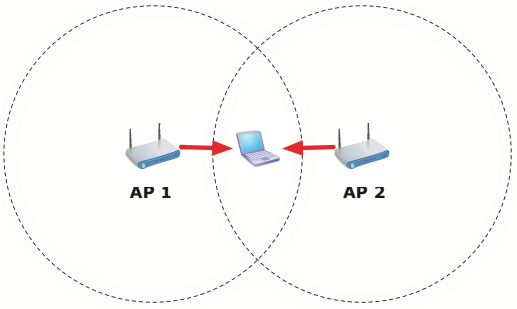

Figure 2.4: AP 1 is hidden from AP 2 and vice versa.

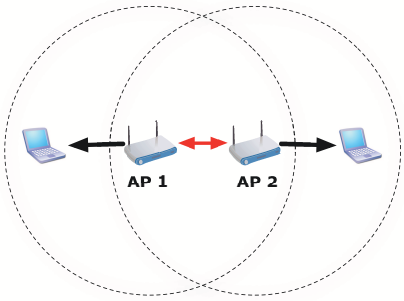

Figure 2.5: AP 1 and AP 2 are in each others carrier-sense ranges and thus cannot transmit simultaneously.

receiver to successfully decode the intended signal (because the signal is of sufficiently high power). This phenomenon is termed *physical layer capture* (or *power capture*) and is not treated as interference in our work.

Hidden terminals can potentially be addressed with the help of virtual carrier-sensing [37][2]. However, identifying hidden terminals in practical deployments is hard and requires active measurements, as we show later.

Exposed Terminals occur when a sender is unable to transmit because it senses transmissions of a co-located sender, even when no harmful interference would occur at their corresponding receivers. In this case, the senders are *exposed* to one another. This is illustrated in Figure 2.5. Exposed terminal problems may be solved by adjusting the carrier sensitivity threshold (CST) at the senders [122]. However, one must be careful not to choose an arbitrarily high value for CST, as doing so may have an adverse affect on co-located nodes with lower CSTs.

Non-802.11 (External) Interference

Other sources of RF interference that also impact performance are from devices that operate on the same $2.4 - 5$ GHz unlicensed bands. Examples include cordless phones, conventional microwave ovens [37] and other wireless technologies such as Bluetooth and Zigbee. Prior work on non-802.11 interference is mostly theoretical (or simulation-based), studying the effects of narrow-band interference on 802.11 networks [74, 99].

[2]Virtual carrier-sensing only works in cases where all interferers are able to successfully decode the RTS/CTS packets. Else they will not be silenced during transmission.

These works focus on the effect of this interference on different physical layer modulation schemes. Recently, Gummadi et al. [63] showed how non-802.11 interference affects other parts of the 802.11 frame reception process as well.

IEEE 802.11 networks mostly use Direct Sequence Spread Spectrum (DSSS) for spreading data across a 22 MHz wide channel to prevent narrow-band interference from non-802.11 devices. Channel hopping techniques such as Frequency Hopping Spread spectrum (FHSS) have also been used to allow transmitters/receivers to hop between different channels and avoid narrow-band interference. These techniques, coupled with CSMA/CA, are currently the only safeguards against interference from non-802.11 RF devices. Recently, some working groups have begun looking at developing standards for minimizing interference between 802.11 and non-802.11 devices such as military radars [12].

802.11 Throughput Anomaly

Infrastructure-based 802.11 networks potentially suffer from a well-known performance anomaly that degrades client performance. DCF causes wireless devices in the same cell to equally contend for access to the wireless medium. Once a node wins access to the medium, the duration for which it occupies the medium depends on two factors. The first is the size of the packet, which is typically 1400 bytes (the MTU specified for the IEEE 802.3 Ethernet standard) [3]. The second is the data rate (or modulation) selected for the transmission, which is a function of the link quality between the client and the AP. Clients far away from the AP have weak links, and thus use a lower data rate for transmission, whereas clients close to the AP have strong links and are able to sustain higher data rates for transmission. Due to this disparity in rates, clients that use lower data rates access the channel for longer time periods, causing other clients in the cell to wait longer to transmit their packets. In such situations, the throughput sustained by all clients in the cell is determined by the throughput of the slowest client. This is termed the 802.11 performance anomaly and was first highlighted by Heuse et al [67].

Time-based fairness has been the proposed as a solution to the 802.11 performance anomaly [112, 70, 93]. The idea is that each client not only gets an equal opportunity to contend for the channel, but also gets an equal amount of time to transmit on the channel. In this approach, the client must choose a suitable transmission data rate and corresponding packet size to meet the deadline requirements specified by the channel access protocol. Another approach performs intelligent MAC scheduling to support

[3] Ethernet MTU is used so that frame conversions from IEEE 802.3 to IEEE 802.11 and vice versa are easily done

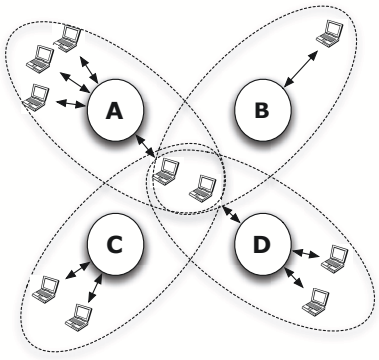

Figure 2.6: Connectivity graph for an enterprise network of 4 APs (A,B,C,D)

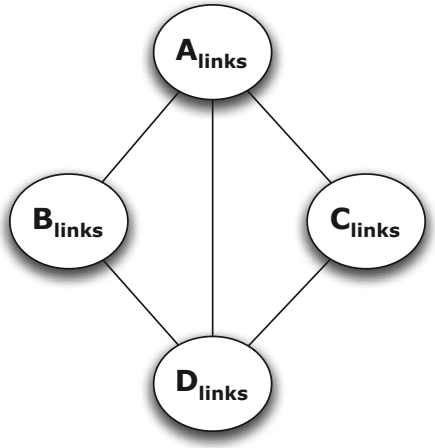

Figure 2.7: Conflict graph for the connectivity graph shown in Figure 2.6. AP A's and D's clients in between the APs experience interference from all APs.

drive-by vehicular Internet access [64]. This approach assumes that each client will eventually enter a "good" region in the AP's cell. This assumption does not hold in enterprise WLANs.

2.3 Modeling Interference in 802.11 Networks

There are two main techniques proposed in prior work for modeling interference in 802.11 networks. They are the *conflict graph* model and the *SINR* model. We briefly discuss each of them next.

2.3.1 The Conflict Graph Model

A common data structure used to incorporate network-wide interference is a conflict graph. This data structure was first proposed in the landmark paper by Yang et al. [121], to model interference in wireless mesh networks. Conflict graphs serve as input to algorithms that optimize network performance in large-scale wireless systems.

In a conflict graph, vertices correspond to the links in the connectivity graph. There is an edge between two vertices if the corresponding links cannot be active simultaneously (or if doing so will cause the links to interfere with one another). Therefore, we add an edge between two links L1 and L2 if either one (or both) of them lies within interference

range of the other (as discussed in Section 2.1.1). Note that the edges in this conflict graph are undirected. This definition of conflict graphs has been applied to distributed 802.11 networks such as wireless mesh networks (WMNs) [72].

Conflict graphs have also been applied to infrastructure-based wireless LANs. In this case, the vertices of the conflict graph represent one or more links from an AP to its clients. If APs directly interfere with each other, or at least one of their clients experiences interference from a neighbouring AP (or one of its clients), an edge is added between the vertices that correspond to the APs links. Thus, the edges encode both direct and indirect interference between the APs. As an example, Figure 2.6 shows a connectivity graph for a typical infrastructure deployment consisting of four APs. The conflict graph for this connectivity graph is shown in Figure 2.7. The conflict graph features an edge between the vertices corresponding to APs A and D and all neighbouring APs because the clients of APs A and D that lie at the center of all the APs experience interference from these neighbouring APs.

The conflict graphs described above have also been extended to incorporate weights on the edges of the conflict graph. This allows preferential treatment of certain edges during the optimization process. For instance, Mishra et al. [89] extend the model to include the number of clients that are affiliated to the AP, to give priority to APs serving greater clients. Similarly, Ding et al. [55] add edge weights that correspond to the degree of separation (in frequency and space) between any two links in a wireless mesh network.

Despite these efforts towards modeling conflict graphs for 802.11 networks, the existing definitions of these graphs have a few shortcomings (details are discussed in Chapter 4). This motivates us to re-examine the current models and extend them to incorporate features that accurately model interference in enterprise WLANs.

A second limitation of prior work using conflict graphs is that they require lengthy measurements to generate the graph. To measure conflicts between all pairs of links, $O(n^2)$ measurements must be performed, where n is the number of nodes in the network. Padhye et. al [98] show that measuring all such conflicts can take up to 28 hrs for only a 22 node wireless testbed. This motivates the design of approaches that reduce the measurement overhead. In Chapter 6, we present a comprehensive overview of prior schemes proposed to measure conflict graphs.

2.3.2 The SINR Model

The Signal-to-Interference Noise Ratio (SINR) model is widely used in simulators such as QualNet [13], to gauge the performance of wireless receivers. At a high level, this

model computes the difference between the signal power and the interference plus noise power at the receiver. Formally, SINR is computed as follows:

$$SINR = S/(I + N) \tag{2.1}$$

where, S is the signal power, I is the sum of all interfered signals, and N is the effect of channel noise. The SINR value is used to compute the bit error-rate, which in turn determines whether a packet was successfully received in the presence of interference. Note that each wireless data rate supported by 802.11 has a minimum SINR threshold, below which correct reception is not possible.

Prior work has used the SINR model as a way to avoid the computational overhead of measuring all pairwise configurations for the conflict graph [105, 101, 76]. Therefore, while the SINR model is itself an interference model, it has primarily been used to reduce the measurement overhead of conflict graph construction.

The basic idea in exploiting the SINR model for reducing measurement overhead is as follows. Each node in the network is instructed to broadcast packets in turn, while all other nodes collect signal strength (or Received Signal Strength Indicator (RSSI)) measurements for those packets. These RSSI measurements then seed the SINR model which predicts interference between pairwise links. This reduces the measurement overhead from $O(n^2)$ to just $O(n)$.

Reis et al [105] were the first to use RSSI measurements in conjunction with the SINR interference model to predict the probability that two links in the network interfere with each other. This model was then used to predict link throughput. Qiu et al [101] and Kashyap et al [76] extended this idea to include multiple simultaneous transmitters, using an N-node markov model to predict throughput.

Limitations

Despite its popularity, the SINR model has a number of limitations. We sub-divide these limitations into those that are inherent to the model and those that are engineering constraints that must be addressed when using the model in the real world.

Model Limitations

1. The SINR model assumes that interference between links is binary (i.e. 0 or 1). In reality, there is a significant gray zone where the impact of interference is not well-defined. Assessing interference in these situations requires real-world measurements.

2. RSSI measurements are assumed to be stable throughout the measurement period. While this may be true in stationary scenarios, it does not hold in general [122]. Furthermore, RSSI measurements generally only work for links with high delivery ratios. For weak links, where we have only a few RSSI measurements, interference cannot be accurately predicted.

Real-world Constraints

1. The SINR thresholds (defined for different wireless data rates) have been shown to differ for different locations [83]. Therefore, they must be computed for each location separately, making it difficult to use this model for mobile clients.

2. RSSI values reported by commodity WiFi cards today are only available for packets that are either correctly decoded or whose PLCP (PHY) header is correctly picked up by the receiver. As a result, packets whose PHY header is corrupted due to interference are not considered when predicting link throughput.

3. RSSI measurements are only taken on the preamble of the 802.11 frame. The mean RSSI on the entire frame is not reported by off-the-shelf commodity wireless cards. Thus, the reported RSSI is not an accurate indicator of the mean signal strength observed on the actual packet.

4. There is no standard definition of RSSI common across all vendors. Each vendor customizes the RSSI metric to suit their needs. Therefore, for each vendor, the conversion from RSSI to signal strength (in dbm) must be done separately.

5. RSSI measurements must be performed at night, when no background traffic is present on the medium (to remove the interference and noise factors from the measurements). While this approach can be applied to predict certain types of conflicts (e.g. AP-AP conflicts), it is not suitable for all conflicts (e.g. AP-client conflicts).

6. The SINR model requires client modifications. Because RSSI measurements must be collected at the receiver, clients must be modified to report these measurements to the APs.

These limitations make it difficult to apply the SINR model for managing interference in enterprise WLANs. As a result, in our work, we propose an alternative framework that reduces the overhead of conflict graph measurement, while preserving the measurement accuracy of prior work [98]. .

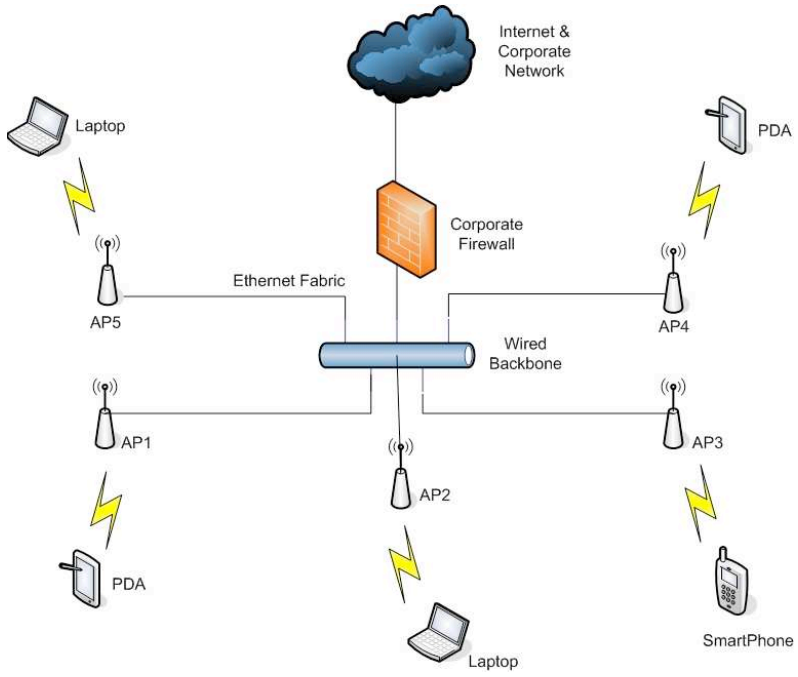

Figure 2.8: Pictorial illustration of a typical enterprise WLAN

2.4 Enterprise WLAN Design: Past and Present

Enterprise WLANs are wireless systems commonly found in corporate offices and university campuses. They comprise a set of APs connected to a wired backbone that carries wireless traffic between the wireless network and the corporate Intranet (see Figure 2.8). For security purposes, enterprises typically shield the wireless network from the Intranet by means of a corporate firewall.

In this section, we discuss how enterprises deploy and configure enterprise wireless systems. Two key techniques exist for optimizing the placement and configuration of APs in the enterprise. These are *static* and *dynamic* optimization. Early enterprise deployments were based primarily on static optimization. However, in recent years, dynamic optimization techniques have been gaining momentum [1, 8, 19, 22].

2.4.1 Static Optimization

Site-surveys are the oldest and most popular technique for deciding the configuration and placement of APs in the enterprise. The configuration that is chosen typically lasts the lifetime of the deployment. Two standard site-survey techniques – manual and virtual site-surveys – are widely used. In manual site-surveys, an RF expert typically obtains

floor maps of the office and annotates them with RF measurements that he has taken at different locations. Using this information and basic rules of thumb, he then places access points and configures them to minimize interference and maximize performance. Manual site surveys can be very cumbersome, especially as the deployment size increases. An alternative method is to perform virtual (or software-based) site surveys. These surveys have the network planner import building maps into a software tool. The tool allows annotation of the map with building specific information (e.g. wall thickness, construction material, etc.). Access points are then placed on the map and their signal coverage predicted using well-known RF signal propagation models (discussed earlier). Propagation models approximate the physical effects of the environment on the propagation of the signal in geographical space. The greater the complexity of the model, the greater its computational requirements. Many techniques therefore reduce this complexity by simplifying the models [60, 120].

Despite these efforts, there are a number of shortcomings to static optimization. First, it is costly and time-consuming to perform. Depending on the deployment size, manual site surveys can take anywhere from a few weeks to several months to complete. Second, static optimization assumes a constant RF environment [37]. In reality, RF signal propagation can change significantly, even over the course of a day [97]. Therefore, there will likely never be a single optimal configuration that is most suitable for the deployment. Thus, network configurations need to be changed to counteract the effects that lead to degraded client performance. This motivates techniques that support *dynamic optimization*.

2.4.2 Dynamic Optimization

Dynamic optimization is a suite of techniques that allow the network to be periodically measured and optimized based on *dynamic* changes that occur in the environment. There are two broad categories for dynamic optimization: 1) network monitoring with manual configuration and 2) network monitoring with automatic configuration. The latter are termed “self-managing” enterprise WLANs.

In the first approach, WLANs are assumed to be capable of automated monitoring and periodically acquire network state to decide whether configuration changes are required at a given point in time. Monitoring is done using either the existing network, or through out-of-network devices (e.g., wireless sensors) that periodically probe and measure the network. The resulting information is then aggregated at a central Network Operations Center (or NOC). An administrator operating the NOC analyzes the computed statistics

and makes any necessary configuration changes. SNMP-based management tools have been proposed for this purpose, and are commonly used in the context of enterprise wireless LANs

In the second approach, the WLAN automatically monitors the network *and* automatically performs configuration changes as and when they are required. Therefore, human intervention is not necessary in such systems. In recent years, the industry has begun shifting to these types of WLAN designs [1, 8, 22, 19]. In the next section, we highlight some architectural features of such dynamic optimization systems.

Enterprise WLAN Architectures

There are two types of dynamic optimization architectures. 1) Decentralized fat-access-points, or 2) Centralized thin-access-points. We discuss each of them in turn.

- *Decentralized Fat Access Points:* Decentralized fat access points are those that have a considerable degree of intelligence (i.e., measurement and configuration capabilities) built into them. They either sense the wireless environment and unilaterally decide the best configuration for themselves, or coordinate with one another to globally agree on the best configuration. Note that AP coordination occurs over the wireless medium and is subject to the wireless channel impairments discussed previously in this chapter. Nevertheless, solutions that use just local information for AP configuration are also known not to be sufficient for generating good and stable configurations [119].

- *Centralized Thin Access Points:* In this architecture, a centralized controller (or switch) connects to all the APs [68]. The APs 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 AP. The APs are 'thin' because they relinquish all decision making capabilities to the controller. An advantage of this architecture is that it is cheaper to maintain since the cost of each AP drops dramatically. As a result, equipment-replacement costs (which are typically the APs) go down. Furthermore, it is also easier to manage and more robust than the Fat AP approach because it does not rely on the wireless medium for network management.

Because of its attractive features, the centralized thin AP architecture has now become the de facto standard for dynamic optimization [1, 8, 19]. In our work, we also

embrace the centralized approach to managing interference in enterprise WLANs. Common management functions in such WLANs include frequency selection and power control. However, fine-grained management techniques such as centralized scheduling are also gaining momentum [109]. In Chapter 7, we present an example of a system that performs fine-grained management of AP selection and traffic scheduling.

Chapter 3

Related Work

In this chapter, we discuss prior work on interference management for enterprise wireless LANs. To put things into perspective, we focus only on competing *systems* that seek to address the same overall problem, but defer discussing prior research on sub-problems related to interference management in enterprise networks. These are covered in subsequent chapters when we discuss our contributions.

We categorize work into model-based approaches (Section 3.2) and measurement-based approaches (Section 3.3). We further sub-divide the latter into approaches requiring client changes (Section 3.3.1) and those that do not require client changes (Section 3.3.2). Finally, we discuss some commercial WLAN offerings in Section 3.4.

3.1 Overview

This dissertation focuses on dynamic optimization for enterprise WLAN design. Therefore, we only discuss prior work in this context and do not cover work on static optimization. In addition, we focus on research contributions from the academic community and briefly comment on some commercial WLAN offerings.

Techniques proposed by the research community can be broadly categorized as model-based and measurement-based. Figure 3.1 illustrates this categorization. We now briefly discuss each of these categories in greater detail below.

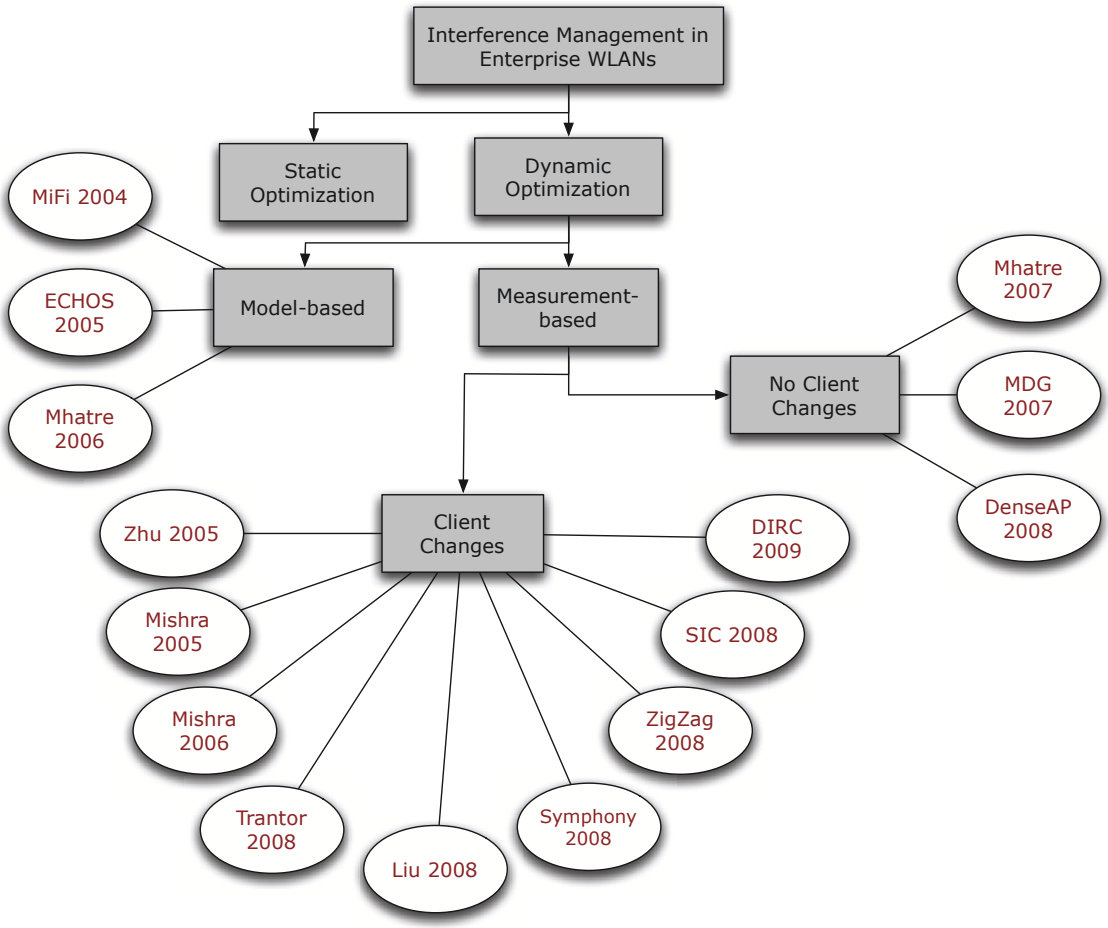

Figure 3.1: Categorization of related work on interference management in enterprise WLANs

3.2 Model-based Approaches

Model-based approaches use an RF propagation model (e.g. two-ray ground model [104]) to predict how RF signals travel through the wireless medium. Early work on interference management in enterprise WLANs is based on this design. We next describe related work that adopts this approach.

MiFi [42] uses a centralized controller design to manage interference in an enterprise WLAN. Interference is modeled with the help of an interference graph (similar to the one described in Section 2.3.1) that is constructed assuming uniform RF propagation, that results in a circular region around each AP. Using this assumption, the interference graph can be constructed through geometrical means and remains constant for a given transmission power of the AP. Using the interference graph, MiFi then uses the now

outdated PCF mechanism in 802.11, combined with a centralized scheduling approach to time multiplex APs such that no two conflicting APs transmit in the same timeslot. This approach has two limitations. First, it uses an abstract propagation model to measure the interference graph. And second, it uses the PCF mechanism which is no longer supported by today's commodity WiFi cards. Moreover, it has only been evaluated in simulation, and thus its real-world performance is not known.

ECHOS [115] proposes a centralized WLAN design to manage the transmit power and carrier-sense threshold (CST) of APs and clients. Each AP is assumed to have as many radios as there are available orthogonal channels. Therefore channel assignment is not necessary for ECHOS. Interference between links is determined with the help of a uniform RF propagation model that results in a circular transmission and interference range around each transmitting AP (as shown in Figure 2.1). The objective in ECHOS is to minimize inter-cell interference by alleviating exposed terminal interference between APs. Therefore, this approach does not deal with hidden terminals.

Mhatre et al. [88] propose an approach for tuning the carrier-sensitivity threshold (CST) of APs to mitigate the interference they experience from the environment. The authors analytically model the problem based on the assumption that APs have a regular hexagonal placement and transmit uniformly in space. By deriving a set of constraints for an objective function (that seeks to minimize interference), they compute an optimal network-wide CST for all the APs in the WLAN. They evaluate their approach via simulations and show that it improves performance over the ECHOS approach described earlier.

Summary: The systems discussed above present some interesting theoretical insights into the performance of different optimization strategies (e.g., centralized scheduling using conflict graphs in MiFi). However, these systems are based on abstract RF propagation models which do not accurately capture interference in the real-world and have also only been evaluated in simulation. In our work, we propose tools and techniques to measure interference *without* the use of RF models and demonstrate their practical application through real-world deployments.

3.3 Measurement-based Approaches

Measurement-based approaches are not based on any RF propagation models. Instead, they are based on the idea that *the most accurate way to determine the impact of interference is to actually measure it.* Prior work in this category follows up on earlier model-based approaches that were inadequate for managing interference in real-world

deployments. To understand the deployment characteristics of these approaches, we divide them into two categories: (1) those requiring client changes and (2) those not requiring client changes.

3.3.1 Client Changes

Zhu et al. [122] propose a system that adapts the network-wide carrier-sense threshold (CST) of both the APs and the clients depending on the packet error rate measured across the network. The network-wide CST is additively increased if the packet error rate is below a threshold and multiplicatively decreased if the packet error rate is above a threshold. Performing network-wide adjustment of CST thresholds requires AP coordination as well as client modifications.

Mishra et al. [89] propose a centralized WLAN approach where they extend the conflict graph model to include weights on the edges of the graph. Weights correspond to the number of clients that would potentially be affected if the APs are assigned the same channel (or frequency). In addition, an interference factor is also considered to represent the degree of separation between channels (in the frequency domain) that are potentially chosen for conflicting APs. In this approach, clients are expected to periodically report (to their APs) the list of APs that they can hear on different channels to construct the conflict graph.

Mishra et al. [90] further build on their prior work by proposing a conflict-set approach to model interference between links in a WLAN. Prior work on modeling interference using weighted conflict graphs have some limitations (as discussed in Chapter 4), motivating the *conflict-set* framework. The conflict set approach maintains two sets for each client, a range set and an interference set. These sets are populated based on client feedback that indicates which APs and clients were heard by a particular client. Once the sets are computed, the network then executes a randomized search algorithm that assigns channels to APs so as to minimize the total interference in the network. While intuitively appealing, this approach does not accurately measure interference between links. This is because the ability to hear a neighbouring AP or client doesn't necessarily imply that it interferes with the client (because of effects such as power capture). The only way to know if two links interfere is to actually *measure* the potential interference.

Trantor [96] is a centralized WLAN design whose aim is to move the management complexity of both APs *and* clients to the central controller. The objective is to prevent client-side decisions from negatively impacting the goal of network-wide optimization. Trantor defines an API for instructing clients to collect measurements and can use these

measurement to construct the conflict graph for the network. Furthermore, Trantor proposes tasks such as traffic differentiation and fault diagnosis for the network as well. While Trantor focuses on the design of a WLAN system, recent work [95] has taken some of these ideas and implemented them on a real network.

Xi et al. [82] propose a system that performs per-link power control for APs and clients that are part of the enterprise WLAN. Transmission powers are assigned using a greedy iterative power control algorithm that uses a conflict graph and attempts to minimize the total number of conflicts in the network. The conflict graph is constructed with the help of the SINR model, which is seeded with RSSI measurements that are periodically collected in the network. While the SINR model is widely used in the literature, it has some key limitations, as discussed in Section 2.3.2. APs and clients exchange RSSI measurements with one another so that every node has complete knowledge of the interference patterns in the network.

Xi et al. [83] recently proposed DIRC, a centralized WLAN architecture that uses directional antennas to improve the capacity of enterprise networks. DIRC uses a conflict graph (generated for all possible antenna configurations) and a TDMA scheduling approach to mitigate interference and improve network performance. As in previous works, the conflict graph is generated using the SINR model. Thus, DIRC requires clients to report RSSI measurements to the controller to generate the conflict graph.

Symphony [102] is an approach that performs synchronous transmit power and rate adaptation on each AP-client link to minimize interference and improve battery life for wireless clients. Symphony operates in phases and requires clients to actively participate in the power and rate adaptation process. During the execution of the algorithm, APs and clients are synchronized to one another and move through phases in lock-step. Symphony implements mechanisms to handle exposed terminals that can potentially arise because asymmetric power levels were chosen for neighbouring links. However, it does not effectively address hidden terminals. In particular, it uses RTS/CTS to handle hidden terminals, which as discussed in Section 2.2.2, does not work effectively in all scenarios.

Zigzag [62] is an approach proposed to combat hidden terminals in WLANs. It is based on a receiver design that uses successive re-transmissions (by the hidden nodes) as a way to bootstrap the process of canceling interference from erroneously received frames to recover the original transmissions. Zigzag supports unmodified clients only in the uplink, i.e. for traffic from the client to the AP. For downlink traffic, clients must be modified to allow decoding at the receiver. Moreover, the proposed modifications require changes to the PHY layer of the radio, which requires specialized FPGAs to implement the decoder.

Successive Interference Cancellation (SIC) [66] has also been proposed to recover signals that experience collisions at the receiver. SIC requires that at least one of the collided signals is recoverable by the receiver. Once that signal is decoded, it can be removed from the collided signal to recover the second weaker signal. Like Zigzag, SIC also requires access to the PHY layer of the radio as well as modifications to the clients.

Summary: The systems discussed above demonstrate interesting ways in which to combat RF interference in enterprise networks. Most of them have also been implemented in real-world testbeds. However, the main drawback of these techniques is that they require client modifications which is undesirable in enterprise WLANs that are inhabited by both a diverse set of users as well as a diverse set of WiFi devices (e.g., baby monitors). They also do not support legacy 802.11 devices, thereby limiting their widespread application. In our work, we propose practical techniques to precisely characterize and mitigate RF interference without requiring client modifications.

Note that the approaches discussed above propose client modifications that range from reporting application layer metrics, all the way down to reporting physical layer information. Having said that, unfortunately, there is no gold-standard approach to modifying clients that can be used as a benchmark to compare against approaches that do not require client modifications. As a result, in our work, we were not able to assess how close these two types of approaches were, in the context of managing interference in enterprise WLANs.

3.3.2 No Client Changes

Mhatre et al [87] propose a method of jointly optimizing the CST and transmit power of APs in a coordinated fashion. The approach does not require any client changes, and each AP selects a power and CST that seeks to meet the performance objectives of the worst client in it's cell. However, in this approach, the authors do not address hidden terminals that can potentially arise because neighbouring APs reduced their transmission power. On the contrary, APs that reduce their transmit power also raise their CST, increasing the chance of hidden terminals between neighbouring cells.

Broustis et al [47] propose the MDG (Measurement-Driven Guidelines) framework, that combines channel assignment, user association, and power control into a unified framework for enterprise WLAN optimization. The authors implement and validate MDG on real-world testbeds. In addition, they prescribe guidelines on how to tune WLAN parameters, based on the specific deployment scenario. However, the prescribed guidelines hold for the common case of multi-channel WLANs [1]. In single-channel

WLANs, it is not clear whether the prescribed sequence of operations still hold. Such WLAN designs do not fall under MDGs configuration guidelines and require further investigation and analysis. Furthermore, the optimization algorithms evaluated with MDG do not accurately measure interference. The channel assignment algorithm does not incorporate interference observed by clients, whereas the power control algorithm is identical to the one described above [87]. Finally, while user association does capture air-time information in its selection criteria (as a way to handle exposed terminals), it does not deal with hidden terminal interference.

DenseAP [94] is a recently proposed centralized WLAN system that seeks to maximize client performance through a combination of client association and load balancing techniques. Clients are associated with only those APs chosen by the central controller, through the use of Beacons with hidden SSIDs. Client affiliations are decided using an *available capacity* metric. This metric estimates the amount of free air time available at each candidate AP and the transmission rate a client is expected to get if associated to that AP. The client is then affiliated to the AP that maximizes the available capacity. Load balancing and handoffs (due to mobility) are also supported by DenseAP. However, although DenseAP accounts for exposed terminal interference using the free air-time metric, it does not implement techniques to address hidden terminals that can also degrade client performance. Detecting hidden terminals in enterprise networks is hard and requires fine-grained coordination among APs. The existing design of DenseAP does not easily support such AP coordination mechanisms.

Summary: While the systems discussed above present interesting ways of managing interference in enterprise WLANs, their key limitation is that they are not able to precisely discover conflicts between links in the network. While they propose heuristics (such as free air-time) as a way to deal with certain types of interference, their solutions are neither precise nor comprehensive. In our work, we systematically address RF interference by precisely capturing conflicts in the form of a conflict graph. We then illustrate the usefulness of the conflict graph by showcasing the gains of using it for a variety of different optimization problems.

3.3.3 Related IEEE Standards

IEEE 802.11 standards bodies have also been scrambling to define protocols and standards that can help enterprise WLANs effectively manage interference. We briefly comment on a few relevant IEEE task groups working towards this goal.

IEEE 802.11k: The IEEE 802.11k standard [24] defines mechanisms by which clients

provide site reports to access points. These site reports contain 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 802.11v: [25] is the latest standard that provides full-featured network management support for IEEE 802.11 networks. 802.11v complements the 802.11k standard by providing necessary support at the infrastructure end. This allows ease of deployment and management and also provides support for services such as load-balancing between access points. The standard also mandates building a common platform to allow access points from different vendors to inter-operate. To achieve this, it plans to use mechanisms proposed in the IEEE CAPWAP standard [23].

While the above standards have been proposed and in some cases incorporated (e.g. 802.11k), WiFi chip manufacturers have yet to widely adopt them. Moreover, the millions of 802.11a/b/g/n devices that have already been shipped represent a significantly large fraction of the user population. Therefore, any solution that requires an implementation of the 802.11k standards limits its usefulness to a small set of users. A better deployment path is to design systems that are legacy compatible and also support upgraded clients (analogous to the 802.11g standard). The framework proposed in this dissertation is developed in this spirit and supports all 802.11 standards to allow widespread adoption. Furthermore, it can easily be extended to support feedback from 802.11k clients as well.

3.4 Industry Solutions

Over the last couple of years, many startups have emerged that are marketing enterprise WLANs solutions [8, 1, 19, 22]. While these WLANs are similar in spirit to those we propose in this dissertation, they are tailored to specific types of hardware and no information is available on the proprietary protocols they use. Furthermore, there is also speculation regarding whether some of these solutions are even standards compliant [17]. In our work, we aim to build solutions that are openly published, implemented on commodity hardware, and compliant with the IEEE 802.11 base standard. Moreover, we develop solutions that require no modifications to end clients.

As discussed in Chapter 2, there are two basic architectures for enterprise WLANs: *Decentralized Fat Access Points* and *Centralized Thin Access Points*. Vendors offering

solutions based on the Fat Access Point design include Auto-Cell [11] and Engim [5]. We note that most vendors offering these types of solutions are no longer in operation.

There are many commercial solutions based on the centralized thin AP design. Meru [8], Aruba [1], Extricom [19], and Trapeze Networks [20] are examples of solutions that use the centralized thin AP approach. Moreover, some architectures combine thin and fat access-point capabilities. Xirrus [22] provides a single integrated device that incorporates multiple APs into a single wireless LAN array. All APs use a common MAC layer and therefore only consist of three components: the base band, RF circuitry, and power amplifier. Therefore, a single device can be used to provide complete coverage for the enterprise. This significantly decreases management overhead. However, the solution does not provide fault tolerance and, in particular, has a single point of failure. While all centralized approaches suffer from this limitation, the integrated WLAN array solution is more problematic because it brings down the entire network (controller plus APs) in such cases. Other centralized designs do not cause APs to fail, in the event that the controller fails. Furthermore, the cost of replacing an integrated WLAN array is also prohibitively high.

3.5 Summary

Prior solutions for interference management in enterprise WLANs span two broad categories: model-based techniques and measurement-based techniques. Measurement-based techniques can be further sub-divided into client-change and no client-change approaches. Because our research espouses wide-spread deployment, no-client change approaches are closest in relation to our work. However, prior work in this category does not precisely measure and model RF interference and instead infers it through indirect means. This can lead to sub-optimal configurations that degrade client performance. With increasing network density, there will be an ever more pressing need to accurately measure interference for enterprise WLANs. This dissertation provides a foundation for such an interference measurement framework, that not only accurately measures RF interference, but also works in an online network and in the presence of legacy clients.

Chapter 4

SMARTA: Designing a Conflict-Graph based Enterprise WLAN[1,2]

In this chapter, we describe the design of a centralized enterprise WLAN architecture that uses conflict graphs to manage interference in the network. The ideas in this chapter develop the *theory* of conflict graphs, as they apply to enterprise WLANs.

Centralized management of network parameters implies the design of centralized algorithms to manage AP configurations. These algorithms take as input the global 'network state' and generate configurations that approximate the global optimal configuration for the network. To capture instantaneous network state, measurements are collected at the APs and fed back to the central controller. Because this dissertation focuses on RF interference, the APs must measure (or infer) RF interference between links and send this information to the controller. The controller, upon receiving this information, encodes it in a format that can be readily used by the optimization algorithms. The conflict graph (presented in Chapter 2) is an ideal tool for encoding such interference information.

[1]This Chapter revises a previous publication: [32] N. Ahmed and S. Keshav. SMARTA: A self-managing architecture for thin access points. In Proceedings of ACM CoNEXT, 2006 (refer to Appendix A)

[2]The content of this Chapter overlaps and significantly extends a Master's of Mathematics thesis entitled: "A self-management approach to conguring wireless infrastructure networks", Nabeel Ahmed, University of Waterloo, 2006.

4.1 Motivation

Conflict graphs (CGs) are a natural framework for modeling interference in 802.11 networks. However, existing approaches for modeling, construction, and use of conflict graphs have some limitations that make it difficult to apply them to enterprise WLANs. These are described in detail below:

- **Inadequate Measurement Approach:** Conflict graphs are typically constructed using standard rules of thumb that are based on either hop distance or a particular RF propagation model. This is shown to be inaccurate especially for indoor environments that are characterized by multi-path fading, scattering, etc [98]. In contrast, measurement approaches that seek to improve accuracy have a lengthy measurement cycle [98, 105], making them in-effective for measuring interference in Enterprise WLANs. While passive measurement techniques also exist that do not have a lengthy measurement cycle [90], they lack accuracy since they assume interference only if nodes are in communication range of each other.

- **Inadequate Model Representation:** Conflict graphs, as presented in Chapter 2 are not adequate for modeling interference in enterprise WLANs. They do not take into account crucial properties of the wireless channel such as interference asymmetry between links in the network and do not distinguish between different types of conflict such as hidden and exposed terminals.

- **Limited Support for Conflict Graph Changes:** In enterprise WLANs, the conflict graph can change rapidly, necessitating the need to (re-)compute it on short timescales. This occurs for two reasons. First, clients come and go in the network, and we are required to measure interference for them. For mobile clients, the environment can change in a matter of seconds as they move about in the enterprise. Second, even for stationary clients (or links), prior work shows that the conflict graph can change over modest timescales [97]. Existing techniques do not prescribe ways to handle such changes and assume that interference patterns are largely static.

- **Dynamically Changing Objectives:** Performance can be defined in a variety of different ways and depends on the application(s) running on the client devices. Because this information is not available a priori, by design, the enterprise WLAN should allow the ability to change performance objectives on-the-fly. Therefore, the enterprise WLAN should provide appropriate tuning knobs to the administrator to allow him to configure the network based on the policies set forth by the

IT department managing the infrastructure. Our investigation reveals that policy specification mechanisms in existing enterprise WLANs are cumbersome and administrators rarely tinker with them for fear of misconfiguring the network [33].

- **Require Client Support:** Existing conflict graph construction techniques that empirically measure interference require changes at the receiver in order to report metrics such as packet loss rate and received signal strength. This inhibits widespread deployment. Moreover, it does not support legacy clients.

Motivated by the problems described above, in this chapter, we present the design and evaluation of a new approach to WLAN configuration that we call SMARTA. SMARTA provides the basis for follow-up work in subsequent chapters. The rest of this chapter is organized as follows. Section 4.2 describes the design goals for the conflict-graph based enterprise WLAN. Section 4.3 presents an overview of the SMARTA architecture and Section 4.4 discusses the models we use to characterize performance. Section 4.5 discusses the limitations of existing conflict graph models and Section 4.6 discusses our extensions for managing interference in enterprise WLANs. Section 4.7 discusses novel techniques for measuring interference in an online network. Section 4.8 presents algorithms for frequency selection and power control that apply the measured conflict graph to optimize network performance. We evaluate the features of the SMARTA architecture in Section 4.9 and end with related work and a discussion in Sections 4.10 and 4.11, respectively.

4.2 Design Goals

In this chapter, we aim to address the problem of designing a practical enterprise WLAN based on conflict graphs. Our goals in designing this architecture are as follows:

- **Free from RF Propagation Models:** RF propagation models are inaccurate especially for indoor environments that experience significant multi-path fading and scattering. To accurately estimate interference, the proposed approach should make no assumptions on RF signal propagation.
- **Richer Conflict Graph Modeling:** As discussed in the previous section, present day conflict graph models are inadequate for optimizing performance in enterprise WLANs. Models specific to such networks need to be designed to allow the optimization framework to make the most out of the information provided to them.

- **Low Overhead - Online Approach:** Measurement techniques with a lengthy measurement cycle are not suitable for enterprise WLANs as environmental changes can take place on timescales of a few seconds. Moreover, measurement-intensive techniques are also not suitable for such networks as measurement traffic shares the medium with data traffic also being carried in the network. Therefore, the proposed approach should support low overhead *online* interference tests, to construct the conflict graph.
- **Ability to Tune System Objectives:** The performance objectives for an enterprise network can change over time, as different applications are run on the network. The WLAN system must provide flexible tuning knobs to allow the administrator to specify and/or change these objectives to suit the needs of the users (e.g., VoIP users are delay-sensitive and seek to minimize end-to-end delay).
- **Infrastructure Only Solution:** To allow rapid deployability into existing WLAN systems, the proposed approach should restrict modifications to only the infrastructure. Modifying clients is not practical and must be avoided.

The SMARTA architecture meets the design goals outlined above. Our infrastructure-based solution, targeted towards enterprise WLANs, does not require client-side modifications, allowing backwards compatibility. Utility functions provide a unified framework for capturing multiple and even conflicting performance objectives. Moreover, SMARTA makes no assumptions about RF propagation and uses dynamic optimization to address varying channel conditions.

At a high level, SMARTA uses active probes to build a conflict graph that accurately models the RF environment without making path loss assumptions. Utility functions are defined on the conflict graph to characterize network performance. Finally, a variety of operating parameters can be used to optimize the computed utility. In this chapter, we study frequency selection and power control as the parameters used to evaluate the SMARTA design. Other parameters may also be considered in conjunction with SMARTA, and we discuss some such parameters in Section 4.11.

4.3 Architecture

The SMARTA architecture is illustrated in Figure 4.1. The central controller coordinates the channels and power levels of the thin access points. The channels and power levels are decided based on optimizing a utility function, whose value is computed using measurements performed by the access points. The controller periodically cycles

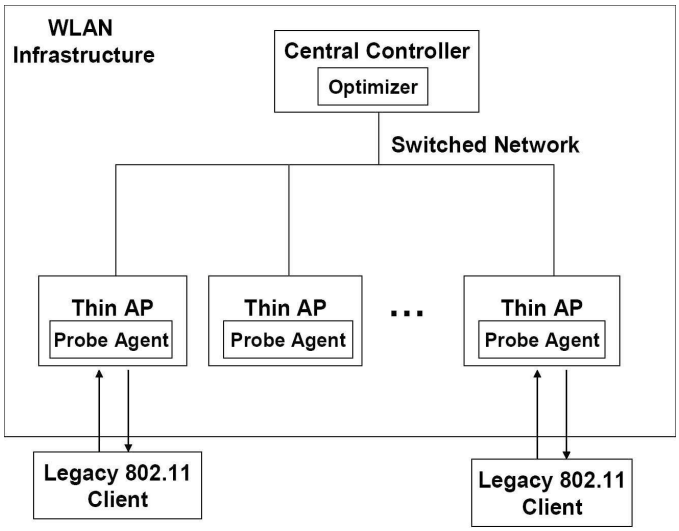

Figure 4.1: SMARTA System Architecture

through five phases: startup, channel assignment, annotation, power-level assignment, and refinement.

In the first or startup phase, the controller obtains the desired performance objective(s) from a network administrator. We assume that the administrator provides the parameters in the form of weights controlling a utility function. We expect manufacturers to provide carefully chosen defaults, so that, in practice, the network administrator could simply choose an objective such as ‘maximize throughput’ or ‘minimize delay’ instead of numerically choosing weights. This is akin to laptop users choosing verbal objectives such as ‘maximize battery lifetime’ or ‘maximize performance’, which are then translated into specific settings for disk spin-down timers and screen brightness.

The utility of a particular system configuration is determined jointly by the weights chosen by the administrator, the current workload, the current RF coverage, and the degree of interference between APs and clients in the system. To keep track of these parameters, the controller computes and periodically updates a conflict graph, where nodes are APs and there is an edge between two APs, if they interfere when assigned the same channel, *assuming they are transmitting at maximum power* (which is the worst case). In the second or channel assignment phase, the optimizer makes use of the CG to generate an assignment of channels for the access points, using the algorithm described in Section 4.8.1. At the end of this step, every AP is assigned a ‘good’ channel. We do channel assignment before power-level assignment because changing an AP’s channel affects all clients associated with it. In contrast, changing its power level is not likely to significantly affect most clients. Therefore, we assign channels at a slower time scale,

and refine power levels at a faster time scale.

In the third or annotation phase, the CG is augmented further to generate an *annotated conflict graph*, or ACG. This is similar in spirit to the conflict set ideas proposed in [90]. The annotated conflict graph adds clients to the conflict graph, which previously only contained access points. During ACG construction, access point channels may be re-assigned to reflect client information in the channel assignment process. The reason for this two-step channel assignment process and its details are discussed in Sections 4.8.1 and 4.8.2.

In the fourth or power-level assignment phase, SMARTA computes appropriate power levels for the access points. The power control algorithm used for this purpose is described in Section 4.8.3.

After this procedure completes, SMARTA moves to the fifth or refinement phase. In this phase, the power levels of access points are altered to account for 'small' dynamic changes in the environment. This allows the system to evolve the configuration in response to changes in the environment. However, there may be circumstances where a large change is observed (e.g., a large group of users flock to a particular location) causing the current assignment of channels and power levels to yield poor performance. This requires re-computing the configuration from scratch. Specifically, if the change in utility exceeds a significance threshold, the system discards the current ACG and starts the optimization process from the beginning, by returning to phase 2. Otherwise, it remains in the refinement phase.

The next sections elaborate on each of these phases in greater detail. We first discuss the utility function model.

4.4 Utility Model

We use utility functions to characterize the benefit from a particular system configuration. The function is typically a linear combination of terms, where each term has a weight reflecting its importance to the network administrator. Note that this approach allows us to overcome the inherent problem of multi-objective optimization with conflicting objectives.

Utility functions can capture any type of performance objective and we discuss some common objectives next. Note that, although we are presenting some typical performance objectives, SMARTA is agnostic to the actual utility function chosen by the network administrator. Here we focus on objectives that maximize aggregate network

throughput. Fairness can also be captured in the utility function, through an appropriate utility function. SMARTA correctly chooses operating parameters to maximize the utility function *independent* of its form.

Let N be the total number of access points, p_1 to p_n represent the performance parameters to be captured, and w_1 to w_n be their respective normalized weights. Then, an example of a typical utility function for a wireless LAN deployment can be stated as follows:

$$U_{total} = \Sigma_{i=1}^{N} U_i \tag{4.1}$$

where,

$$U_i = w_1 p_1 + w_2 p_2 + ...w_n p_n \tag{4.2}$$

Equation 4.1 represents the aggregate utility of the wireless LAN, and Equation 4.2 represents the utility obtained by each of the access points (i representing a given access point). Next, we describe some example instantiations of p_i.

The Utility of Throughput

The utility gained from throughput depends on the nature of the client application. If it is real-time (e.g., U_{rt}), then, as long as the throughput exceeds the required minimum value, full utility is achieved. On the other hand, for non-realtime applications, utility (e.g., U_{nrt}) monotonically increases with increasing throughput[3]. Suppose n non-realtime clients and m realtime clients are associated to the access point. Then, the aggregate utility provided to all clients is,

$$U_{clients} \quad = \quad \Sigma_{i=1}^{n} U_{nrt} + \Sigma_{i=1}^{m} U_{rt} \tag{4.3}$$

where U_{nrt} is a monotone function and U_{rt} is a clamped function, of the achieved throughput. The achieved throughput can be obtained for an AP by counting the number of packets sent to (or by) the client.

Effect of Interference on Utility

Suppose client a is associated with AP A and is potentially interfered with by AP B. How should this be modeled? Our intuition is that if B is mostly idle, then a is mostly

[3]These utility functions are simplified and only meant to illustrate the way in which utility functions can be defined. More sophisticated utility functions can be defined, to suit the requirements of different applications

unaffected. However, if B is mostly busy, then a is likely to pay a price for this. Essentially, we want to map B's load to its expected effect on a, that is, the disutility to a due to the drop in its throughput.

Analytical models that quantify the effect of such interference are known, but they are quite complex even for very simple scenarios [50]. They are also limited in their ability to accurately model the impact of interference. Instead, we choose to empirically analyze (to first order) the effects of interference on the throughput obtained by the interfered node, as follows.

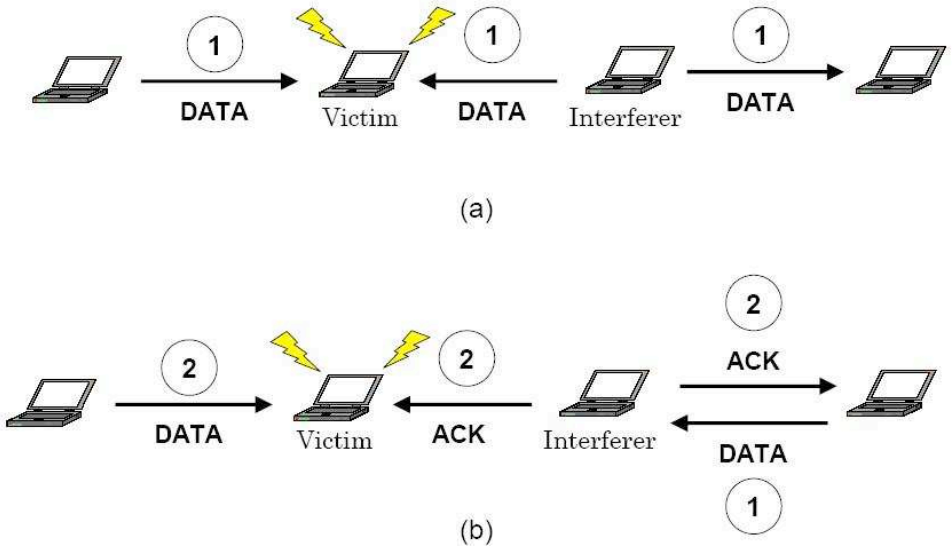

Figure 4.2: (a) illustrates a Data-Data collision scenario where *victim* is the node experiencing interference. (b) illustrates a Data-Ack collision scenario wherein the direction of traffic flow at the *interferer* is reversed. The steps that occur in each scenario are labeled accordingly.

We use the high fidelity Qualnet [13] simulator and vary the sending rate of the interferer, which is transmitting UDP-based CBR traffic. The interfered node also transmits similar traffic at rates high enough to saturate the medium. This simulates the worst case by analyzing the impact of interference on high-throughput flows. We analyze four collision scenarios (Data-Data, Data-Ack, Ack-Data and Ack-Ack) using a simple four node topology, two of which are illustrated in Figure 4.2[4]. The results are shown in Figure 4.3. Packet inter-departure times at the interferer are independent and identically distributed

[4]For each scenario name, the first packet type corresponds to packets being received by the interfered node, whereas the second packet type represents packets interfering with the reception at the interfered node.

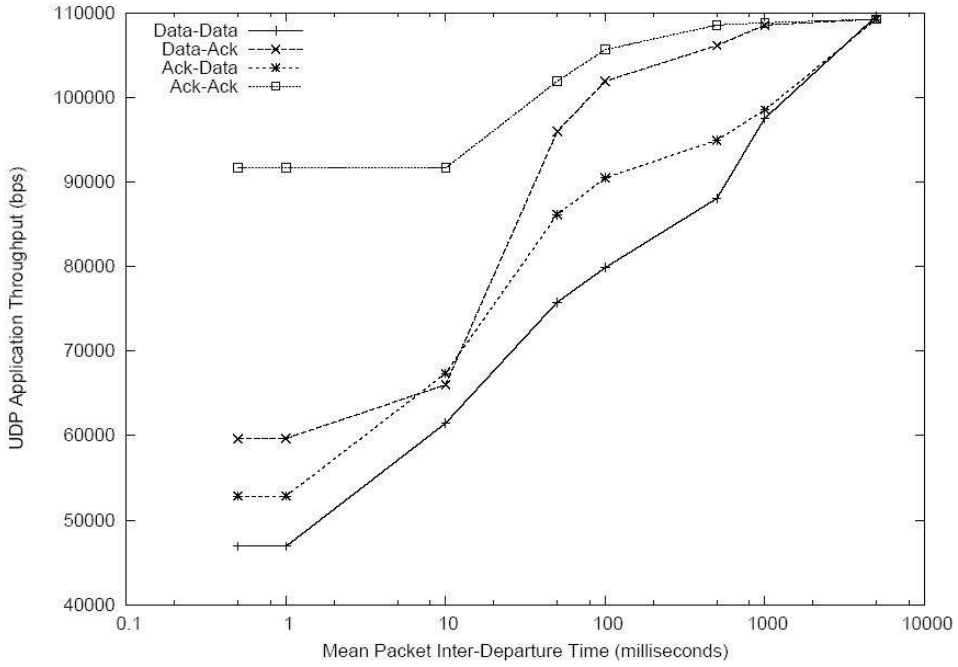

Figure 4.3: Throughput obtained by a node in the presence of interference. The x-axis indicates the mean delay between successive packets sent by the interferer (see Figure 4.2)

using an exponential distribution, with a mean shown on the x-axis. Data-Data collisions have the greatest impact on the drop in throughput of the interfered node. These values obtained from simulation can thus be used to *quantify* the effect of interference by subtracting the carried load (shown in Figure 4.3) of the interfered node from its true offered load. Of course, this is by no means an exhaustive study, but our goal is to attempt to measure the degree of non-linearity in the effect of an interferer's load on the interfered node's throughput. As can be seen, for the most part, the effect is log-linear, and we therefore model it with a simple log-linear model, using an empirically-derived slope. In particular, the effect of interference is a function of the load of the interfering source (represented by the value on the x-axis in Figure 4.3). Note that recent work on characterizing interference has found that the linearity relationship holds in realistic settings as well [97]. This leads us to believe that the result presented in Figure 4.3 is not simply an artifact of the way QualNet models interference.

As described in more detail in Section 4.7, in reality four interference scenarios can occur in a wireless LAN deployment, based on nodes that are participating in the scenario (i.e., whether they are access-points or clients). These are inter-access-point interference (IAP), access-point-client interference (OAP/OC), and inter-client interference (IC) [5]. Thus, the total interference in the network is the sum of these individual interferences and can be expressed as:

$$\begin{aligned} U_{int} &= -(\Sigma_{i=1}^{N}\Sigma_{j=1}^{N} IAP_{ij} eff_i + \Sigma_{i=1}^{N}\Sigma_{v=1}^{K} OAP_{iv} eff_i \\ &+ \Sigma_{v=1}^{K}\Sigma_{i=1}^{N} OC_{vi} eff_v + \Sigma_{u=1}^{K}\Sigma_{v=1}^{K} IC_{uv} eff_u) \end{aligned} \tag{4.4}$$

where, IAP_{ij} is the interference that access point i causes on access point j, OAP_{iv} is the interference access point i causes on client v, OC_{vi} is the interference client v causes on access point i, and IC_{uv} is the interference client u causes on client v. N and K are the total number of access points and clients, respectively. The functions IAP, OAP, OC, and IC are boolean functions that indicate the presence or absence of interference between pairwise nodes. eff_i is the (assumed log-linear) effect of interference by access-point/client i on the throughput of the interfered access point or client.

4.5 Limitations of Existing Conflict Graph Models

In Chapter 2, we hinted at the limitations of existing conflict graph modeling strategies in the context of enterprise WLANs. In this section we highlight two key limitations that

[5] We do not consider external interference in our model.

make the case for a new conflict graph model for enterprise WLANs.

- **Interference Asymmetry:** It is widely known that wireless channels can be asymmetric. Therefore, the channel quality from $A \rightarrow B$ may be different from the channel quality from $B \rightarrow A$. Because of channel asymmetry, it follows that interference can also be asymmetric. Therefore, it may be the case that A interferes with B, while the reverse is not true. In this case, the conflict graph should feature a *directed edge* from $A \rightarrow B$. Existing conflict graph models assume channel symmetry and do not capture this important property of the wireless channel.

- **Type of Conflict:** All conflicts are not the same. While some conflicts may be due to carrier-sensing interference (i.e., exposed terminals), others may be due to collision-induced interference (i.e., hidden terminals). Identifying these interference types is important from the perspective of network optimization. For instance, some techniques (such as centralized scheduling [109]) require knowledge of the conflict type to determine the correct action to take to optimize network performance.

The limitations identified above motivate a new approach to modeling conflict graphs for enterprise WLANs. We introduce this new approach in the next section.

4.6 The Annotated Conflict Graph

In the SMARTA framework, a conflict graph is defined as a graph $G = (V, E)$, where V is the set of vertices and E the set of edges such that:

- $V = \{ap_1, ap_2, ap_3, \ldots, ap_n\}$, where ap_i is access point i.
- $E = \{(u, v) | f(ap_u, ap_v) \leq 0\}$
- $f(i, j) = -(IAP_{ij} eff_i)$,
  where, IAP_{ij} indicates the presence/absence of interference from access-point i on access point j and eff_i is the effect of interference on ap_j[6].

A conflict graph is therefore a *directed graph* where each edge represents interference (or conflict) caused by an access point at which the edge originates, on an access point at which the edge terminates (see Figure 4.4). Due to wireless channel characteristics,

[6]The function f(i, j) is only defined for access points that interfere with each other when transmitting at maximum power using the same channel, and not across all pairs of APs.

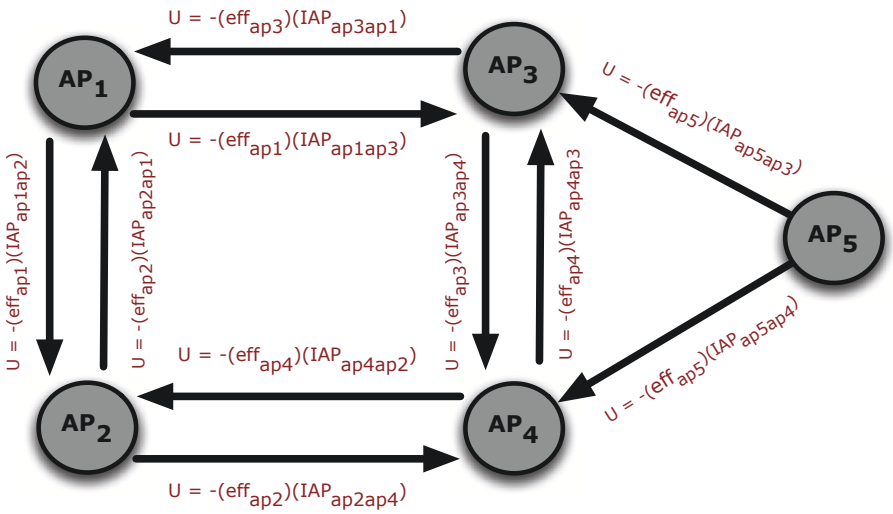

Figure 4.4: The base conflict graph without client information. The conflict edges are directed and annotated with the 'disutility' they cause to the AP experiencing interference.

interference between access points may not be symmetric. Furthermore, the conflict graph is a multigraph, where an edge may exist between nodes due to carrier-sensing interference, collision-induced interference, or both.

The conflict graph is used during channel assignment to minimize the number of conflicts that occur between access points. This reduces to a graph-colouring problem, which is NP-hard [65]. In Section 4.8.1, we discuss a heuristic for channel assignment based on the derived conflict graph.

To perform power control, it is necessary to extend the conflict graph to include clients and AP loads, similar to the approach discussed in [90]. This *annotated conflict graph* has two types of edges between a client and an access point. If a client is associated with an access point, an undirected *association edge* is added between them. If a client interferes with an access point to which it is not associated, or an access point interferes with a client to which it is not connected, a directed *interference edge* is added between them. Finally, if clients interfere with one another, a directed edge is added between them. Figure 4.5 shows an illustration of the ACG. Note that channels that had been assigned before the creation of the ACG may be refined during ACG construction. This is elaborated in greater detail in Section 4.8.2.

Interference edge weights are derived using techniques described in Section 4.4. Association edge weights correspond to the utility that clients receive from their access points.

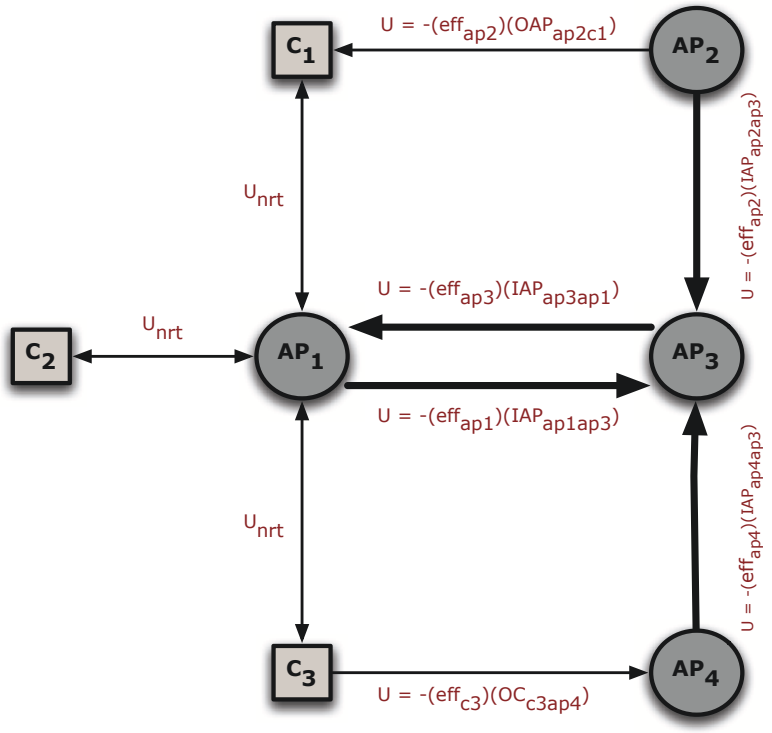

Figure 4.5: Annotated Conflict Graph. Circular vertices are access points and square vertices are clients. The base conflict graph (shown in Figure 4.4) contains only circular vertices. Clients have the same channel as their associated access point.

We point out that the conflict graph models the maximum possible number of conflicts, which corresponds to all access points transmitting at maximum power and using the same channel.

4.7 Constructing the Conflict Graph

The annotated conflict graph requires a number of parameters to compute the utility of the network. This can be divided into two parts; disutility corresponding to interference in the environment, and positive utility corresponding to utility that clients receive from the network. As explained earlier, interference disutility consists of two parameters: (1) The impact of interference (which is a function of the interferer's load, as discussed in Section 4.4), and (2) A boolean function that indicates whether or not two nodes interfere with each other. The latter is captured by means of infrastructure-based testing using a probing agent, discussed next (contrasted with the client modifications required by [90]). Positive utilities are computed by passively observing statistics such as the

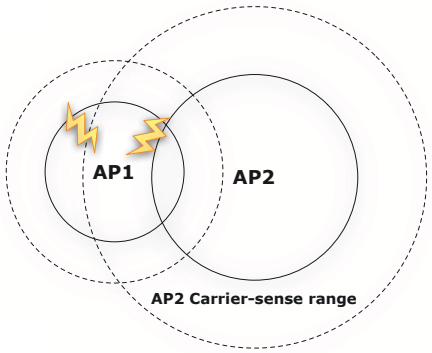

Figure 4.6: An example illustrating zero-hop interference

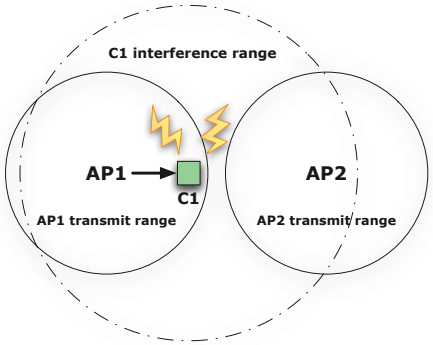

Figure 4.7: An example illustrating one-hop (Overlapping AP) interference. AP_1 and AP_2's carrier-sense ranges have been elided for clarity

number packets sent and received by each AP to and from each client (per observation interval).

We classify interference scenarios in terms of the distance of the interference (in hops) from the infrastructure. For instance, inter-AP interference is zero-hops away from the infrastructure, since APs are directly connected to the wired backbone. The basic intuition is that as the interference moves further away from the infrastructure, it becomes progressively harder to detect and resolve. For each scenario, we prescribe a test to detect the existence of that scenario. In the sequel, the *Tester* is the entity that transmits the probe packet. It may also observe interference for nodes that are not able to do so themselves, e.g. legacy 802.11 clients. A *Sensor* is a node that checks to see if the *Tester* is interfering with it. All tests assume time synchronization; techniques to achieve synchronization within a few microseconds are described in [57]. Note that these tests do not assume any underlying wireless propagation model for their operation, making them applicable to real-world scenarios.

4.7.1 Inter-AP (Zero-Hop) Interference

If the carrier-sense range of an access-point covers a neighbouring access-point, the overlapped access-point suffers carrier-sensing interference from transmissions of the neighbouring access point (as shown in Figure 4.6). Inter-access point interference is 'zero hop' interference because interference is experienced *zero hops* from the infrastructure.

The test for detecting zero-hop interference is as follows. One access-point acts as

the tester while all other access-points act as sensors. The tester transmits m broadcast packets and the sensors listen for interference. During a broadcast, the sensor observes whether there is a change in the state of the channel, (i.e., whether the channel transitions from *idle* to *busy*). If so, then with high likelihood, the sensor is in carrier-sensing range of the tester. If it is also able to decode the packet, then it is in transmission range as well. The tester sends m broadcast packets for this test to increase confidence in the results. As illustrated in [27], a relatively small value of m, around 5, suffices for this purpose.

Each access-point performs this test. Therefore, the total number of tests required to detect zero-hop interference is bounded by $O(N)$, where N is the number of access-points.

4.7.2 AP-Client (One-Hop) Interference

We now describe two *one hop* interference scenarios that involve both clients and access points.

Overlapping Access Point (OAP)

Consider the case where an access-point lies in the interference range of a client associated with a neighbouring access-point. The client experiences interference from this access-point, from whom the client may or may not be hidden. If the client is hidden, packets being sent by it will be suppressed due to contention, and those being received will be susceptible to collision with packets transmitted from the interfering access-point. This is shown in Figure 4.7, where C_1 is associated to AP_1 and experiences interference from AP_2.

To detect this scenario, the following test is performed. The tester, which is the access-point to which the client is associated, transmits an RTS packet to the client, while the sensor which is the access-point that is interfering with the client simultaneously transmits a broadcast packet. Once RTS transmission is complete, the tester sets a timer equal to ($SIFS + Delay_{CTS} + Delay_{bcast}$), awaiting receipt of a CTS from the client, where $Delay_{CTS}$ is the propagation delay for a CTS packet and $Delay_{bcast}$ is the propagation delay for a broadcast packet[7]. If the broadcast packet and the RTS packet collide at the client, the client will not receive the RTS transmission correctly. Thus, it will not respond with a CTS, causing the tester to time out. The tester can then assume

[7]We wait the additional broadcast propagation delay to ensure the client has sufficient time to reply with a CTS if it carrier senses the sensor's broadcast but does not actually experience interference from it.

the RTS packet collided with the sensor's broadcast. The test completes after either the tester receives a CTS from the client or it times out in the process. This test is repeated m times. Since we need to perform this test for each client-AP pair and there are a total of C clients and N APs, the number of tests required to detect OAP interference is bounded by $O(NC)$. While this may appear to be excessive, in Chapter 6, we show that even for a modest sized network of 20 nodes, all such tests can be performed in a matter of seconds.

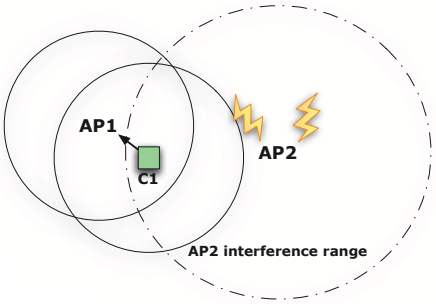

Figure 4.8: An illustration of *one-hop* (Overlapping Client) interference. AP_1 and C_1's carrier-sense ranges have been elided for clarity

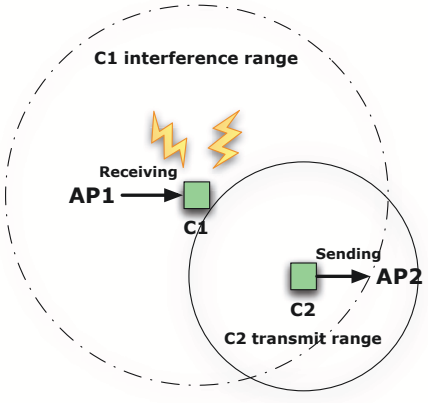

Figure 4.9: An illustration of *two-hop* interference. C_2's data packets collide with data packets being received by C_1. AP_1's transmit and carrier-sense ranges, as well as C_2's carrier-sense range have been elided for clarity.

Overlapping Client (OC)

In this scenario, the client lies in the interference range of an access-point other than the access-point to which it is associated. If the access-point is hidden from the client, packets being sent by the AP will be suppressed due to contention, and those being received will be susceptible to collision with packets transmitted from the interfering client. This is shown in Figure 4.8, where C_1 is associated with AP_1 and causes interference on AP_2.

In order to detect OC interference, the following test is performed. The tester, which is the access-point to which the client is associated, transmits an RTS packet to the client. Upon receiving the RTS, the client responds with a CTS. During the CTS transmission, the sensor which is the access-point that is experiencing interference from the client observes to see a change in the state of the channel. If the sensor detects a change,

then client-access-point interference exists between the sensor and the client. Once the tester receives the CTS packet from the client, the test is complete. This process is also repeated m times to increase our confidence in the result.

Note that if the sensor also experiences inter-access-point interference from the tester, then it must ignore channel state changes during the transmission of the RTS. Thus the sensor ignores state changes for a duration equal to the propagation delay of the RTS packet, from the time at which the tester initiated the RTS transmission (assuming that the APs are tightly synchronized to each other). In this test, all neighbouring APs can simultaneously act as sensors, effectively limiting the number of such tests that need to be performed. Because this test needs to be performed for each client in the network, and we have C clients in total, the total number of tests required to detect OC interference is bounded by $O(C)$.

4.7.3 Inter-Client (Two-Hop) Interference

Clients may also mutually interfere with each other. For this scenario, we are interested in the case where the interfering clients are associated with separate access points because clients connected to the same access point can mitigate interference using RTS/CTS. Note that clients interfere with each other only if their respective access points use the same channel for communication.

For this case, two scenarios can arise, one of which is shown in Figure 4.9. In this scenario, the client experiences interference from a neighbouring client while it is receiving data (C_1). Therefore it is not able to correctly decode packets from the sender. The second scenario corresponds to clients that mutually contend for the medium. This scenario is described in greater detail in [27].

The following test detects inter-client interference for the scenario shown in Figure 4.9. The tester (any one of the APs) sends a dummy data packet to its client. Once transmission is complete, the sensor (second AP) waits a SIFS interval, and initiates transmission of a dummy data packet to its client. Once transmission is complete, the sensor awaits an acknowledgement of its data packet. If it receives an acknowledgement within a timeout period of ($SIFS + Delay_{ACK}$), where $Delay_{ACK}$ is the propagation delay for an ACK packet, then the tester's client does not interfere with the sensor's client.

The intuition for this test is the following. The sensor transmits its data packet when the tester's client is responding to the tester with an ACK. If the tester client's ACK collides with the data transmission being received by the sensor's client, the sensor's

client will not be able to properly decode the data transmission. Therefore, it will not respond to the data packet with an ACK. Timing out of the sensor on the ACK is thus an indication of interference from the tester's client on the sensor's client. Interference detection in the reverse direction can also be done using a symmetric test. This test is also performed multiple times to reduce chances of a poor channel from affecting the results of the test. In the worst case, each client must perform such a test with all other clients, causing the overhead of this interference test to be bounded by $O(C^2)$.

Note that all the interference tests described in Section 4.7 must be conducted in a *'clean'* (i.e interference-free) environment. To arrange for this, the controller asks all APs to both stop their transmission and to force clients in their range to also stop transmission by broadcasting a CTS-to-self [27][8]. This generates the interference-free environment in which to conduct interference tests. We have analytically studied the overhead of conducting such tests along with techniques to mitigate it [27]. In Chapter 6, we practically evaluate the feasibility of this measurement approach on an enterprise-scale WLAN testbed.

4.8 Optimization Algorithms

We first discuss our approach to channel assignment and then discuss the details of power control.

4.8.1 Channel Assignment

Channel assignment attempts to allocate orthogonal channels to nodes in the conflict graph that have an edge between them. Once completed, channels should be rarely changed because this disrupts service for clients. This is particularly important in the SMARTA architecture because legacy IEEE 802.11 clients cannot be instructed to change channels and are therefore disconnected if the AP changes its channel.

To minimize channel changes, channel-assignment is done on the basic conflict graph that deals only with access-point conflicts. Of course, we still need some way to deal with client conflicts and this is done during construction of the annotated conflict graph. The algorithm to perform channel assignment is called Randomized One-point optimization

[8]Note that CTS-to-self may affect the behavior of clients in tests that use RTS-CTS packets. However, these probe packets may easily be replaced by Data-Ack packets that do not suffer from such problems

($RanOp$) and bears some similarity to the approach described in [90]. Note that we consider the cost of re-associating clients by minimizing the number of times the channel assignment algorithm is invoked. However, once invoked, the current channel assignment algorithm does not consider the re-association cost. An algorithm that does consider this cost can also be designed for the SMARTA system.

The $RanOp$ algorithm first assigns a random channel to each access point and computes the current total number of conflicts[9]. Then, considering each access point (a_i) in turn it computes the gain in utility (in terms of reducing the total number of access-point conflicts) by switching that access point to a different channel. It computes the gain in utility for the access point on all channels and selects the channel C that yields the greatest gain for a_i. It then checks whether changing a_i to C yields an improvement in utility that is larger than the best utility gain seen in the iteration so far. If so, (a_i, C) is labeled as the best improvement seen so far. Because the algorithm performs this operation across all access points, it selects the access point and channel change that yields the largest gain in overall utility. This process repeats until we reach a configuration where any further one-point alterations do not yield a gain in utility. Because the solution of the algorithm may depend on the initial assignment of channels to access points, we perform multiple runs of the algorithm and choose the best solution (in terms of utility) among them.

4.8.2 Channel Refinement

In the second phase of channel-assignment, we refine channel allocations as an optimization of the assignment we computed previously. Note, for channel refinement we only consider optimization of assignments that keep the number of inter-AP conflicts constant. Inter-AP conflicts are considered the most severe type of conflicts and those that are likely to persist over longer periods of time than conflicts involving clients. This is why we only consider them in the RanOp algorithm. For channel refinement, whenever we add a client to an AP (to construct the ACG), we try all other channels for that AP to see if we can reduce the total number of client conflicts, keeping the number of inter-AP conflicts constant. If such a channel is available (e.g., in the case of 802.11a), the access point is switched to that channel. If not, the access point remains on the same channel.

[9]Note that the algorithm only considers those edges in the conflict graph that correspond to interferers that actually carry data traffic.

Algorithm 1 wIR Power Control Algorithm

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

4.8.3 Power Control

Power control can be done quickly, even on a per-packet basis. However, two constraints make the power control problem challenging. First, power control needs to ensure that clients do not lose service by reducing an AP's power level by too much. Second, every alteration to access-point power causes the underlying ACG to change. Therefore, we need to re-compute (or refine) the ACG for every change in access-point power.

Our power control technique proceeds in two steps. First, we compute appropriate power levels for all access points, taking the change in the ACG into account. Second, we refine access point power-levels to allow the system to adapt to changes in the environment.

The algorithm for computing optimal power-levels (called *weighted Iterative Reduction* (wIR) and shown as Algorithm 1) proceeds as follows. Initially, all access points are set to transmit at maximum power and we compute the total utility of this configuration ($ComputeTotalUtility(A)$). Note, the algorithm re-computes this utility in every iteration, before performing the steps outlined further. In each iteration, the algorithm finds the access point that has the greatest number of conflicts ($MaxConflictAP(A)$). This is the AP whose sum total number of conflicts on all incoming edges from neighbouring APs is the greatest[10]. The algorithm then re-weights these incoming edges

[10]This particular algorithm does not consider client conflicts, though more sophisticated versions of it

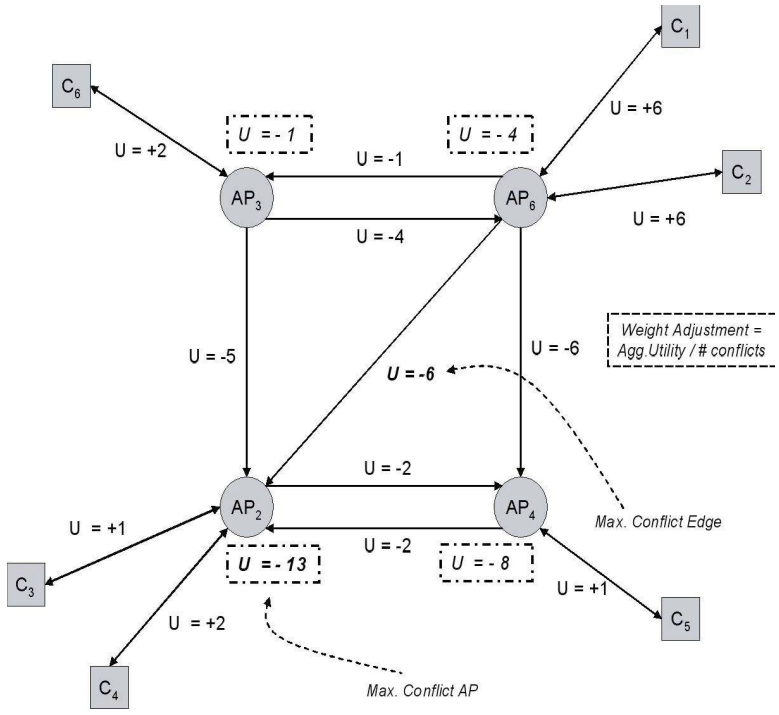

Figure 4.10: AP_2 is identified as the maximum conflict AP and the edge from AP_6 to AP_2 represents the maximum conflict edge, before edge re-weighting is done.

($AdjustWeight(\theta, z_i)$) as follows: For each AP that interferes with the maximum-conflict AP, the incoming conflict edge's weight is increased in proportion to the amount of utility that this AP provides to its clients (as shown in Figures 4.10 and 4.11). Thus, edge weights are adjusted by adding a $\frac{U}{E}$ positive value to the original weight, where U and E are the aggregate client utility provided and the total number of access point conflicts caused by the AP from which the edge emanates. After edge re-weighting, the algorithm selects the access point which induces the greatest conflict on the maximum conflict AP ($MaxConflictEdgeAP(\theta, Z)$), and instructs it to reduce its power level by one step ($ReducePowerLevel(\gamma)$). This 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). This approach rewards APs that have more active clients, so that they are less likely to have their power reduced.

can easily incorporate such information.

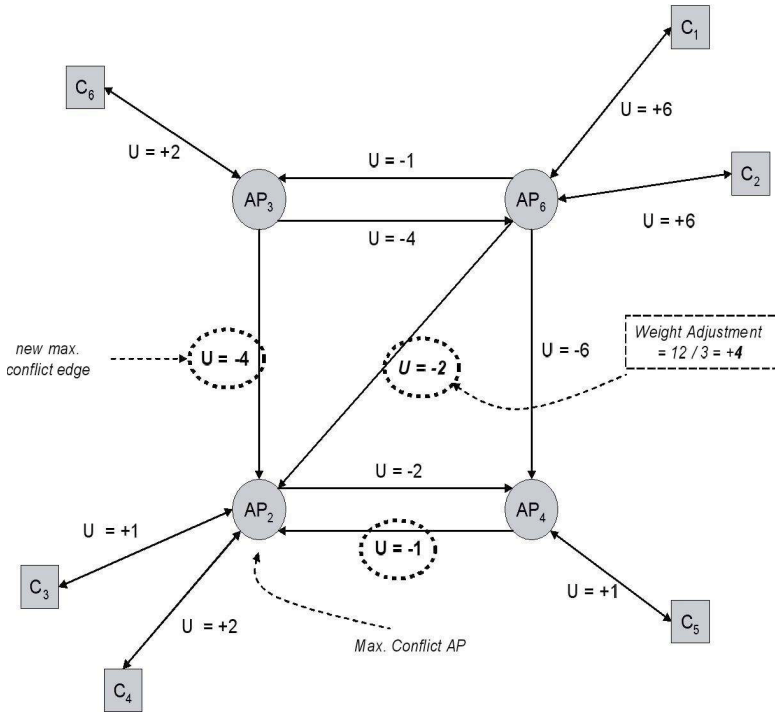

Figure 4.11: After edge re-weighting (shown in dashed circles), AP_3 is identified as the AP that has the maximum conflict edge to AP_2. AP_3 and AP_4 edge weights to AP_2 only change slightly because these APs provide very little utility to their clients.

4.9 Evaluation

We now present an evaluation of our architecture, evaluating interference estimation, optimization, and the ability to dynamically re-configure the network in response to changes in the wireless environment. We do not present a validation of the interference estimation approach and refer the reader to [27] for the details.

We first describe the simulation environment and network scenarios we considered in our evaluation and then discuss our results.

4.9.1 Methodology

Simulation Environment

We used the well-known QualNet simulator [13]. The central controller is emulated by means of a coordination component. Each access point houses two radios, and thus two MAC layers: A standard IEEE 802.11 compliant MAC layer and an *Environmental*

Figure 4.12: DC AP layout blueprint. Stars indicate AP locations.

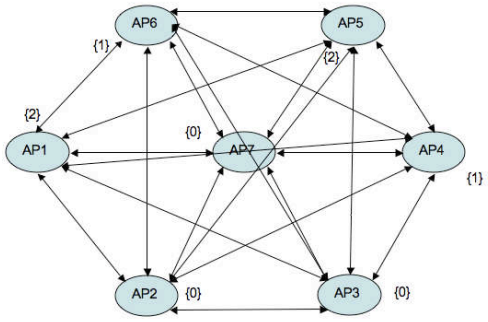

Figure 4.13: Circular (Star) topology conflict graph with channel assignment (using 3 channels) shown in curly brackets.

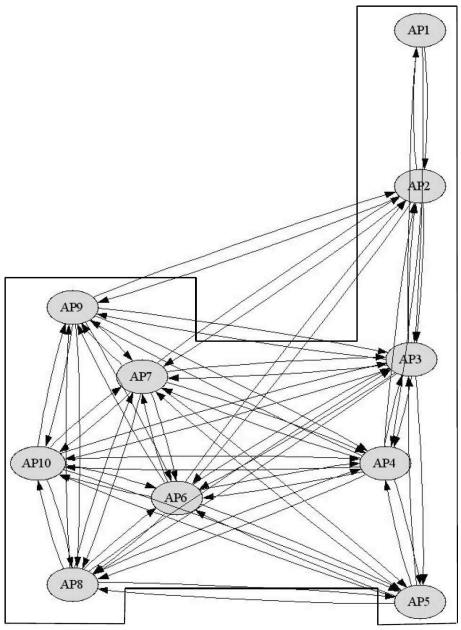

Figure 4.14: DC AP Conflict Graph at transmit power of 30 dbm. For clarity, figure only shows inter-AP conflicts.

Sensing (ES) MAC. The ES MAC supports the functionality for the probing agent. It periodically conducts the tests outlined in Section 4.7. Data is only sent on the 802.11 MAC. The clients implement the ES MAC in simulation for the sake of simplicity. We note that this precludes the need to 'clean' the environment for the interference tests, as described in Section 4.7. In practice, a client does not require multiple interfaces/MACs, and, in fact, can be completely unmodified.

Every 5 minutes, the central controller recomputes the current utility of the system[11]. If this drops by more than 20%, the controller instructs access points to re-initiate interference estimation tests and recompute the conflict graph. Because we are interested in aggregate network performance, 20% turned out to be adequate in distinguishing small-scale changes from large-scale changes. Using this information combined with statistics collected by passively sniffing traffic on the IEEE 802.11 MAC, the central controller re-runs the RanOp channel assignment and wIR power control algorithms. Once complete, the controller re-evaluates the utility of the system at the next scheduled time step.

We have focused on specifying utility as the throughput that a client obtains from its access point, with the goal of maximizing aggregate network throughput. The statistics we captured (on the IEEE 802.11 MAC) in order to compute this metric include informa-

[11] We chose a 5 minute interval because it suited the movement speed we picked for our mobility experiments

tion on access point load and the number of packets sent/received per second from each client[12]. Unless otherwise indicated, each client also implements Auto-Rate Fallback (ARF) and thus the data rate will likely change during the course of the simulation. Note that the current utility function for the client does not directly account for the data rate on the link, but infers it based on the number of packets received per second from the client. This utility function may not be ideal for the wIR algorithm where power reduction can affect the data rate on the link. In this regard, a more sophisticated utility function that takes data rate into account can be designed and tested with wIR.

Interference is also modeled in the utility function, and is assumed to have a log-linear relationship to the throughput received by clients, (i.e., we use the load of the interfering source to compute the degree of interference). Both parameters, throughput and interference are assumed to carry equal weight in the utility function. We used the two-ray ground model in our simulations [26]. For each scenario, we initiated CBR traffic from access points to clients with 512 byte packets. We evaluated two forms for our proposed optimization algorithms; one that only performs channel assignment (RanOp), and the other that also performs power control (RanOp-wIR). These were evaluated against the channel and static power configuration currently chosen by the network administrator for the building seen in Figure 4.12. Channels were assigned based on an extensive site survey that was carried out for the building. Moreover, to illustrate the benefits of our proposed centralized channel allocation algorithm, we compare it against a decentralized Least Congested Channel Search (LCCS) approach, discussed in [59]. LCCS is the current state-of-the-art algorithm for channel assignment and operates as follows: Each AP periodically observes data transmissions from other access points and clients on its channel. If the transmissions exceed a pre-specified threshold, it moves to a channel that is less congested. LCCS serves to show how well local tuning can perform in comparison to centralized channel assignment.

Simulation Scenarios

Our evaluation has three parts. In the first part, we present micro-benchmarks to illustrate the correct operation of SMARTA. In the second part, we simulate a large university building that we will call ‘DC’ (illustrated in Figure 4.12). This allows us to gauge the effectiveness of SMARTA in a more realistic enterprise environment. For this scenario, we assume clients are stationary and are continuously receiving traffic. Finally, we also present micro-benchmarks for client mobility. These micro-benchmarks allow us to observe the behavior of the SMARTA system in dynamic scenarios.

[12] We used EWMA to smooth out abrupt changes to each metric.

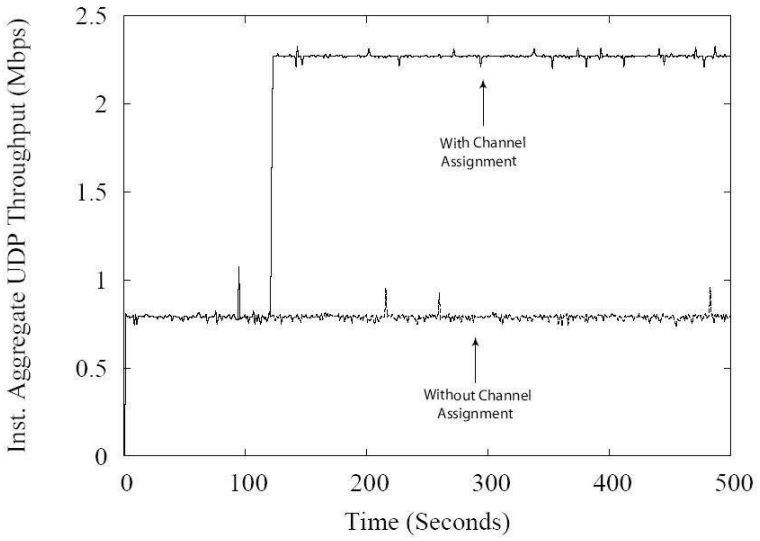

Figure 4.15: Instantaneous aggregate client throughput on Linear topology. Improvement seen is a result of channel assignment.

4.9.2 Results

We first discuss two micro-benchmarks to validate the correct operation of SMARTA.

Micro-benchmarks

Linear Topology: We first consider a simple linear topology with four APs. The transmit power of the APs is set such that an AP interferes both with adjacent APs and with neighbours of the adjacent APs. Clients are placed in between APs. Even if we consider just 3 channels, we can trivially produce a conflict-free colouring where AP channel assignment from left-to-right is given as (1, 6, 11, 1). This sequence can be repeated for an arbitrarily long AP chain, illustrating that linear topologies (typically found in hallways) are easier to address using just channel assignment, without power control. This result is shown in Figure 4.15. At $t = 120s$, when RanOp channel assignment is initiated, the aggregate network throughput improves significantly and remains steady thereafter. At this point, each adjacent AP is on a different channel and the number of conflicts falls to zero.

Circular (Star) Topology: Next, we consider a circular topology where both channel assignment and power control prove to be useful in optimizing network throughput.

The circular topology we considered is illustrated in Figure 4.13. If we use only 3 channels, and have access points transmit at a nominal power of $20dbm$, a channel as-

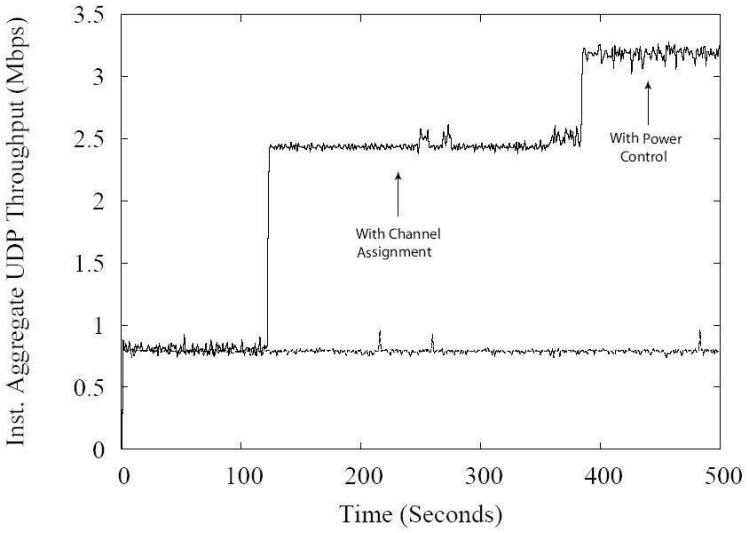

Figure 4.16: Instantaneous aggregate client throughput on Star topology. Initial improvement is because of channel assignment while the subsequent improvement comes as a result of power control.

signment for this topology will always yield solutions where at least two APs conflict with each other[13]. Thus, there are opportunities to improve network performance with the help of power control. This is illustrated in Figure 4.16. Improvements in throughput occur in identifiable stages where initially, all APs are transmitting using the same default channel. At $t = 120s$, SMARTA initiates channel assignment, producing the channel assignment shown in Figure 4.13. Note, because the topology considered here is a clique, a good channel assignment will equally partition APs across each of the channels, where the total number of conflicts is minimized. RanOp produces an assignment which maintains this property, resulting in a total of only 10 conflicts. This validates the ability of RanOp in finding good channel assignments for this topology.

At time $t = 250s$, wIR power control begins. At $t = 380s$, we observe a significant increase in aggregate network throughput (the cause of the delay is explained later). While, RanOp produces an assignment that is almost 200% better than the original default assignment, wIR further improves performance by almost 25%.

Note that wIR terminates if changes in power levels do not produce observable improvements. This requires us to observe the network after each change. We find an observation window of 3s to be suitable (which is the reason why power control operates

[13]Note, although RanOp may not produce the same assignment of channels to APs in each invocation of the algorithm, the sum total number of conflicts across all assignments remains the same.

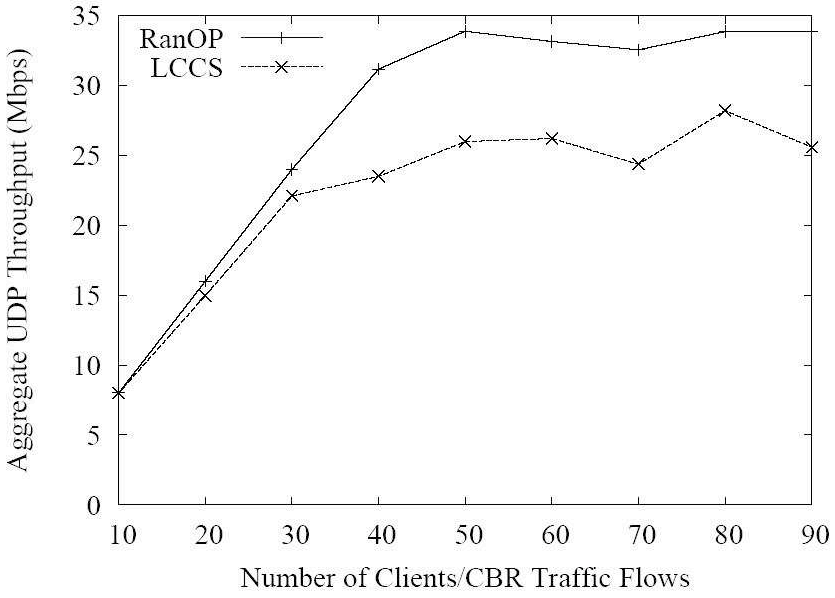

Figure 4.17: Aggregate client throughput at 30 dbm using 12 channels

on slow timescales). Of course, the accuracy of the observation is a function both of the length of the observation window and the dynamic nature of the wireless environment. This is a tuning parameter for the system and can be set based on the environment under consideration. Power control also requires an up-to-date ACG upon each iteration, which can incur an additional overhead. In Chapter 6, we show that this overhead is acceptably small and is on the order of a few seconds in the worst case.

A More Realistic Scenario

We now discuss results of running SMARTA on the DC topology (Figure 4.12). For these results, we randomly distribute clients within the coverage radius of each of the access points, whose size is determined by the transmit power of the access point. We analyze the performance of SMARTA on scenarios exhibiting a high degree of interference. Note that the degree of interference is affected by the transmit power of the APs/clients, the number of clients, and the client distribution [36]. While the transmission power of access points is tunable and controllable, client density, distribution, and power are not. Therefore, in order to independently study the effects of each, we decouple them in our simulations. For our results, we use the metrics of aggregate network throughput and per-

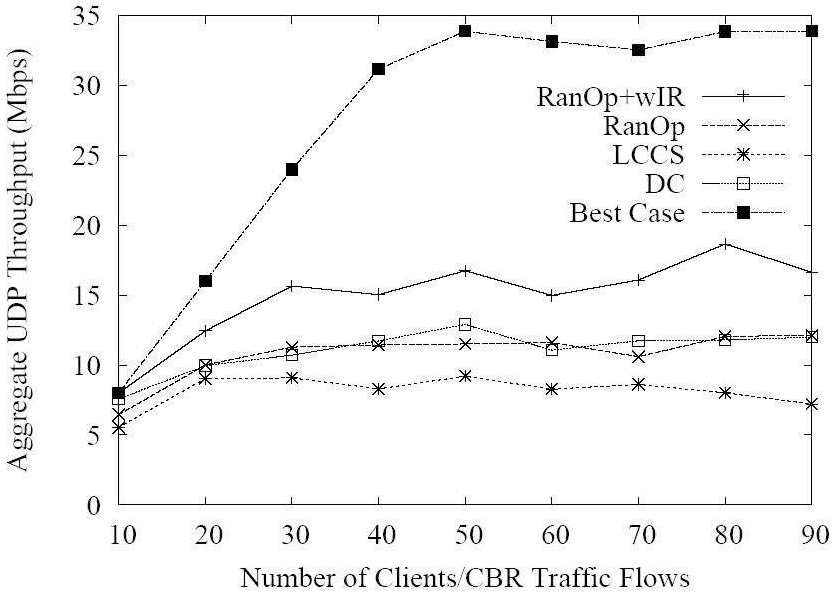

Figure 4.18: Aggregate client throughput at 30 dbm using 3 channels

packet delay to compare the different algorithms. We have also analyzed the distribution of flow throughputs across clients, the details of which are provided in [27].

In our scenarios, APs transmit at 30 dbm and clients transmission power is set equal to that of the APs to facilitate connectivity even at coverage boundaries. We use 30 dbm to stress test our system. We have analyzed the performance of the algorithms in low power scenarios (i.e. 20 dbm) as well and obtained similar results.

Referring to Figure 4.14, we see many access point conflicts. There are also client conflicts, not shown for clarity. With 802.11a (i.e 12 orthogonal channels), we can trivially eliminate all conflicts by assigning an independent channel to each AP. In this situation, the best possible solution is to assign a separate channel to each AP and setting each APs transmit power to maximum. We call this configuration 'best', and use it to benchmark solutions generated in other scenarios.

Throughput: Figure 4.17 shows aggregate client throughput against client density, for the case where we have 12 available channels. The RanOp curve corresponds to the 'best' curve since we observed that our algorithm always produced conflict free assignments in this scenario. Because no power control is required in this scenario, RanOp-wIR performs identically to the best case. We observe that, at high densities, due to its decen-

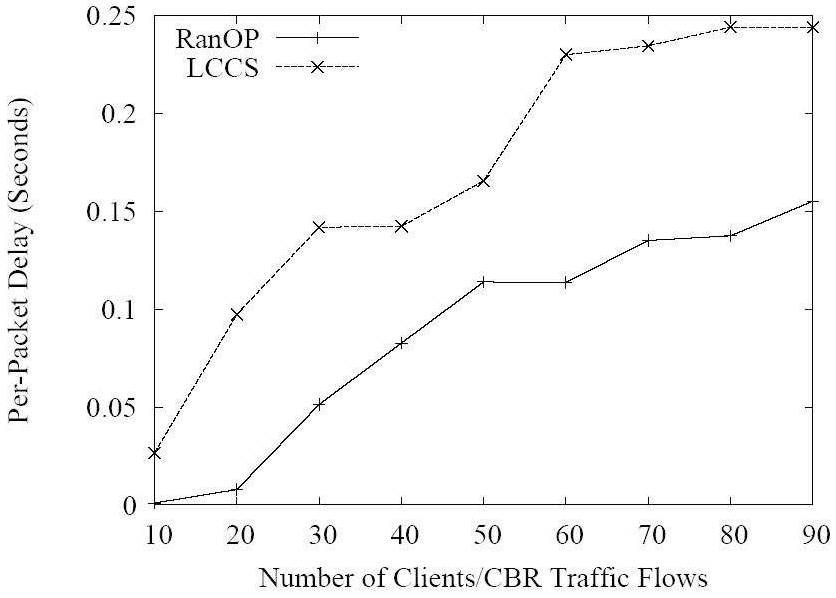

Figure 4.19: Per-packet delay at 30 dbm using 12 channels

tralized approach, even with 12 channels, LCCS is unable to optimally assign channels to access points. This is because LCCS is AP centric in nature and does not consider client conflicts when picking the best channel for the AP. Of course, in low density environments, LCCS performs close to the best case because of the lower degree of interference.

Figure 4.18 shows aggregate client throughput for the 3 channel case (the 'best' curve is shown for reference). Not surprisingly, aggregate client throughput drops significantly for all the algorithms. However, observe that RanOp combined with wIR performs the best in this scenario. Because channel assignment cannot eliminate all conflicts, power control yields further improvements. However, there is still a significant performance gap between the 'best' curve and our algorithms. Aside from the limited number of channels, this is because of the limitations of power control. Channel assignment has the ability to eliminate all types of conflicts (i.e., zero, one, and two-hop conflicts) whereas power control can only address OAP and zero-hop conflicts. This is because of the inability to adjust client powers. As a result, even a provably optimal power control strategy may ultimately be unable to eliminate all conflicts in such cases. Nevertheless, we still observe significant improvements over LCCS.

We also plot the performance curve for the hand-tuned DC channel configuration

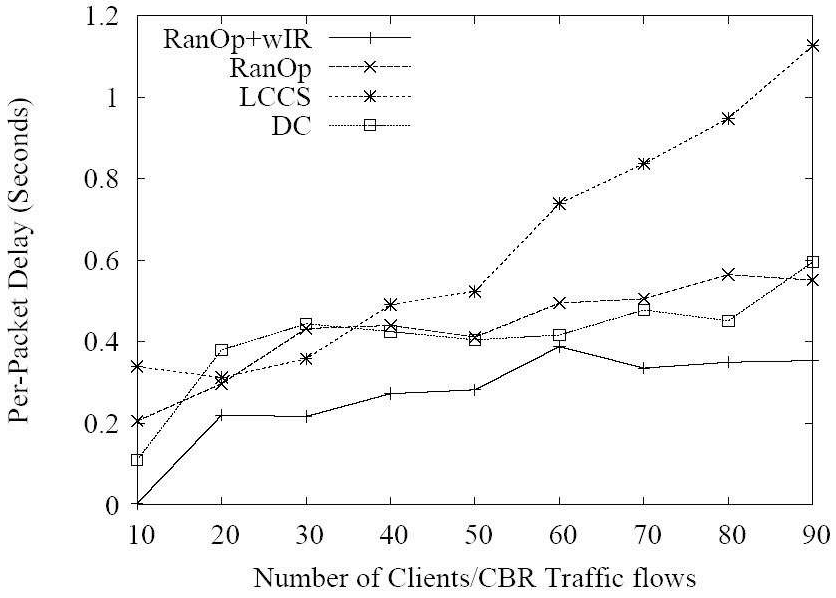

Figure 4.20: Per-packet delay at 30 dbm using 3 channels

(i.e., the configuration decided by the network operator at our university)[14]. This configuration performs similar to the RanOp algorithm used in SMARTA. The reason that RanOp does not yield significant improvements over the hand-optimized DC assignment is because of the large number of conflicts. The number of conflicts significantly decreases the number of possibly good configurations that yield high throughput. Nevertheless, we still observe that in these scenarios, RanOp is able to perform just as well as a carefully hand-optimized channel assignment and 50% better with the addition of transmit power control.

Per-Packet Delay: We also analyze per-packet delay for each of the algorithms. Per-packet delay is a crucial metric for delay-sensitive applications such as voice and multimedia. Note that the results we discuss here use the same throughput maximization utility function as was used for the previous results. We expect a utility function catering specifically to per-packet delay to perform even better.

Figure 4.19 plots per-packet delay results against client density, using 12 channels. The results for RanOp and combined RanOp-wIR are identical. We observe a signifi-

[14]The hand-tuned DC configuration made use of only 3 channels (i.e., it operates on 802.11b/g). Therefore, we were not able to show its results for the 12 channel case

cantly lower per-packet delay for the RanOp algorithm than that for LCCS. Moreover, the delay values for RanOp are almost always below 150 ms across the board. This is an interesting result since the delay budget for most voice applications falls within this range. Thus, we believe that the centralized RanOp algorithm is well suited to supporting such applications even in very dense scenarios characterized by a large number of AP/client conflicts.

Figure 4.20 presents similar results for the 3 channel case. Not surprisingly, per-packet delays have increased over the 12-channel case due to increased MAC contention delays and a larger number of collisions. However, we observe that RanOp-wIR provides the lowest per-packet delay primarily because power control reduces APs collision domains significantly, thereby reducing MAC contention. LCCS performs the worst in this case with per-packet delays of over one second in very dense environments, demonstrating its limitations in these scenarios.

Effect of Mobility

We analyze the impact of mobility on the SMARTA system. Recall, SMARTA triggers re-computation of access point configurations if the change in utility is significant, (i.e., exceeds a predefined utility change threshold α). For the purposes of our simulation, we set this threshold to 20%.

We construct two scenarios to analyze the impact of mobility. Note that these scenarios assume nomadic clients that use the network while stationary at a particular location. This is in contrast to mobile clients that use the network while on the go. In the first scenario, a client moves between a set of access points, as shown in Figure 4.21. This is typical for an employee that might periodically go for meetings to offices of fellow employees. We use this scenario to illustrate the stability of SMARTA in reacting to small-scale changes that may occur in the environment. In the second scenario, clusters of users move from different access points to a common access point. This is likely to occur in situations where groups of people gather together for a scheduled meeting and represents a large-scale change that SMARTA must handle.

Small-Scale Scenario: Figure 4.21 illustrates the user mobility pattern considered in this scenario. A single user starts at AP_1 and moves between access points, finally ending up in between them. Note, the user disconnects and re-connects with AP_4 even during movement step 3. Changes in aggregate network throughput are illustrated in Figure 4.22. Before the initial move, at $t = 120s$, SMARTA computes optimal channel and power level configurations for the access points, causing the aggregate throughput

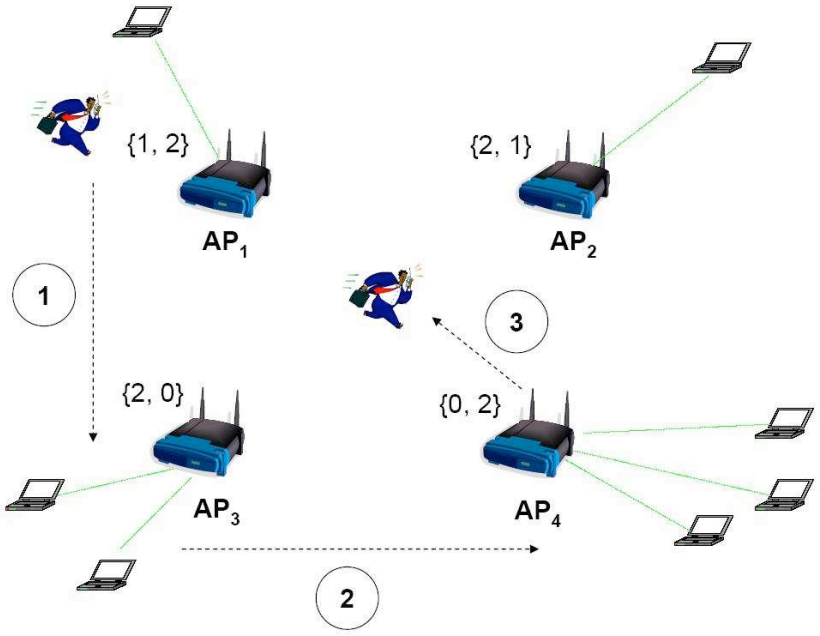

Figure 4.21: Micro-benchmark setup for analyzing the impact of mobility. User mobility is shown as dotted arrows with labels indicating the steps followed by the client. For the large-scale scenario, channels are shown in curly brackets where the numbers (left-to-right) depict assignments before and after a large-scale change.

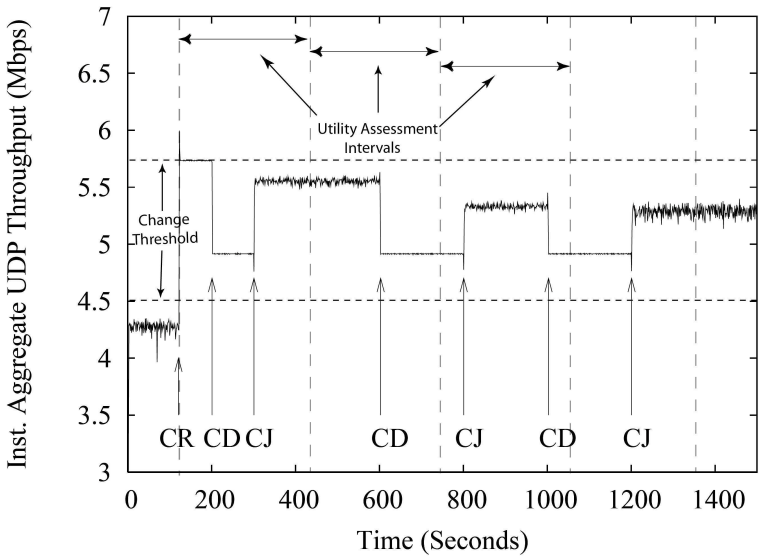

Figure 4.22: Instantaneous aggregate client throughput in small-scale scenario. Following events are shown: CR=Configuration Re-computation, CJ=Client Join, and CD=Client Disconnect. Utility assessment intervals shown as vertical dotted lines and change threshold shown as horizontal dotted lines.

to increase to approximately 6 Mbps. At time $t = 200s$, the user disconnects from AP_1 and connects to AP_3 at $t = 300s$. During this interval, the utility drops by approximately 16%, which is not below the change threshold and increases again at $t = 300s$. Thus, at $t = 420s$, when utility re-assessment is done, SMARTA does not initiate re-computation of channels and power levels. This process successively repeats without the utility change ever falling below the change threshold. In summary, we observe that SMARTA's use of utility-based triggers allows it to be resilient to oscillations that may occur as a result of small-scale changes in the RF environment. This is particularly crucial for legacy clients that may be affected by continuous changes in access point channels.

Large-Scale Scenario: For this scenario, we use the same setup as was used in the previous scenario. However, in this case, two groups of users move from separate access points to a common access point. Channel assignments based on initial user distribution are shown in Figure 4.21. At $t = 120s$, SMARTA performs optimal channel and power level assignment for all access points. At $t = 400s$, all clients from AP_1 and AP_4 disconnect and proceed to move towards AP_3. At $t = 600s$, all clients connect to AP_3, subsequently increasing its load. Utility re-assessment between the time when clients disconnect and re-connect is disabled to illustrate the effect of re-configuration af-

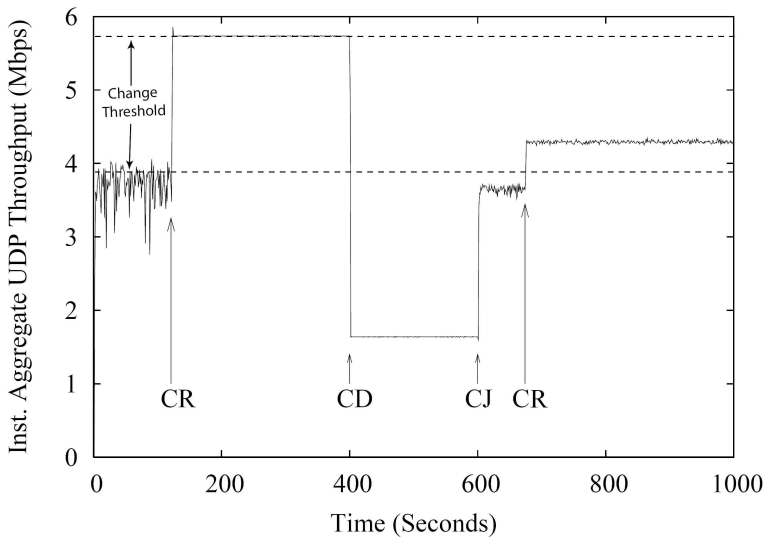

Figure 4.23: Instantaneous aggregate client throughput in large-scale scenario. The events shown are similar to those illustrated in Figure 16.

ter clients re-connect to the network. In reality, SMARTA would already account for this case during periods of disconnection as it will observe a large decrease in utility and re-assign channels and powers as a result to maximize utility for currently connected clients. At $t = 600s$, when clients re-connect, an increase in utility is observed. Note, SMARTA is only aware of the utility that was computed at $t = 120s$, during the last time re-configuration was performed. At $t = 680s$, a significant drop is observed and SMARTA re-initiates computation to improve system utility. Notably, the utility improvement is not very significant and in particular, does not match the utility of the configuration at $t = 120s$. This is due to the large number of clients connected to AP_3 and the excessive load on it. This reduces per-client throughput and contributes to the drop in aggregate client throughput even after the configuration is refreshed.

The scenarios outlined above provide insight into the ability of the SMARTA architecture to accurately determine the type of change that occurred in the environment. The utility change threshold is a tuning parameter for our system and can be set to suit the needs of the deployment environment.

4.10 Related Work

The work on the SMARTA architecture spans a wide range of research problems that have been independently studied in the prior literature. Furthermore, it is also relates to

prior research proposals on managing interference in enterprise WLANs. Since we have covered the latter in detail in Chapter 3, we focus only on sub-problems that are also addressed for the SMARTA system.

Interference Detection: Interference detection has been well-studied in the literature [36, 50, 72]. However, most of these techniques infer interference using higher layer (e.g. NET/MAC layer) statistics that are impacted by multiple physical layer RF phenomena [107]. Therefore, the accuracy of these approaches in detecting interference is limited. In contrast, Qiu et al. [100] adopt a trace-driven simulation approach in which 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. Similar to the ideas in this work, Padhye et al. [98] propose an approach of running controlled pairwise experiments to detect and quantify RF interference. However, their approach requires injecting synthetic flows into the system and can take a considerable amount of time to run, making it infeasible for use in online networks (the details of this approach are discussed in Chapter 6). In contrast, we show that it is possible to run simple and efficient tests on-the-fly to accurately detect RF interference.

Channel Assignment: The most common technique to mitigate interference in an enterprise WLAN is to perform channel assignment. AP channel assignment has been studied extensively in the literature [45, 77, 90] and is a well-known NP-hard problem. A number of heuristics have been proposed for this problem [45, 90]. For example, Mishra et al. [90] use a randomized search algorithm that incorporates client interference in the channel assignment process. We adopted similar techniques in our system.

Transmit Power Control: Transmit power control also has a significant influence on the performance of a wireless network [78]. Optimal power-level assignment is similar in hardness to channel assignment, and, for the coverage planning problem, has been shown to be NP-complete [28]. Many techniques have been proposed in the literature for computing power levels for the access points [36, 39]. We also propose a heuristic that we show works well in optimizing the performance of our system.

4.11 Discussion

We now briefly comment on the scope of SMARTA. In this chapter, our goal is to design a system that practically applies the theory of conflict graphs to manage interference in an enterprise WLAN. As part of its design, SMARTA features the following:

- *Fine-Grained Control:* The correctness of the interference tests hinge on the abil-

ity to tightly *synchronize* APs and precisely control the timing of actions at these APs. Such tight control not only requires elimination of potential latencies while implementing the system, but also a comprehensive WLAN design. We discuss the details of how this can be achieved in subsequent chapters of this dissertation.

- *OmniDirectional Antennas:* SMARTA is designed for WLANs where the APs transmit using omnidirectional antennas. This is crucial to maximize the effect of silencing. However, transmitting the probes themselves can be done using directional antennas.

- *Low-level Signal Information:* Carrier-sensing interference in SMARTA is discovered using energy-on-the-air measurements. This requires the AP's radio to report such energy measurements to the controller. Depending on the hardware platform, this may or may not be possible. Moreover, silencing is crucial for such a test, as background transmissions could be mistaken for the interfering AP's signal.

- *Additional Optimization Parameters:* Aside from frequency selection and power control, other parameters such as CCA tuning and association control can also be integrated into the SMARTA system. Recent work has shown that these additional optimization techniques are likely to improve the performance of the WLAN system even further [47]. Our goal in this work was to highlight the benefits of the interference measurement framework proposed for SMARTA, instead of designing a comprehensive WLAN system.

In this chapter, we focused on the design and prototyping of SMARTA. In subsequent chapters, we take the design of SMARTA's interference measurement system and implement it on an enterprise-scale WLAN testbed. Furthermore, we integrate the measurement system into two optimization schemes and show how it can *practically* improve performance for a variety of different applications.

Chapter 5

The Platform: An Enterprise WLAN for Centralized Control

In this chapter, we describe our efforts towards the design, deployment and maintenance of a wireless testbed for evaluating algorithms for centralized control. The testbed is used to implement and test the interference measurement and management algorithms explored in this dissertation. The rest of this chapter is organized as follows. We outline the motivation and design goals for the testbed in Sections 5.1 and 5.2. We discuss alternative design choices we explored for the testbed in Section 5.3. The hardware and software details of the testbed are discussed in Section 5.4. Finally, we present some performance results for the testbed in Section 5.5 and conclude the chapter in Section 5.6.

5.1 Motivation

Practical research in wireless networking necessarily involves field experimentation. This is because existing RF models are far from adequate in capturing RF properties such as propagation and interference. This is especially true for indoor environments, where RF signals experience a great degree of multi-path fading, due to reflection, diffraction and scattering effects (as discussed in Chapter 2). Moreover, a seminal paper by Kotz et al. [79] shows that evaluating wireless protocols (and algorithms) via simulation provides little insight into their practical performance because most simulators poorly model real-world RF effects [13]. Because this dissertation focuses on designing practical interference management techniques, we take the ideas developed in Chapter 4 and evaluate them on an enterprise-scale WLAN testbed. Specifically, we deploy a 38 node wireless

testbed across two floors of our Computer Science building, at the University of Waterloo.

Numerous wireless testbeds have been proposed in prior work [44, 46, 51] and the corresponding insights have added significantly to our understanding of how to design such systems. In this chapter, we argue that testbed deployment guidelines depend on the architecture of the network under consideration. For instance, distributed WLAN architectures have fundamentally different requirements as compared to centralized architectures. For centralized WLANs, network optimization is handled by the central controller. Therefore, the delay and delay jitter on the path from the controller to the air-interface of the APs affects the ability to correctly perform such optimizations. In this regard, we are interested in experimentally evaluating the feasibility of centralized control, for applications such as traffic scheduling [35] and data rate adaptation. As a result, we assume both a centralized control and data plane. Therefore, the central controller is ideally co-located at the edge router through which all wireless traffic is aggregated.

As discussed in Chapter 3, numerous enterprise WLAN vendors [8, 1, 22, 19] also embrace the centralized WLAN design. However, our private discussions [34] with some reveal that they typically use special-purpose hardware/software for centralization. By contrast, our objective is to determine whether it is possible to realize such centralized control using off-the-shelf commodity hardware.

5.2 Design Goals

We begin by listing the design goals for a centralized WLAN testbed. We subdivide the discussion into goals for centralized control, and those that are necessary for any testbed.

High Throughput: A centralized data plane necessitates a high throughput backbone that connects the central controller to all the APs. Such centralization forces traffic to flow from a single aggregation point (which is the controller/edge router). This causes traffic to be concentrated on a small set of egress links sourced at the central controller. These links (as well as the controller) should be capable of handling the capacity demands typical of such a WLAN deployment, e.g. up to 200-400 simultaneously active users for a moderately-sized deployment [114].

Tight Centralized Control: Centralized WLANs are motivated by the desire to move complexity from the APs to the central controller. Operations such as frequency selection, power control, data rate adaptation and packet scheduling can potentially be performed centrally. This requires tight centralized control, which consists of: 1) Ensuring that the paths from the controller to the APs are of low latency (for fine-grained

centralized control), and 2) Ensuring that the actions at the APs are tightly synchronized (to correctly coordinate their actions). These require low delay and delay jitter on the path from the controller to the AP's air interface.

Advanced Radio Management Features: As discussed above, centralized WLANs are designed to manage the configuration parameters of the APs in a centralized fashion. Some of these parameters, such as the contention-window size, carrier-sense threshold, and other medium access parameters require lower-layer (firmware) access which is hard to obtain (due to licensing restrictions) on commodity radios. Therefore, we require a radio platform where we have the greatest degree of flexibility in tuning these parameters for the AP.

Real-time Traffic Monitoring: Another crucial component in the design of centralized WLANs is support for real-time traffic monitoring by the central controller (via the APs). This allows the controller to track client performance and react to changes that reduce throughput and lead to poor network connectivity. A key concern here is that we should be able to monitor traffic at potentially high rates and with low overhead. A low overhead approach would minimize the chance that the monitoring traffic interferes with other data traffic also flowing in the network.

We now discuss a set of requirements that are necessary for any testbed.

Standardized Hardware: In our work, we strive to build a testbed that mimics a real-world WLAN deployment. In doing so, it becomes easier for any network designer to interpret our results and map them to other WLANs deployed using similar hardware. For this purpose, we require the use of off-the-shelf commodity hardware that is easily available and in widespread use today. Note that commodity platforms are those based on open standards, are cheaply available, and in widespread use by the industry. However, depending on their functionality, some commodity platforms may or may not expose certain tuning parameters for the radio.

Ease of Management: An important requirement for any wireless testbed is that it should be easy to deploy and manage. Nodes should be deployed without an extensive site survey. Furthermore, the network should ideally be configured from a single location and any changes and updates should propagate to respective nodes. Nodes should also be rapidly (re-)configurable and support many operational modes (e.g. APs, clients, or sniffers).

Non-Intrusive Hardware: The hardware platform should also be non-intrusive, as argued in [46]. In other words, nodes should not take up too much space, make too much noise or generate too much heat so as to disrupt on-going activities in their surroundings. Furthermore, for security reasons and to ensure minimal hardware tampering,

nodes should be placed in closed offices/rooms.

5.3 Alternative Design Choices

During the design of our testbed, we evaluated a number of possible platforms for our nodes and weighed them against the requirements outlined in Section 5.2. In this section, we briefly outline three platforms and state why they are ill-suited for our centralized testbed.

Off-the-shelf APs:: We considered off-the-shelf, configurable APs for our testbed [2, 7]. While this appears to be a compelling choice, there are numerous drawbacks of such a platform. The primary concern was the lack of access to lower layer functionality. Off-the-shelf APs only allow a few parameters to be tuned through a simplified web-based interface. Some manufacturers support open source platforms such as OpenWRT [10] and some testbeds [84] use these as their nodes. However, OpenWRT does not allow access to the actual radio's firmware, making it limited in functionality relative to the platform we describe in the next section. It is worth noting that there have been recent efforts to make the firmware for some radios (e.g. Atheros) more openly available. However, these efforts are preliminary and support exists for only a few platforms.

Low Power Embedded PCs: These PCs include the Soekris net4826 platform [14]. This is a single-board computer with a 266 MHz processor and 128 Mbytes of SDRAM. It has two mini-PCI slots and is priced at $200. It also supports Power-Over-Ethernet (PoE), allowing the device to be remotely rebooted by disabling/enabling the Ethernet interface. However, the Soekris platform has some shortcomings. For instance, our experiments revealed that the Soekris becomes unstable when the wired-to-wireless traffic load on the node is high. Furthermore, the platform is primarily built for low power environments. Thus, there are a number of power saving features that are incorporated into the platform. For example, the auto-halt feature for the Soekris powers down the CPU when the number of interrupts per second decreases below a particular threshold. This is undesirable for our system because it violates the tight centralized control requirement outlined in the previous section.

Laptop PCs: Another possible choice for hardware platform was to use laptops as nodes. This approach has been used by Draves et al [56]. There are three reasons why laptops are not well-suited for our system. (1) Compared to single-board computers [14, 15], laptops are not as customizable, especially lower-end models where most peripherals are integrated onto the mainboard to save cost. (2) Laptops are more likely to be vandalized or stolen than small embedded PCs that are less familiar to most people.

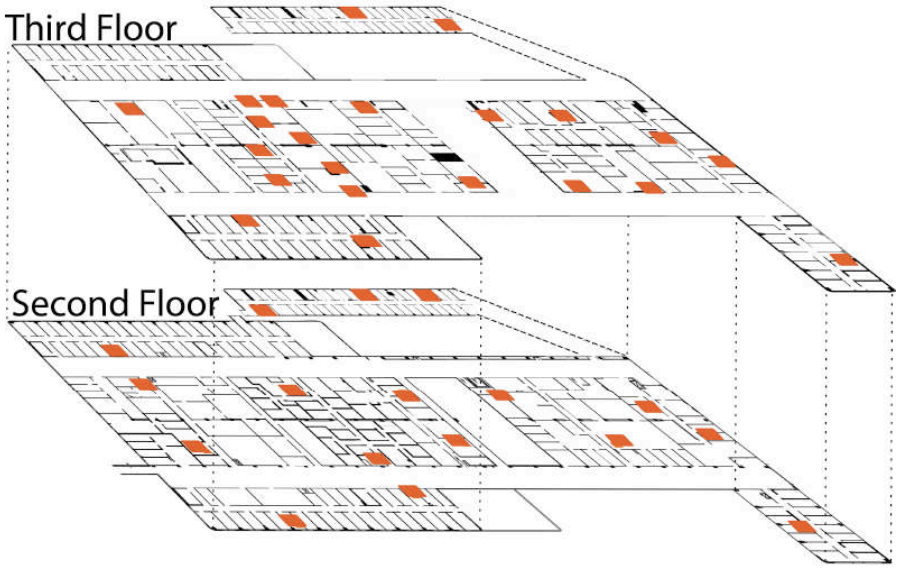

Figure 5.1: AP locations are shown as the orange squares in the figure.

(3) Depending on the laptop's hardware, it is likely that it may generate a significant amount of noise (due to the use of on-board CPU fans). As deploying non-intrusive hardware is an important requirement, laptops are not suitable for our testbed.

Next, we present the details of the hardware and software that we subsequently chose for our testbed and discuss how it meets the requirements presented in Section 5.2.

5.4 Design

Our testbed is deployed across two floors of the William Davis Centre building at the University of Waterloo. It comprises a total of 38 nodes. The layout of the nodes is shown in Figure 5.1.

5.4.1 Hardware

Our WLAN testbed operates on the existing wired backbone of the computer science department. We assign a separate VLAN for the testbed which is used to route traffic to and from the nodes and the controller. Therefore, the only hardware we require is the central controller and the testbed nodes (as shown in Figure 5.2). We now discuss these two pieces in greater detail.

Controller: The central controller is implemented on a desktop PC. The desktop is a dual core 2.66 GHz machine with 2 GB RAM and Gigabit Ethernet connectivity. The desktop serves two purposes. It acts as a central NFS server for the nodes that remotely boot over the network. It also functions as the central controller for the centralized WLAN that configures and manages the APs of the testbed[1]. We discuss the software details of the controller in Section 5.4.2.

APs: Each of our APs is a VIA EPIA EN12000EG mainboard (having a 1.2 GHz $C7$ nanoBGA2 processor) with 1 GB of DDR RAM. This platform is considerably more powerful than the Soekris platform described earlier. It not only allows our nodes to support high throughput and tight centralized control, but the larger memory also avoids expensive disk I/O potentially caused by paging. In addition, the mainboard does not contain any fans but instead dissipates heat via a large heat sink that sits atop the processor. This eliminates the noise factor almost entirely.

The VIA EPIA EN12000EG mainboard also features Gigabit Ethernet. Furthermore, to create Gigabit links from the central controller to each AP, many of our nodes are plugged into Gigabit Ethernet drops in the wiring closets[2]. Each wiring closet is subsequently connected to all others using optic-fiber lines.

To log wireless trace data, we use a 40 GB Toshiba IDE hard-drive installed at each node. Some prior work uses diskless nodes that only mount an image from the NFS server [46]. We use a local hard drive for data logging to prevent trace collection from generating wired (NFS) traffic that may interfere with our experiments. Note that the speed at which we log data is still limited by the maximum I/O interface bandwidth supported by the hard drive. In fact, our initial experiments revealed that the default disk access mode on our hard drives (termed Programmed Input/Output or 'PIO') was insufficient for high-speed data logging. Therefore, we modified this access mode to Ultra-DMA which substantially improved disk I/O performance, allowing us to log data at rates higher than the link speed supported by the wireless radio. However, we note that use of a hard drive may not be ideal because hard drives are prone to failure. Nevertheless, because the hard drives function only as local stores, their failure is not catastrophic because the primary filesystem is mounted remotely via NFS.

One drawback of the board we chose is that it does not come with integrated mini-PCI slots but instead has only one PCI slot (the wireless cards we would like to use only

[1]Ideally, these two functions would be separate but for the sake of simplicity, we merge them into a single host on our testbed

[2]We were not able to connect all APs to Gigabit Ethernet drops due to the limited number of such drops. Instead, some APs were plugged into 100 Mbps Ethernet ports

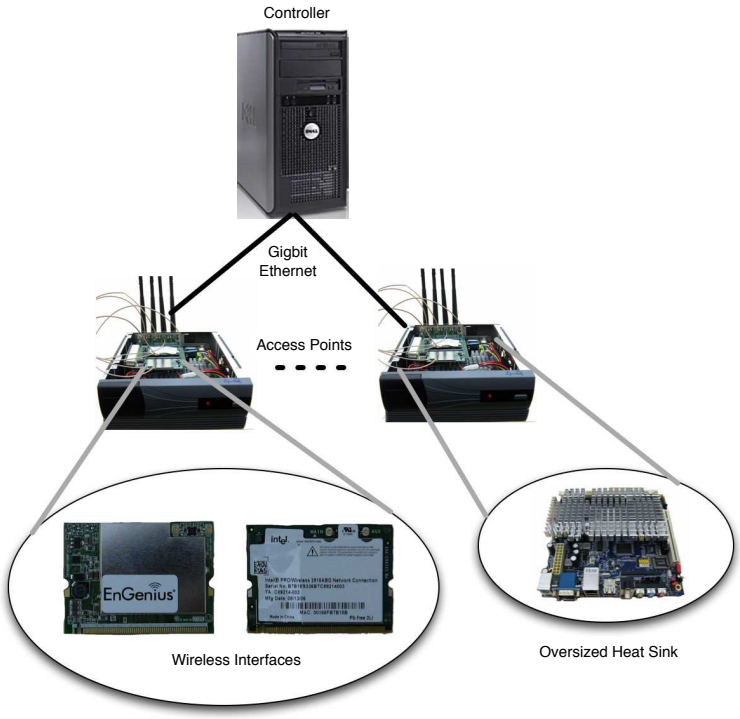

Figure 5.2: An Overview of our testbed architecture and it's components

support mini-PCI). Therefore, we cannot attach two wireless interfaces on the board[3]. To attach two mini-PCI wireless interfaces, we equip each node with a Routerboard PCI-to-mini-PCI adapter. This adaptor allows us to attach up to four mini-PCI cards on the node. The drawback of this setup (and other similar platforms) is that co-located wireless radios can interfere with one another and therefore must be shielded. In our current setup, only one interface actually transmits data while the other only passively sniffs traffic. Therefore, it is not necessary to shield the two wireless cards from one another.

The radios we use in our testbed are described next.

- **Intel 2915ABG Card:** We use the Intel 2915ABG wireless card (with the *ipw2200* driver) to act as the AP or client (depending on the node's configuration). Through our partnership with Intel, we have firmware access for this card and can tweak low-level parameters not exposed by other commodity radios. For instance, we can dynamically adjust the Carrier Sense Threshold (CST) for the radio. To support functionality for centrally measuring interference, we modified the firmware to

[3]We require two interfaces, one for transmitting data and one for capturing traffic

support disabling of binary exponential backoff and sending of CTS-to-self packets to silence the medium.

- **EnGenius EMP-8602 (Atheros) Card:** The secondary radio serves two purposes. First, it allows us to debug the operation of the Intel radio by observing packets that it sends out. Second, it allows us to passively monitor traffic in realtime to observe the performance of links in the network. After experimenting with numerous cards, we found the EnGenius EMP-8602 card (based on the Atheros chipset) to be the best suited for this purpose. We use the *madwifi-ng-r2657* driver with this card, which exposes a significant amount of information on captured packets. In addition, because MADWiFi is open-source and supported by a large community of users, new features are constantly added and bugs fixed over time.

One limitation of our hardware platform is it's power consumption. Our nodes are more powerful than the Soekris boards described earlier. Therefore, they draw more power as well (approximately 18 W at peak power). This precludes use of Power-Over-Ethernet (POE) and we therefore require both a power source and an Ethernet connection at locations where the nodes are installed. However, because we place our nodes in closed rooms/offices (for security reasons), finding a power source close to them is not too hard.

5.4.2 Software

The software architecture of our testbed is designed to support a robust and remotely manageable system that provides easy-to-use tools for quickly configuring the network. We briefly discuss this architecture next.

Node Software: Our testbed is configured such that nodes can remotely boot over the network via NFS. This has a number of advantages. (1) Package/Software updates on the nodes requires only the NFS mounted image to be updated[4]. In our work, software/code updates are frequent and involve making changes to the kernel/driver and userspace code. Individually updating software at each node is tedious, cumbersome, and prone to error. (2) It is possible to experiment with different (Linux) kernel versions that different network cards support. Using NFS, we can effortlessly switch between different Linux kernels for experimentation. (3) The 1 GB of DDR RAM allows each node to store the NFS mounted kernel and filesystem ($\sim$600-700Mb in size) in main memory, thus minimizing the amount of NFS traffic that would be generated from memory pages

[4]It is not necessary that all nodes mount the same image. Some may mount a different image, based on the requirements for the experiment

being swapped to disk. Nevertheless, the filesystem still needs to be periodically synced via NFS with the central server.

Server Software: The central server runs Ubuntu 7.04, with the 2.6.20-15 kernel. It runs the TFTP, BOOTP, and NFS daemons to support remote booting and NFS for the nodes. The server maintains a database of MAC address→IP mappings for all nodes so that each node's wired Ethernet interface is assigned a unique IP address from the 192.168.1.∗ subnet. Wireless interfaces on the nodes are also assigned IP addresses similarly, but from the 192.168.2.∗ subnet. Nodes are configured to use PXEBoot to boot over the network and download the kernel via TFTP. Once the kernel is booted, the filesystem is mounted (via NFS) from the central server.

Handling Failures: Recall that our nodes do not support Power-over-Ethernet. Therefore, in the event that a node crashes or hangs during an experiment, we cannot power cycle it by activating/de-activating the wired Ethernet interface. Instead, we use the hardware watchdog on the VIA EPIA EN12000EG mainboard. The hardware watchdog is comprised of two parts, a hardware countdown timer and a userspace software daemon. The countdown timer's job is to count down to zero, starting at a chosen initial value. Once it reaches zero, it performs a hardware reset. The software daemon operates in userspace and resets the timer to prevent the hardware from resetting. It does this by periodically writing to a hardware register that returns the timer to its initial value. If the OS hangs, the timer reaches zero and causes a reboot. Note that the OS could hang during boot-up as well. To allow a reset in this state, we statically compile the watchdog into the kernel and ensure that it is the first service to load during boot-up. Finally, as an added safety feature, we also install a userspace monitoring daemon (at the node) that periodically checks for successful connectivity to the NFS server and reboots the node if it fails to ping the server after a number of attempts.

Network Management: One of the cornerstones of building an easily manageable testbed is streamlining the process of configuring and managing the nodes in the testbed. We implement standard testbed tools to facilitate network management. The tools are divided into those that check the current configuration of the nodes and those that modify these configurations. The modifications range from making configuration changes to the wireless interface to initiating an entire system reboot. These testbed tools operate partly on the nodes and partly on the central server.

Wireless Traffic Monitoring: As discussed earlier, an important part of centralized control is to enable high-speed logging of wireless traffic. We covered the hardware aspects of such logging in the previous section and now discuss the software aspects. Traffic logging is split into two parts. In the first part, we capture wireless traces and

write them to the local disk on the node (during an experiment). In the second part, we transfer the data from the disk to the central NFS server for further processing. For the second part, we simply copy the logs to the NFS mounted filesystem image on the node, which is then eventually synced with the central server. Note that simultaneously copying these logs across all nodes generates a lot of traffic. Therefore, we perform the second part in sequence for each node. Also, if we are interested in processing the logs in realtime, we do so at the nodes themselves and then send summaries of these logs to the central server. This reduces the overhead of sending raw traces over the network, which could cause congestion in the backbone.

5.4.3 Network Deployment

An important task during any enterprise WLAN deployment is deciding the placement of nodes in the building. Traditionally, site surveys have guided such decisions. Most testbeds today also use a similar deployment strategy [46]. In contrast, our testbed is deployed in a uniform *grid-like* fashion. This method of deployment is motivated by two trends. First, site survey-based techniques have been largely unsuccessful since network usage patterns change over time (e.g. due to corporate restructuring). Second, access points (or nodes) have become increasingly cheap, thereby allowing dense deployments. This creates a network that has both greater coverage and greater capacity. However, increased density also brings about problems of interference between nodes, which needs to be managed. Our testbed opens up the space for work on interference mitigation and dynamic network re-configuration. We plan to study how network optimization can handle dynamic changes that occur in a dense enterprise WLAN. Hence we create a testbed of 38 nodes, covering an area of 120m x 65m. In the future, we plan to install additional nodes and study the impact of increased network density on overall network performance.

5.5 Experiments

We now present some experimental results collected on our testbed. The results showcase the performance of our testbed with an emphasis on its network throughput and centralized control capabilities. We also comment on our experiences with using the testbed.

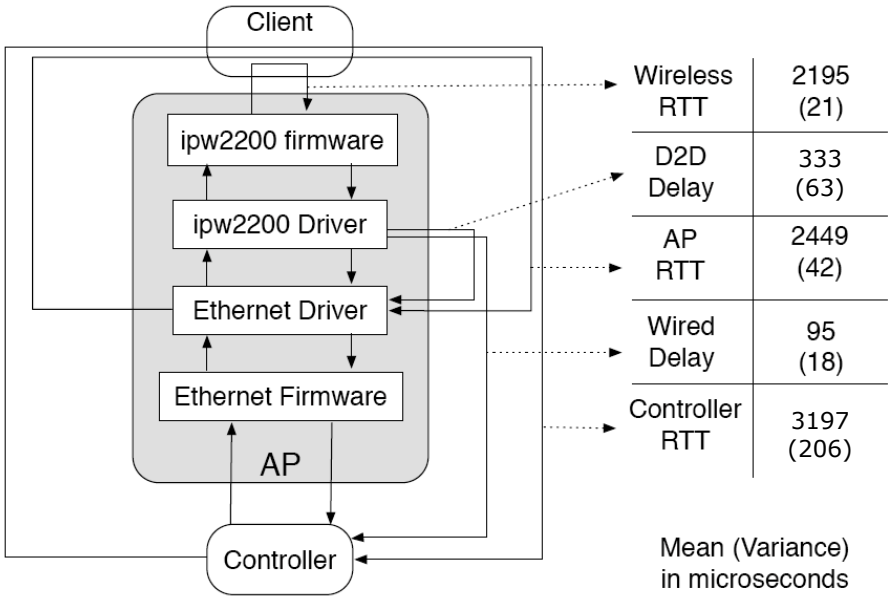

Figure 5.3: Delays in each component of our testbed.

5.5.1 Performance

Path Latency

As a first step, we were interested in characterizing the latency (or delay) that exists in different components of our testbed. We did this by instrumenting the system with timestamps at different points along the controller-to-AP path, as shown in Figure 5.3. These delays were measured over a period of $24h$ (the numbers in the brackets show the delay variance for each component.). We observe that aside from the delay in transmitting a frame on the air (which is fixed), the remaining delay accounts for $\approx 30\%$ of the total controller-to-AP RTT. Furthermore, the total observed delay jitter is close to $\approx 200us$. While these delay numbers are reasonably tight, in Chapter 6 we show how to reduce them even further by optimizing certain parts of the data path.

We also studied how well we are able to synchronize APs in our testbed. As discussed in Section 5.2, this is important to ensure that certain control actions (described in greater detail in Chapter 6) occur simultaneously at the APs. The details of the experiments conducted and their corresponding results are presented in Section 6.5.1.

AP ID	1	2	3	4	5	6
Mbps	291	150	95.4	95.3	286	95.2

Table 5.1: Bandwidth measurements from the controller to 6 APs

Network Throughput

Our aim in studying network throughput is to determine the capacity of our testbed, or how many users it is capable of handling. This allows us to gauge its scalability properties, in the context of data plane centralization. This turns out to be challenging for two reasons. First, finding a large number of users (e.g. $\sim 200 - 400$) for experimentation is hard. Second, real-world workloads are typically bursty and vary considerably among users [114]. Thus, we *estimate* capacity by performing bandwidth measurements between the controller and each AP in our testbed. Our goal is to determine the peak throughput achievable on any controller-to-AP link. This gives us a rough estimate on the maximum bandwidth (or capacity) that the controller can provide to the network. We conducted measurements with all APs, performing 5 iterations for each. Note that because we use a VLAN, our measurements are potentially affected by cross-traffic in the backbone. To minimize this effect, we performed measurements at night and over the weekend. Table 5.1 presents results for 6 APs in our testbed. Note that the bandwidth to some APs is below 100 Mbps. This is due to the presence of legacy 100 Mbps switches that are still in use by our department. The department is currently transitioning from 100 Mbps Ethernet to Gigabit Ethernet. Once complete, we expect the performance to significantly improve for lower bandwidth links as well. Nevertheless, based on our measurements, we estimate the peak capacity of our testbed to be ~ 300 Mbps, which is close to the practical limit of Gigabit Ethernet [110].

To put these bandwidth measurements into perspective, consider the following scenario. Suppose each user is running a bandwidth-intensive application such as (MPEG-1) video at a rate of 1.5 Mbps [6]. Assuming that the users are uniformly distributed in the building such that no AP is overloaded, the maximum number of users the testbed can support is ~ 200 (given a peak capacity of 300 Mbps). This is the worst case because user traffic is typically bursty and it is unlikely that all users will be streaming video at the exact same time instant. Moreover, with recent advances in video compression techniques such as MPEG-4 [4], bandwidth requirements for these applications have gone down as well. Therefore, realistically speaking, our testbed should be capable of handling a significantly larger number of users than currently estimated.

5.5.2 Experiences Using the Testbed

We now briefly report on some experiences using the testbed.

The testbed was relatively easy to manage. In most cases, failures resulting from power outages and node crashes were automatically handled by the hardware watchdog. Nodes took $\sim$ 3 minutes to recover from a failure[5]. However, there were rare occasions where some nodes failed to restart. These nodes were inaccessible (through SSH) but were connected to the NFS server, thus preventing reboots via the watchdog. To handle these cases, we wrote a simple program (running on the controller) that temporarily blocked traffic to that node. This forced the hardware watchdog to initiate a reboot. Once installed, this program resolved most of the remaining unhandled failures.

Developing code on the testbed was also relatively easy[6]. Because we are using NFS, we implemented most of our algorithms on the controller. Testing modifications to the driver/firmware was a simple matter of compiling the code (on the controller), copying it to the NFS mounted image, and loading it on the node. For modifications to the node's kernel, a node restart was required.

5.6 Summary

Designing a centralized WLAN testbed on commodity hardware is a challenging problem. It requires carefully thinking through its requirements and identifying the correct platform that meets these requirements. In this chapter, we considered a number of possible platforms for building a centralized WLAN and outlined their limitations. In doing so, we presented the design of our own centralized WLAN testbed that has a number of unique features. We presented some experimental results to showcase how well our testbed is able to meet the requirements for centralized control. Finally, we also discussed our experiences with using the testbed and found that it is not only easy to manage, but also facilitates rapid development of protocols and algorithms that can be studied for centralized control.

[5]This was configured for the hardware watchdog at each node

[6]Most of our implementation involved modifications to the driver/firmware of the Intel 2915ABG radios

Chapter 6

Micro-Probing: Practically Measuring Conflict Graphs for Enterprise WLANs[1]

In this chapter, we describe our efforts towards practically realizing the conflict graph measurement framework presented in Chapter 4. The resulting implementation (dubbed 'Micro-Probing') represents the core contribution of this dissertation and lays the groundwork for novel and exciting research in the area of enterprise WLAN optimization (as we discuss in Chapter 8). Micro-Probing is implemented and tested on the wireless testbed described in Chapter 5. While we focus on the measurement framework in this chapter, we assume the conflict graph model proposed in Chapter 4 of this dissertation.

There is a significant gap between theory and practice when designing protocols and algorithms for wireless systems [79]. Practical constraints need to be accounted for in the design of wireless protocols, thereby necessitating real-world implementation. While implementing SMARTA's interference measurement framework, we faced numerous engineering challenges; such as the need for micro-second level synchronization between APs and accurate silencing of the wireless medium to properly conduct interference measurements. In addition, our implementation efforts also revealed difficulties in correctly implementing certain features of SMARTA, such as using energy-on-the-air measurements to detect carrier-sensing interference. This required re-designing certain interference tests to overcome these practical challenges. These and other implementation aspects are described in greater detail in this chapter.

[1]This Chapter revises a previous publication: [30] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of RF interference. In Proceedings of ACM CoNEXT, 2008 (refer to Appendix A)

The Micro-Probing interference measurement framework presented in this chapter is subsequently used in two network optimization systems for enterprise WLANs. The first system (called 'Overcast') is an optimization framework that supports mobility for VoIP clients in single channel WLANs. The details of Overcast are described in Chapter 7. The second system (called 'CENTAUR') is an optimization framework that supports centralized scheduling of downlink data traffic in enterprise WLANs. We refer the reader to [109] for details on the CENTAUR system.

The rest of this chapter is organized as follows. Section 6.1 describes bandwidth tests, the current state-of-the-art approach to conflict graph construction. Section 6.2 covers the theory of micro-probing and Section 6.3 briefly compares micro-probing with prior techniques to conflict graph construction. Section 6.4 discusses the design of our prototype implementation. Sections 6.5 and 6.6 benchmark micro-probing's core components and evaluate its performance against bandwidth tests. Finally, a discussion and some limitations are covered in Section 6.7.

6.1 State-of-the-Art

We now describe the details of the current state-of-the-art approach used for conflict graph construction [98], which we briefly touched on in Chapter 4. In this technique (termed 'Bandwidth tests'), a sender broadcasts packets at the highest possible rate and all receivers measure the packet delivery ratio, in the presence and absence of simultaneous packet transmissions from a potential interferer. If the interferer's presence causes a drop in throughput at any of the receivers, we infer that a conflict exists. For instance, if we observe performance degradation when links *l1* and *l2* are simultaneously active, we say that those two links interfere. Note that interference between links is not 'binary', but instead a ratio between 0 and 1, as we describe later. In Section 6.6, we discuss the specific metric used by bandwidth tests to estimate the degree of interference between pairwise links.

Bandwidth tests are prescribed for measuring the conflict graph for a particular fixed configuration of the nodes. Typically, all nodes are assumed to transmit at maximum power and use the base rate (i.e., 6 Mbps) for their transmissions. Because measurements are done in a pairwise fashion, the measurement complexity is $O(n^2)$, where is n is the number of nodes. Furthermore, to account for time-varying channel conditions, measurements are done over sufficiently long time periods to factor in the typical channel noise. Measurement times are typically between 20 and 30 seconds per link pair.

Because of their systematic approach, bandwidth tests are able to accurately estimate

interference between links. However, there are some limitations of this approach, in the context of enterprise WLANs. First, this approach suffers from significant measurement overhead and can take hours to run even for a modest sized network of 20 APs. Second, it requires that the network be idle for the duration of the measurements to preserve measurement accuracy. This may be acceptable for measuring inter-AP conflicts (e.g. overnight), but does not work for clients that come and go in the network. Finally, bandwidth tests also require clients to report measurements to the APs. These drawbacks make them infeasible for online estimation of RF interference. This motivates the design of the Micro-Probing approach, which we describe in subsequent sections.

6.2 Theory of Micro-Probing

In this chapter, we focus only on downlink conflicts, i.e. those due to traffic sent from APs to clients. Because most traffic in today's enterprise WLANs is downlink in nature [1], downlink conflicts are the dominant form of interference in enterprise WLANs. Two types of downlink conflict can be captured in a conflict graph: i) conflict due to carrier sensing between contending APs, and ii) conflict due to AP-client collision (as discussed in Section 2.2.2). Micro-probing implements two different tests to differentiate between the two scenarios. While the collision-induced test is similar to the OAP test described for SMARTA, carrier-sensing based conflicts are detected using a different approach, described next.

Testing for Carrier-Sensing interference: In order to test for Carrier Sensing (CS) induced interference, we need to have both wireless transmitters transmit at the *same* time. Micro-probing instructs one AP, AP_i, to initiate a series of broadcast transmissions at well defined time instants $t_1, t_2, ..., t_m$. AP_j, is then instructed to also transmit at the same time instants plus a slight offset (≈ 50 microseconds) to ensure AP_i acquires the channel first. If AP_j, is delayed by approximately one frame time before transmitting, we infer that it is in CS range of AP_i. In our implementation, we use an estimate of MAC service time (MST) to detect such an event (we discuss MST in detail in Section 6.5). Given that this test needs to be performed between each pair of APs, the total number of tests required is $O(N^2)$, where N is the number of APs in the network.

The carrier-sensing interference test above attempts to detect exposed terminals. While carrier-sensing allows us to determine whether two APs are 'exposed' to each other, it does not tell us whether disabling carrier-sensing (and operating the links in parallel) could lead to a collision at the receivers. We do not test for this case because we assume APs are 802.11 standards compliant and therefore do not disable carrier-sensing to alle-

viate exposed terminal interference. However, the design of micro-probing is not averse to such functionality and can incorporate it, should that become necessary.

Testing for collision induced interference: To test for collisions at the receiver we proceed as follows. We initiate a transmission between AP_i and its client, say C_1, at time t_0. AP_j is then instructed to send a broadcast frame at the same time. If AP_i does not receive an ACK within SIFS, we can infer a collision at the receiver[2]. As in SMARTA, this test is repeated m times to account for temporal channel impairments from affecting our tests.

Collision induced interference can be observed only in the absence of carrier sensing induced interference. If the AP cannot simultaneously transmit with a neighbouring AP, then testing for collisions with that AP is unnecessary. Given that there are a total of C clients (and therefore links) in the network, and there are N-1 APs that must be tested for interference against each link, a total of O(CN) tests need to be performed. However, because some APs are likely exposed to each other, the number of actual tests is expected to be much lower.

Silencing: Note that, as discussed in Chapter 4, the interference tests described above would give incorrect results if they are conducted while other traffic is being carried in the network. To ensure that the wireless medium is silent, we need to force all APs and clients in the neighbourhood to be silent. We do this by having the APs conducting the test broadcast a CTS-to-self or Ack packet (with an appropriate NAV duration) *before* initiating a test. We study the efficacy of this method of silencing in Section 6.5. To ensure that the impact of silencing is minimized, we choose the smallest possible NAV that is sufficient to accommodate an active test. The duration for an active test is typically between 1 and 2 ms, and depends on the packet size and data rate. This overhead is sufficiently small to accommodate even delay-sensitive applications such as voice, where the typical inter-packet arrival time is on the order of 20 - 30 ms.

6.3 Comparing Micro-Probing with Prior Techniques

In this section, we briefly compare micro-probing with prior approaches to conflict graph construction. To do this, we first briefly cover existing approaches to CG construction and then qualitatively compare these techniques with micro-probing where our goal is to measure interference in an enterprise WLAN.

[2]We, as in prior work [98], assume good quality links

6.3.1 Existing Approaches to CG Construction

Prior work on conflict graph construction can be categorized into passive and active techniques. We discuss each of them in turn.

Passive

Passive approaches collect traces using monitors deployed throughout the building. Monitors are dedicated hardware devices that sniff wireless traffic and collect traces in order to perform management tasks. The traces are processed at a centralized aggregation point and are subsequently fed into interference inferencing algorithms. Jigsaw [51] and WiT [85] are examples of systems that adopt passive techniques. Passive techniques are also popular among enterprise vendors such as Aruba [1], primarily because they don't introduce any traffic into the network for measuring interference. Nevertheless, their predictive power is heavily dependent upon on how densely the monitors are deployed in the building because with increasing density the probability that a monitor is close to any given link increases. Furthermore, passive techniques *predict* interference from collected traces, hence they are likely to be less accurate than techniques that actively measure interference.

Active

Active approaches inject control traffic into the network to estimate interference between wireless links. There are two categories of such active approaches: pure measurement techniques and measurement-modeling techniques. Pure measurement techniques include bandwidth tests and estimate interference in the manner described in Section 6.1. In what follows, we discuss the second approach to active interference measurements.

Reis et al [105] propose an optimization for bandwidth tests where they combine measurements with the SINR model to reduce the overall number of measurements. Their work was recently extended for the case of multiple interferers, carrying different amounts of traffic load [76, 101]. An element common to all such modeling-based proposals is the use of RSSI to predict interference. Unfortunately, RSSI is only available if the 802.11 preamble for a packet is received correctly, i.e., the interferer is likely in communication range of the receiver. Lee et al [81] address this limitation by proposing the use of two radios: a high-power radio to reach interferers outside of communication range and a low-power radio for normal communication. Nevertheless, like bandwidth tests, these measurement schemes also require receiver statistics, which makes

	Passive	Active	Micro-Probing
Low Control Overhead	✓	✗	✓
Accuracy	✗	✓	✓
No Network Downtime	✓	✗	✓
Low Feedback Delay	✗	✗	✓
No client modifications	✓	✗	✓
Captures Weak Interferers	✗	✓	✓

Table 6.1: Comparing active, passive, and micro-probing techniques

them harder to deploy in enterprise WLANs. Moreover, these techniques are likely to be less accurate than pure measurement schemes because they perform fewer measurements and infer interference based on models that make simplifying assumptions about the RF environment.

There is also work that combines active and passive techniques to measure interference, called CMAPs [118]. CMAPs opportunistically discovers exposed terminals by first disabling carrier-sensing and observing link performance. If the performance degrades, carrier-sensing is enabled on the link. However, the limitations of this approach are (i) It requires the interferers to be in communication range, and (ii) It requires client modifications to report packet delivery statistics. Aside from the interference mapping schemes discussed above, there is also work on studying properties of RF interference in 802.11 networks. Niculescu et al. [97] highlight properties that can reduce the overall complexity of measuring interference. These properties include linearity of interference with respect to the source's sending rate, and independence of multiple interferers. Das et al [54] study remote interferers that do not individually interfere, but when combined can cause significant interference. However, they point out that the occurrence of this phenomenon is rare. These studies add significantly to our understanding of how RF interference impacts link quality and performance in IEEE 802.11 networks.

6.3.2 Comparison Summary

We broadly classified prior work as either passive or active. The main underlying theme is that while passive techniques incur little to no cost in terms of measurement overhead, they are less accurate than active techniques. Conversely, active techniques are more accurate than passive techniques but suffer from high overhead. This dichotomy motivates

the development of a new approach that captures the best of both worlds. Micro-probing is an attempt to achieve this objective.

In order to put active, passive, and micro-probing techniques into perspective with one another, we first outline the key features that are necessary for building an online interference estimation system. These features are listed in Table 6.1 and discussed in greater detail next.

Control overhead indicates whether or not a technique requires the use of measurement packets to estimate interference. Active techniques by definition require such packets while passive techniques do not. On the other hand, active techniques are highly accurate because they directly measure interference between links whereas passive techniques only predict the same. However, some active techniques require excessive downtime for measuring interference while passive techniques do not. Both active and passive schemes suffer from high feedback delay (i.e. slow response times) because active techniques have a lengthy measurement cycle whereas passive techniques have a lengthy processing cycle (trace merging/synchronization, time series analysis, etc). Active techniques also require client statistics and therefore are not legacy-compatible. Finally, weak interferers (i.e. those outside of communication range of the target link) are hard to capture using passive techniques while some active techniques (e.g. bandwidth tests) can capture such cases. In summary, both active and passive techniques lack at least one feature necessary for online estimation of RF interference. In contrast, micro-probing incorporates all these features and is therefore our technique of choice for online estimation of RF interference.

6.4 Design and Implementation

In this section, we outline the design of our micro-probing system. A high-level overview of the architecture is shown in Figure 6.1. It consists of a central controller that sends probing requests to APs and APs that carry out experiments and respond with results. We describe the details of our implementation next.

6.4.1 Controller Implementation

For the implementation, we used the testbed described in Chapter 5. Therefore, the central controller was implemented on a standard Linux desktop PC, connected to the APs via a wired backplane comprising both 100 Mbps and Gigabit Ethernet wiring. As

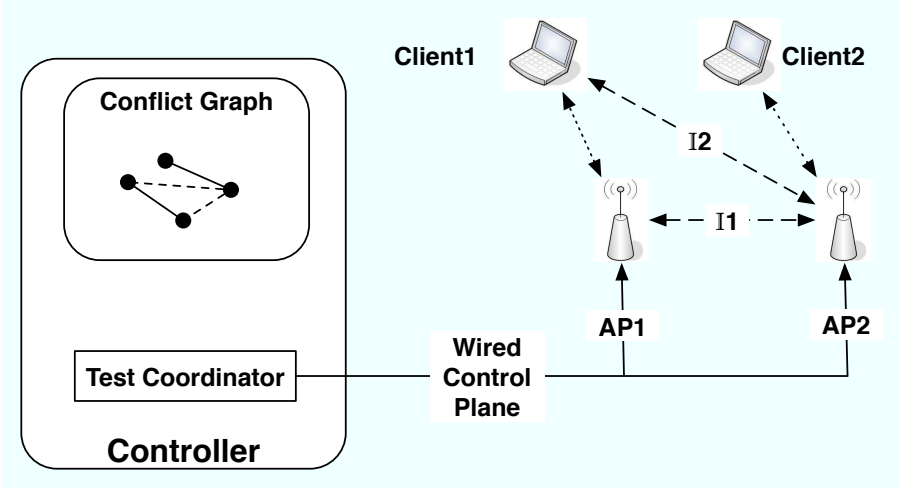

Figure 6.1: High-level overview of Micro-Probing Architecture

explained earlier, our testbed operates over our department's wired backplane (using a VLAN) and we do not use a dedicated backbone for our network. Therefore, our active tests can suffer from cross-traffic in the backbone. We evaluate the impact of such traffic in the next section. The central controller software was implemented in user space for extensibility and flexibility.

6.4.2 AP Implementation

As explained in Chapter 5, the APs consist of a 1.2 GHz VIA Processor [15] with 1 GB of DRAM. We installed the 2.6.16.19 Linux kernel on the APs and implemented Micro-probing on the Intel 2915ABG card.

The software architecture of the AP is shown in Figure 5.3. It consists of three parts: (i) the ethernet driver that receives commands from the controller, (ii) the wireless driver that executes these commands inside the kernel, and (iii) the firmware/microcode that interfaces with the wireless driver and sends out the micro-probes. We discuss our modifications to each part next.

Kernel/Driver Modifications: To minimize processing delays while forwarding probe requests from the wired Ethernet interface to the wireless interface of the AP, we implemented a direct driver-to-driver (D2D) communication path, where the Ethernet driver directly invokes functions implemented by the wireless driver. This bypasses time-consuming packet processing tasks and other unpredictable in-kernel events that can affect the accuracy with which micro-probes are synchronized over the air. When a

micro-probing request is received on an AP's wired interface, its embedded command is parsed, and the appropriate wireless operation is immediately invoked.

On the reverse path (AP $\rightarrow$ Controller), the AP uses an in-kernel raw socket to send responses (to micro-probing tests) back to the controller (using the controller's IP address). Note that raw sockets can only be used for sending packets, not receiving them. The controller receives responses by capturing packets on the appropriate Ethernet interface.

Firmware Modifications: The Intel 2915ABG NIC software comprises of two parts: 1) *Firmware* that interfaces with the wireless driver, and 2) *Microcode* that interfaces with the firmware. The firmware implements an RTOS (Real Time Operating System), responsible for handling macro-timescale operations, such as AP beaconing. The microcode consists of a set of specialized hardware acceleration blocks that are used for micro-timescale operations, such as counting down backoff timers for IEEE 802.11's randomized backoff algorithm.

We modified the firmware and microcode running on the wireless NIC to support transmission of micro-probes from within the firmware. Constructing a probe packet in the driver would require a DMA-copy of the packet from kernel-space to firmware memory. This is unnecessary since the payload of the probes doesn't carry any useful information. Note that this implementation choice has no effect on the applicability of micro-probing but is simply a way of eliminating unnecessary processing overhead in the driver.

As discussed in Chapter 2, Binary Exponential Backoff (BEB) is a standard mechanism by which 802.11 compliant devices coordinate access to the medium. Such medium access techniques are unsuitable for micro-probing because they prevent the interference scenarios (outlined earlier) from occurring. We therefore disable randomized back-off when sending out probes. Note that we only disabled back-off for our micro-probes, not other packets. Therefore, all of our extensions in the driver, firmware, and microcode are 802.11 standards compliant.

Silencing: Silencing the network is a crucial requirement for micro-probing. It is challenging to achieve because the environment may be populated with both 802.11 as well as non-802.11 devices such as microwave ovens and cordless phones. In our system, we achieve silencing by instructing the driver/firmware to send CTS-to-self packets with a duration equivalent to the execution time of an active test. The silencing packet is transmitted immediately preceding the micro-probe transmission and this is performed before each and every test. We present results on the effectiveness of silencing in Section 7.3.

Synchronization: The controller communicates with the APs participating in a test using a single broadcast UDP packet sent over the wired LAN. This serves two purposes. First, it tells an AP what to do during a test. We use a single control packet to encode multiple actions, one for each AP[3]. Second, it allows us to synchronize APs to one another through the use of wired MAC layer broadcasts to support *reference-based broadcast synchronization* (RBS) [57]. Reference broadcasts use the packet's time-of-arrival at the APs to *mutually* synchronize them. A key underlying assumption is that all APs receive the broadcast packet at the same time instant. In the next section, we evaluate the extent to which RBS-based synchronization can be achieved. Note that synchronization accuracy is dependent on the transmission duration of the probes. For a probe of size 1400 bytes, the transmission duration is approximately 1800 us, at 6 Mbps. Therefore, synchronization to within a few tens of microseconds is sufficient for micro-probes of this size.

We now briefly describe two alternative approaches that we considered before deciding to use RBS-based synchronization. The first approach is NTP-based synchronization [9]. Here, the controller is the master and the APs act as slaves. The master's job is to periodically synchronize the slaves to it's own clock. Unfortunately, NTP is known to provide accuracies in the range of $1 - 5$ ms, which is inadequate for our purposes.

The second approach is to synchronize APs with the help of TSF timestamps encoded in the Beacons of neighbouring APs, as is done in [51]. However, this approach is significantly more complex than RBS-based synchronization. The complexity arises in scenarios where the APs performing the test are not in communication range of each other and therefore can't decode one another's Beacons. In this scenario, a third AP's Beacons (that is in range of the other two) is required to support synchronization of the two APs. This is a significantly complex process, and as we show later, is unnecessary because we can achieve similar levels of accuracy using the simple and lightweight RBS-based approach to synchronization.

6.5 Performance of Micro-Probing

The effectiveness of micro-probing depends on: 1) our ability to tightly synchronize APs, 2) our ability to silence the network before an experiment, and 3) our ability to use MAC service time (MST) as a mechanism to detect carrier-sensing induced interference. In what follows, we evaluate the effectiveness of these techniques.

[3]Note that we only require a few bytes of information per AP. Given an Ethernet MTU of 1400 bytes, we can easily scale to a large number of APs

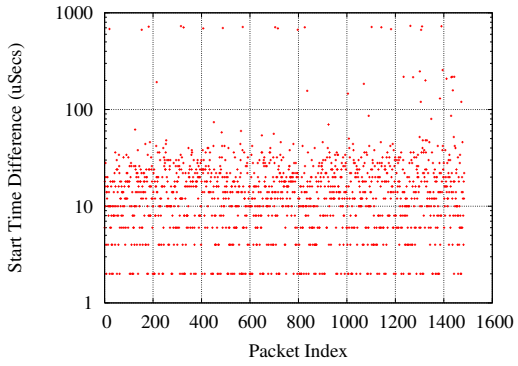

Figure 6.2: Synchronization error between APs

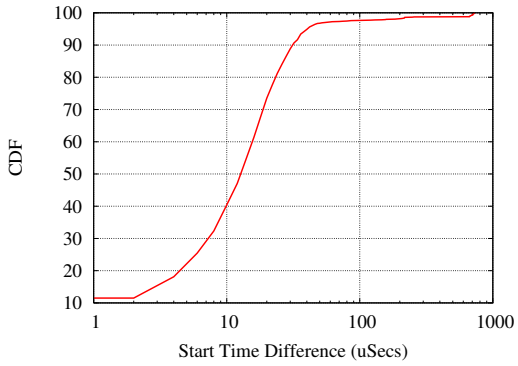

Figure 6.3: CDF of synchronization error

6.5.1 AP Synchronization

Our evaluation of AP synchronization is subdivided into: 1) Characterization of delays in our system, and 2) Analysis of the degree to which our micro-probes can be synchronized.

Delay Characterization

In Chapter 5, we studied the path latency of the wireless testbed that we deployed to test micro-probing and other centralized control algorithms. We found that the one way delay, excluding the wireless transmission delay, accounted for $\approx 20\%$ of the total end-to-end delay. In this section, we re-measure these delay values, but this time also implement the driver-to-driver (D2D) optimizations discussed in Section 6.4.2.

As expected, we find that apart from the D2D delay (see Figure 5.3), all other system components exhibit approximately the same delay. However, the D2D delay falls dramatically from $333us \pm 63us$ to $27us \pm 15us$, representing an almost 12 fold improvement. Furthermore, the total delay jitter falls to $\approx 100us$, which is remarkably tight[4]. This highlights the importance of optimizing the data path between the controller and the AP. Next, we test how tightly APs can be synchronized using the wired broadcasting approach described in Section 6.4.

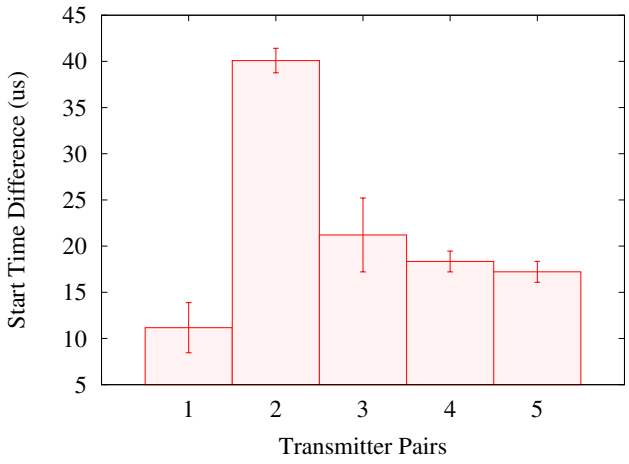

Figure 6.4: Mean synchronization error across 5 links

Probe Synchronization

We now investigate how tightly probe transmissions can actually be synchronized over-the-air using RBS-based synchronization. We select five random AP pairs from our network. For each AP pair, we send a stream of probes from the controller to both APs. On each AP, we use its secondary radio to capture packets. Due to power capture effects, all collisions at the secondary radio of the AP are resolved in favor of that AP's transmitting radio. To decipher whether or not the APs were simultaneously transmitting micro-probes, we analyze the traces that were collected by the secondary radios. We synchronize them to a common time base, in order to correlate events between traces. For synchronization, we use reference beacons from a third AP that is in communication range of the two transmitting APs. Because beacons are transmitted at $100ms$ intervals, we are able to re-synchronize the traces every $100ms$, within which the effect of clock drift is almost negligible [51]. We then compute the difference in the start times of the micro-probes and plot them for all such packets.

Figure 6.2 shows the result of the experiment for one of the AP pairs (start time difference is shown on a log-scale). We observe that the start time difference is mostly on the order of tens of microseconds. The CDF of the plot in Figure 6.3 further indicates that most of the mass lies between $7-40us$. Figure 6.4 summarizes our results across all five AP pairs. Again, we observe that most points lie in the $10-25us$ range. Based on the synchronization requirements we outlined in Section 6.4.2, these results provide strong

[4]Note that the total delay jitter sees improvements in the forward and reverse directions of the path from the controller to the client.

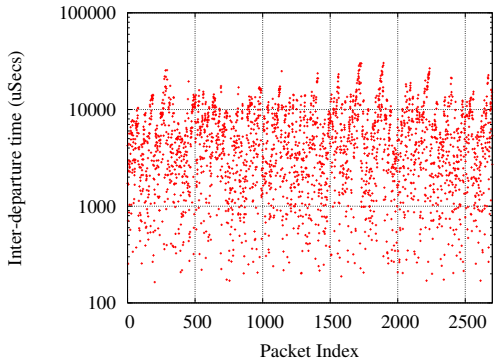

Figure 6.5: No Silencing

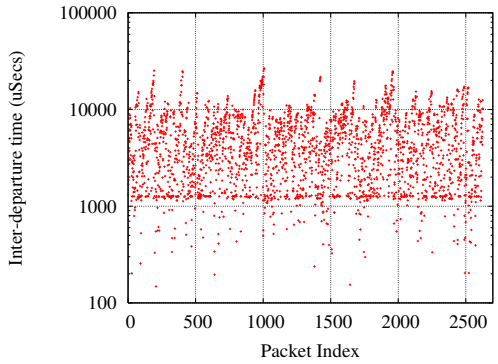

Figure 6.6: Silencing in Scenario 1

empirical evidence that RBS-based synchronization is highly effective in synchronizing APs at micro-second level granularity.

6.5.2 Silencing Ability

We now examine an AP's ability to silence the network for short periods to perform a micro-probing experiment. We outline two scenarios in which we test silencing:

- **Scenario 1: (Co-located Enterprise WLAN):** We study the effectiveness of silencing when both our testbed and our department's enterprise WLAN [1] are operating on the same channel (using IEEE 802.11b/g).
- **Scenario 2: (Standalone Enterprise WLAN)** We study the effectiveness of silencing on a channel not occupied by our department's WLAN (using IEEE 802.11a). In this scenario, we generate UDP streams from several APs on our own network and observe how effectively a co-located AP is able to silence such data traffic.

We evaluate these two scenarios to understand how well silencing works in the presence and absence of other co-located wireless networks. We test silencing using both CTS-to-self packets (used in 802.11g 'protection mode') and Ack packets with an appropriate NAV duration. Since both approaches yield similar results, we report only on the former.

Our experimental setup is as follows. One AP broadcasts CTS-to-self packets (with a NAV=1 ms [5]) at regular intervals and we use its secondary radio to observe the environment. The secondary radio records the time period between the end of the CTS-to-self transmission and the beginning of the next received packet. If this interval, referred to as

[5]We studied silencing for NAV values of up to 3 ms and obtained similar results

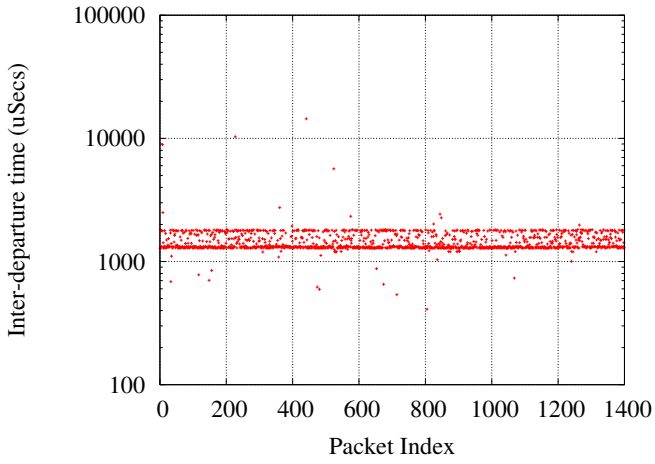

Figure 6.7: Silencing in Scenario 2

the *inter-departure time*, is greater than or equal to the NAV value listed in the CTS-to-self, then silencing was successful. Otherwise, it was not. To ensure that the CTS-to-self packets are correctly received at the neighbouring APs, we place the secondary radio of these APs in monitoring mode.

Figure 6.5 shows a plot where no silencing was performed (the NAV was set to $1us$), whereas Figure 6.6 presents results for silencing with a NAV of $1ms$ (for scenario 1). Comparing these two plots, we can identify a significant clustering of data points around the $1000us$ mark on the second plot. This indicates that silencing is able to successfully silence some nodes. Upon further investigation, we find that CTS-to-self silencing managed to reduce the number of packets within the $0 - 1000us$ range by only about 66%, compared to the case when no silencing was performed. We provide two explanations for this observation. First, we believe that the APs that are part of our department's wireless network do not comply with the IEEE 802.11 standard and ignore silencing packets. Second, because of the unpredictability of RF signal propagation, there may still be neighbouring APs that do not correctly receive CTS-to-self silencing packets. This motivates a coordinated approach to silencing where neighbouring APs also send out silencing packets. We discuss this approach in greater detail in Chapter 8.

Figure 6.7 presents results for scenario 2. In this case, we observe that the silencing period is almost always obeyed, with 99.92% of the packets lying outside the silencing period[6]. From this result, we argue that silencing is highly effective in cases where devices properly implement the IEEE 802.11 standard[7].

[6]Note that for this scenario, we generated traffic at rates high enough to saturate the medium

[7]We verified compliance for the wireless device vendor we used in our testbed

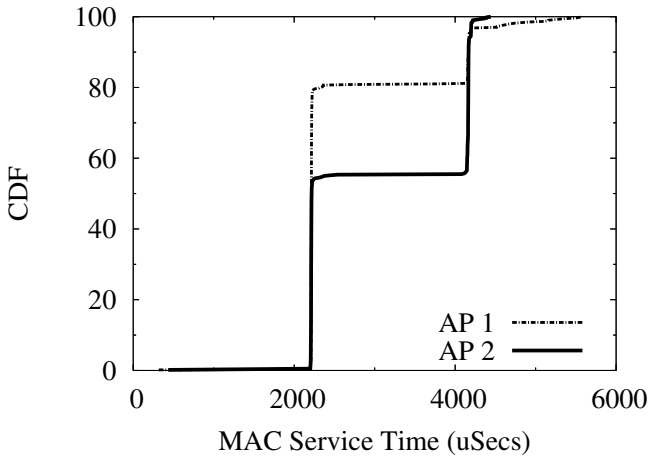

Figure 6.8: CDF of MST without staggering

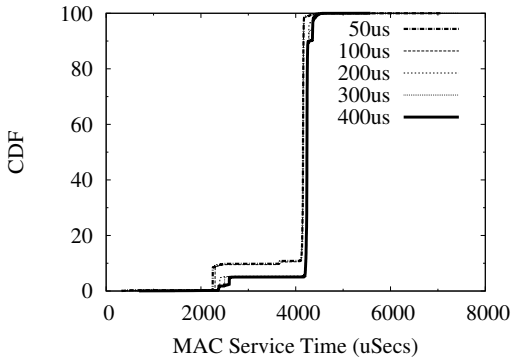

Figure 6.9: CDF of AP1 with staggering

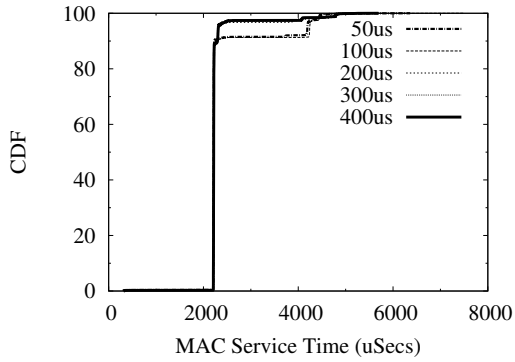

Figure 6.10: CDF of AP2 with staggering

In summary, the results of our investigation lead us to the following conclusion: In general, silencing appears to be an effective tool for generating an interference free environment. However, its effectiveness depends on whether co-located 802.11 devices correctly implement the standard and are actually able to receive silencing packets transmitted by the intended APs.

6.5.3 MAC Service Time

In Section 6.2, we proposed the use of MAC service time to detect carrier-sensing interference. We define MAC service time (MST) as the time taken by the wireless firmware in processing a packet transmission request. If during this time, the NIC carrier-senses another transmission, it backs off and thus takes a longer time to process the transmission request. Therefore, an increase in MST is indicative of carrier-sensing interference,

and micro-probing uses this method to detect such cases. In what follows, we assess the accuracy of using MST in detecting carrier-sensing interference.

Our experimental setup consists of a pair of APs whom we instruct to simultaneously transmit probes, while we record the MST values in the firmware. Figure 6.8 plots the CDF of the MST values we collected at the two APs. Observe that the MST is clustered at two points ($2000us$ and $4000us$). The fact that the second value is twice the first, indicates that the APs are backing off to each other's transmissions (given that the packet size and transmission rate are the same for both probes). However, note that neither of the APs always wins access to the channel before the other. Consequently, the measured MST values fluctuate considerably across runs, making interpretation of aggregate results difficult. To address this issue, we use *packet staggering*.

Packet staggering slightly delays transmission of one AP's probe so that the other AP's probe always wins access to the channel, if they are mutually exposed. This causes the first AP's MST to almost always be higher than the second one. This allows us to accurately determine that the first AP carrier-senses the second, as seen in Figures 6.9 and 6.10. By staggering for 50 us, we see an almost 20% improvement in detection accuracy. A larger staggering interval improves accuracy further because it envelopes probes that are affected by random in-kernel delays. Hence, MST combined with packet staggering is able to detect the presence of carrier-sensing interference between pairs of APs with a high degree of accuracy (i.e., 90% and above).

6.5.4 Summary

We have verified that the three important requirements of micro-probing can be met in practice: 1) AP synchronization using wired MAC-layer broadcasts that achieves synchronization accuracies on the order of tens of microseconds, 2) Silencing using 802.11's virtual carrier-sensing mechanism that works well especially in the presence of 802.11 compliant devices, and 3) MAC service time to detect carrier-sensing interference that in combination with packet staggering achieves accuracies of 90% and above.

6.6 Evaluation of Micro-Probing

We now proceed to evaluate the accuracy and overhead of micro-probing with respect to bandwidth tests. We first outline our evaluation methodology and then present our results.

6.6.1 Evaluation Methodology

Testbed Setup

We compare micro-probing with bandwidth tests on the 38-node wireless testbed described in Chapter 5. We use a data rate of 6 Mbps for all our experiments. Furthermore, we use 1400 byte packets because we want to study the effect of interference on real-world data traffic, which typically uses packet sizes equal to the Ethernet MTU. Our experiments use IEEE 802.11a, which is minimally used by other networks in our building. For bandwidth tests, we generate traffic at rates high enough to saturate the medium. At the receiver, we measure the packet delivery ratio for each link.

For micro-probing, traffic is generated by the controller and probe requests are broadcast to APs at 10 ms intervals. The value of the control parameter m (the number of experiments to perform per link) is fixed at 10. We later show how we empirically derived this value for our testbed.

Evaluation Metrics

We compare bandwidth tests and micro-probing using the Broadcast Interference Ratio (BIR) metric proposed in [98]. The BIR for bandwidth tests is computed as follows. We first measure R_{AB}, the number of packets received by node B on link A $\rightarrow$ B when all competing nodes are silent. We then measure R^C_{AB}, the number of packets received by B on the same link in the presence of a competing transmitter C. Because antennas are omnidirectional, it does not matter whom C is transmitting to–in other words, all links with C as the transmitter are potentially in conflict with link A $\rightarrow$ B. Then, BIR is computed as:

$$BIR = R^C_{AB}/R_{AB} \tag{6.1}$$

Note that a BIR of 0 means that link A$\rightarrow$B cannot deliver packets when C is active. This indicates that C and A are hidden terminals with respect to B. A BIR of 0.5 indicates that A and C share the air, when A is communicating with B, which means that A and C are exposed terminals. Finally, a BIR of 1 indicates that C does not interfere with link A$\rightarrow$B[8]

[8]We note that this metric is a slight modification to the one originally proposed in [98], which combines the interference effects between link pairs into a single metric.

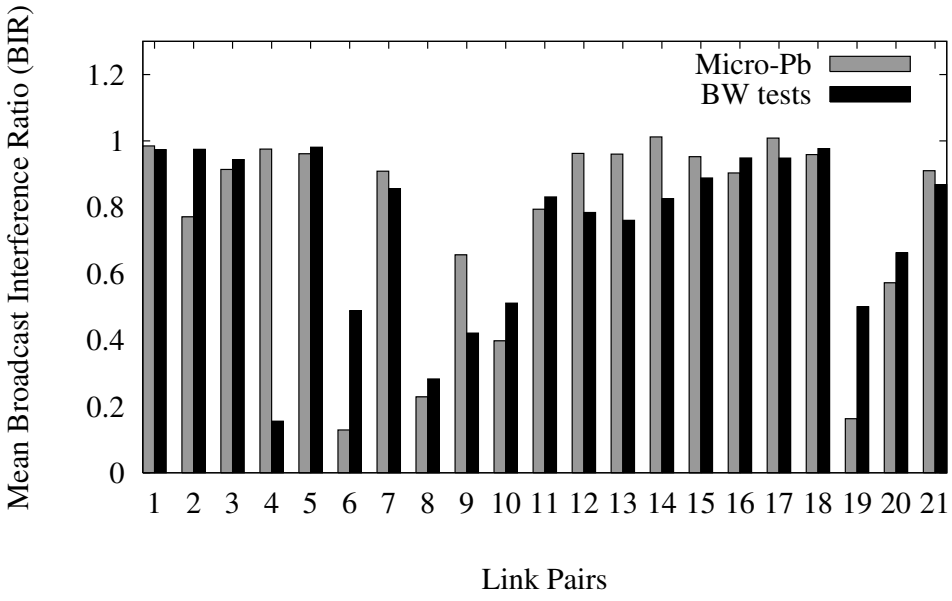

Figure 6.11: Mean BIR using micro-probing and bandwidth tests

For micro-probing, the BIR value is computed in the same way as shown in Equation 6.1. However, the numerator for micro-probing is different from bandwidth tests. The value in the denominator is the same because this is the link delivery ratio in the absence of interference. We now focus on computing the delivery ratio in the presence of interference.

Carrier-sensing interference: To estimate the impact of interference between two carrier-sensing senders, we adopt the following approach. We first send out probes synchronously from both APs. If m is the total number of probes sent out, then the number of timeslots for transmission in an interference-limited scenario would be $n+2*(m-n)$, where n is the number of "timely" successful transmissions. Notice that each transmission that was delayed will take 2 time slots and thus we have to multiply $m-n$ by 2. Therefore, the drop in delivery ratio representing the impact of interference between the two links is defined as:

$$DR_{interference} = m/(n + 2 * (m - n)) \tag{6.2}$$

Note that R^C_{AB} in Equation 6.1 and $DR_{interference}$ in Equation 6.2 both amount to number of packets transmitted per unit time and thus are comparable.

Collision-induced interference: In this case, the drop in delivery ratio due to interference is simply the number of successful packet deliveries (n) over the total number of tests m.

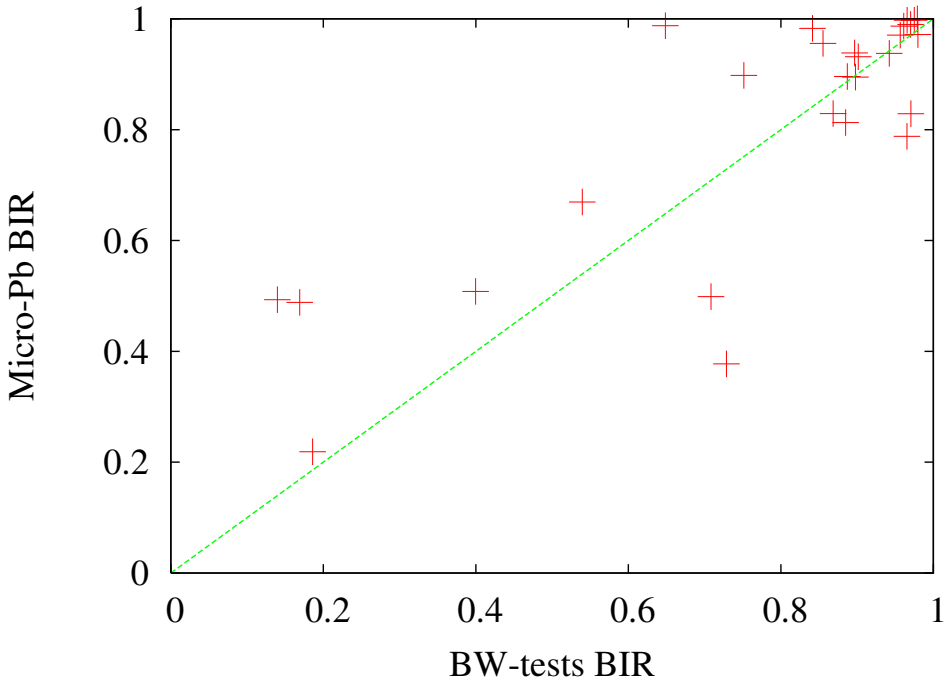

Figure 6.12: Mean BIR computed using Micro-Probing vs. Mean BIR computed using Bandwidth Tests

$$DR_{interference} = n/m \tag{6.3}$$

The goal of micro-probing is to quickly determine interference between pairs of links, which can be done within a few milliseconds. This captures a snapshot of interference between pairs of links and not the long term effect of interference. This can be problematic for weak links where channel conditions vary considerably over time. For such links, micro-probing can be run multiple times to capture the long term affects of interference on the link.

We perform 20 iterations of each experiment and plot the mean and median BIR across different link pairs. Recall that for micro-probing we have no knowledge of receiver statistics and thus estimate BIR based on information available at the sender.

In prior work [97], it has been shown that bandwidth tests need only be run on nodes with good link quality, because poor links would rarely be selected during client association. Therefore, to do an apples-to-apples comparison, we compare micro-probing with bandwidth tests only on 'good' links. We use the ETX metric [53] for this purpose. ETX describes the quality of a bidirectional link, i.e., the expected number of transmissions

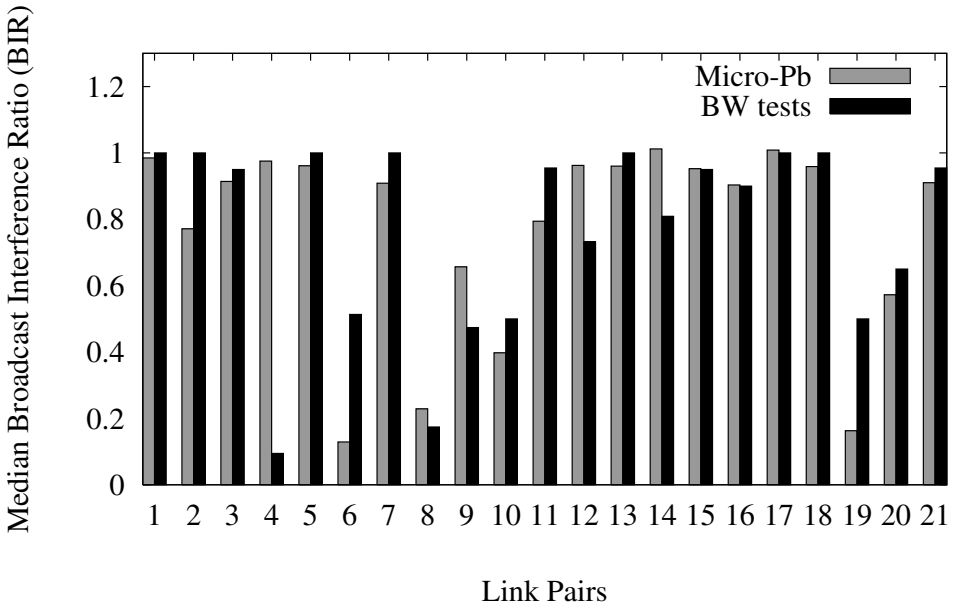

Figure 6.13: Median BIR using micro-probing and bandwidth tests

required to send a packet in either direction of the link. We select six AP-client links (i.e. 12 nodes) whose ETX metric [53] in both directions is smaller than three. For these links, we obtain a fairly diverse set of interference scenarios and choose a total of 30 such scenarios. In what follows, we refer to each interference scenario as a 'link pair'.

6.6.2 Accuracy

Mean BIR: Figure 6.11 shows the mean BIR of running bandwidth tests and micro-probing on 21 of the 30 link pairs. We observe that 14 out of the 21 link pairs have almost identical BIR for micro-probing and bandwidth tests. Four link pairs show a variation of less than 20%, while the last three show a fairly large variation in values. We also observe from this figure that most BIR values lie either close to 0.5 or 1. This indicates that many links are either isolated from one another or suffer carrier-sensing interference. Only 2 links appear to be suffering from hidden terminal affects, where the BIR is between $0.1 - 0.3$.

Figure 6.12 shows a scatter plot of the mean BIR computed using micro-probing and bandwidth tests (we remove one clear outlier point from the plot). We also show the $y = x$ line for reference. We see that many data points are clustered close to this line (correlation coefficient=0.8), with a few largely deviating from it. As observed earlier, we clearly see a clustering of points close to 1 and 0.5, indicating a larger presence of

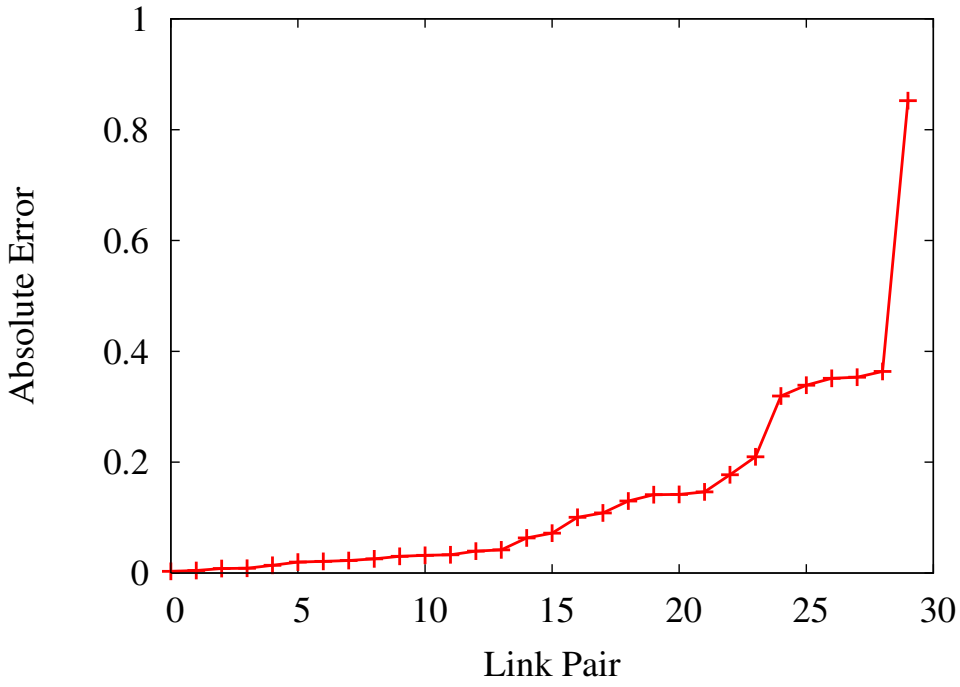

Figure 6.14: Absolute error in mean BIR

isolated and exposed terminals than hidden terminals.

Median BIR: We also compare the median BIR of micro-probing and bandwidth tests, to examine whether *individual* runs have a large deviation in value between the two schemes. Figure 6.13 shows a bar plot of the median BIR of the two schemes. We find that this plot agrees well with the mean BIR shown earlier, indicating that individual runs do in fact match fairly well with the mean value across those runs.

Degree of Error: We also quantify the degree of error in the values computed using micro-probing and bandwidth tests. We plot the absolute difference between the *mean* BIR values of micro-probing and bandwidth tests across the 30 link pairs (Figure 6.14). We observe that approximately 60% of the link pairs have an error of less than 0.1. Due to the unpredictable nature of RF signal propagation, we believe that this falls within the margin of error for computing BIR. Our results also show that 80% and 97% of the link pairs have absolute errors of less then 0.2 and 0.4 respectively. These results again confirm that the BIR computed using micro-probing closely correlates with that of bandwidth tests for most link pairs.

Impact of m: In all earlier tests, we fixed the value of m (i.e. the number of experiments to perform per-link) to 10. We now study the sensitivity of BIR to the value of m selected for micro-probing. To do so, we perform an experiment with $m = 50$. We then

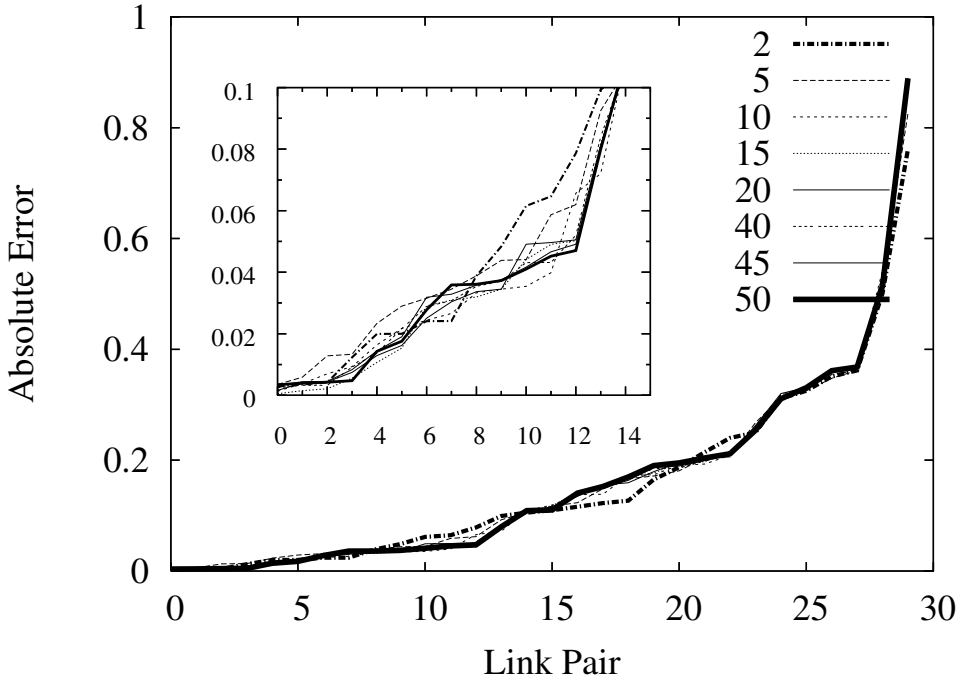

Figure 6.15: Mean Absolute Error

sub-sample the results for different values of *m* and compute the corresponding degree of error for the mean and median BIR as was done before (see Figures 6.15 and 6.16). Contrary to intuition we observe that the mean degree of error between micro-probing and bandwidth tests remains constant across different values of *m*. When we zoom into the left hand side of the graph, we observe that there is only a marginal increase in error, as we decrease the value of *m*.

We also plot the median degree of error (Figure 6.16) and observe a somewhat larger variation for different values of *m*, as is expected. However, even for the median, we observe that the increase in error due to small values of *m* is not too high and remains within ± 0.1 of the median for $m = 50$. This leads us to believe that even small values of m are sufficient to yield close to the same level of accuracy as large values. To investigate this further, in Figure 6.17, we show confidence intervals for the mean BIR across different values of m for 3 link pairs. The intuition behind selecting these 3 link pairs is to study variance across link pairs with high, moderate, and low BIR. The confidence intervals in Figure 6.17 show that the variance stabilizes as the value of m goes beyond 15. This result provides a basis for selecting a sufficiently small value of m that works well for most links.

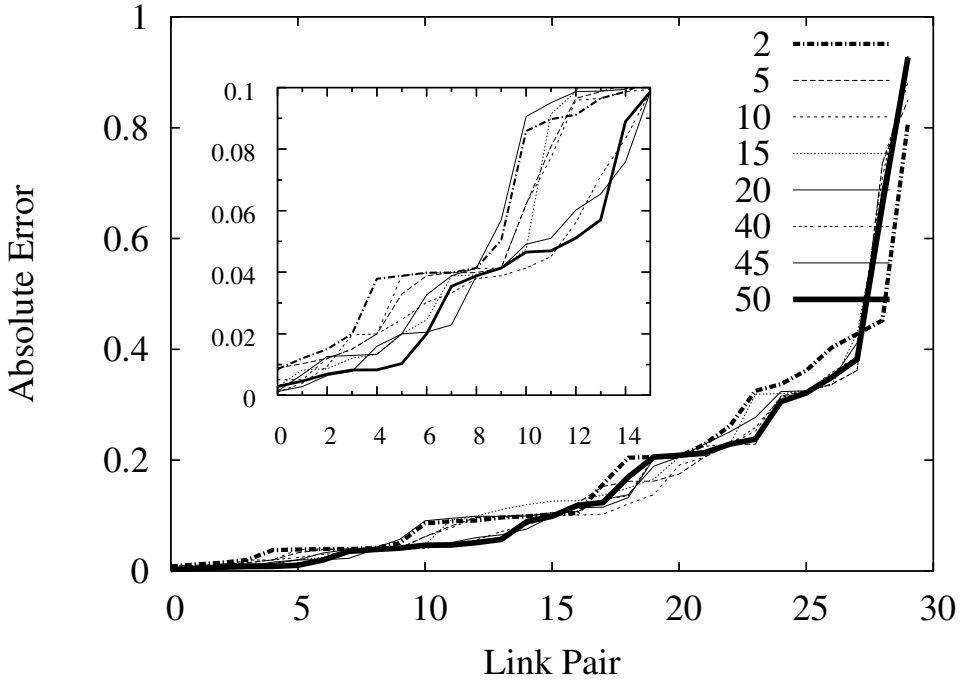

Figure 6.16: Median Absolute Error

6.6.3 Overhead

We now compare the running time of micro-probing and bandwidth tests on two topologies (see Table 6.2). On a small topology of 12 nodes (6 APs, 6 clients), we observe a speedup of 202. On a larger topology of 20 nodes (10 APs, 10 clients), we observe a speedup of 418. These results confirm that micro-probing substantially reduces execution time as compared to bandwidth tests.

We also present the mean running time of micro-probing on a per link basis. Figure 6.18 presents the *Round Trip Time* (in μs) of a micro-probing test (for a probe size of 800 bytes). RTT is defined as the time elapsed between the point the controller sends a micro-probing request to the AP, to the time it receives a response for that request (measured at the controller). We observe that the RTT for all tested APs lies between 1100 and 1300 μs. Considering a 1300 μs RTT per probe and a value of 15 for m (from Section 6.6.2), we estimate that micro-probing requires a running time of approximately $20ms$ per-link[9]. By comparison, bandwidth testing requires a measurement time of $20 - 30$ seconds per link [98], which is approximately $1000 - 1500$ times slower[10]. This again

[9]Note that this can be reduced by using smaller duration probes to support applications such as VoIP

[10]We only observe a three orders of magnitude speedup in conflict graph construction time because we

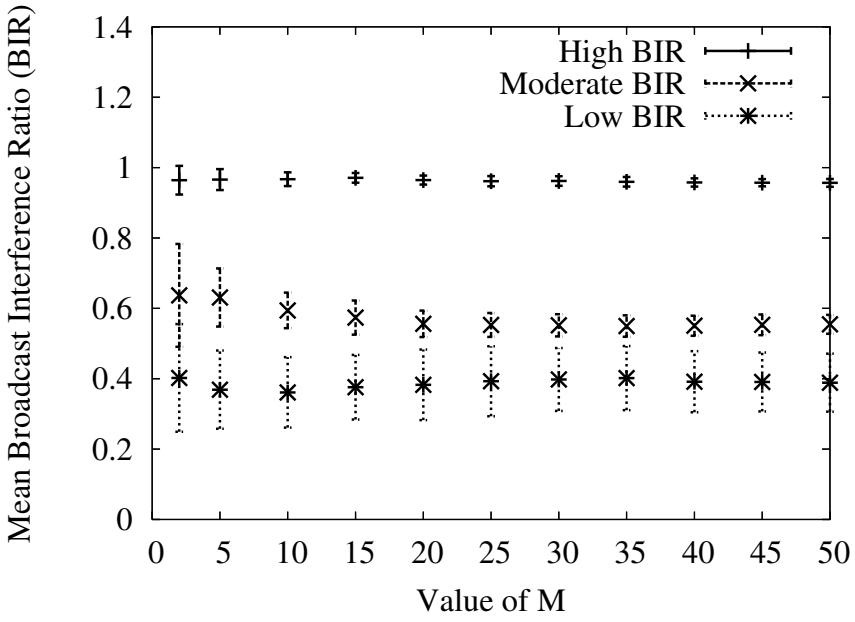

Figure 6.17: Confidence intervals for links with high, moderate, and low BIR ratios

Measurement Scheme	Running Time (12 nodes)	Running Time (20 nodes)
Bandwidth Tests	16.2mins	1hr 11mins
Micro-Probing	0.08 mins	0.17 mins
Speed-up	202	418

Table 6.2: Overhead of bandwidth tests and micro-probing on small and large topologies.

illustrates the gains from using micro-probing over bandwidth tests in terms of reduced measurement overhead.

6.7 Discussion

We now briefly comment on the scope of Micro-Probing. Micro-Probing has been proposed for measuring interference in enterprise WLANs. Having said that, there are a few points worth highlighting in regards to this approach:

- *Tight Centralized Control:* The need for tight AP time synchronization, silenc-

artificially introduced delays in between our interference tests. This was to prevent crashes of the wireless radio firmware, which we observed was unstable when interference tests were conducted back-to-back

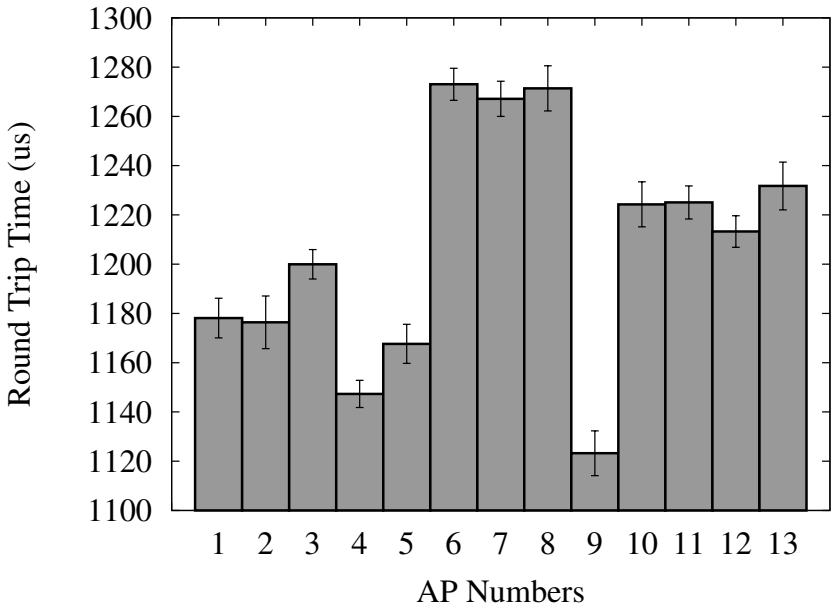

Figure 6.18: Average round-trip time as measured at the controller for a 3 hour period

ing, and modifications to the lowest layers of the networking stack, all make implementing micro-probing an engineering challenge. In this work, we show how we can overcome these challenges using a variety of techniques that put together, demonstrate the real-world feasibility of micro-probing.

- *Milli-second Level Sampling:* Micro-Probing measures interference at millisecond level timescales. Such small sampling intervals only allow it to capture a snapshot of interference between link pairs in the network. Therefore, micro-probing may not capture the mean impact of interference between links[11]. In such situations, the mean impact of interference can be derived by re-measuring interference at different times and computing the mean across these measurements as an estimate of pairwise interference between links.

- *Centralized enterprise WLANs:* Our current implementation of micro-probing targets centralized enterprise WLANs and does not apply to decentralized networks. We believe this design choice to be reasonable as a majority of the enterprise WLAN industry has begun shifting towards centralized WLANs for reasons outlined in Chapter 3.

[11]However, we have shown that for most links, sampling at this timescale is sufficient to characterize interference between links

- *Single Administrative Domain:* Micro-probing functions across a single administrative domain, i.e. all APs are under administrative control of a single enterprise. To allow it to function across multiple administrative domains, mechanisms such as those proposed for IEEE 802.11v are necessary that define a standard upon which heterogeneous enterprise WLANs are able to communicate.

Micro-Probing can be applied to other wireless systems as well, such as wireless mesh networks (WMNs), sensor networks, and agile spectrum sharing systems such as cognitive radio. For WMNs, while techniques such as silencing are applicable to clean the air for interference tests, other mechanisms such as synchronization require designing new techniques that can operate without the help of a wired backbone. In this regard, approaches such as beacon-based synchronization can be applied to achieve the same effect [51]. Other mechanisms such as the use of MAC Service Time (MST) for discovering carrier-sensing interference can be applied to decentralized WMNs.

Aside from it's application towards measuring interference for various wireless systems, the core components of micro-probing have general application to other problems as well. For instance, the silencing technique proposed in this work has been successfully applied to the problem of increasing VoIP client capacity for 802.11 networks with legacy clients [71]. Furthermore, the MAC Service time metric can also be used to estimate the total load being experienced by an AP [86].

Chapter 7

Overcast: Supporting VoIP mobility using Conflict Graphs

In this chapter, we present a scheme that uses the micro-probing system in chapter 6 to optimize performance for *continuously* mobile VoIP clients. Note that while we focus on VoIP in this chapter, the proposed system is equally applicable to other delay-sensitive applications as well (e.g., video). Continuous mobility is defined as mobility where a user actively uses the network as he or she moves about in the building. Supporting continuous mobility for VoIP clients is a challenging problem [94]. This stems from two factors[1]: (1) Handoff delays that cause service disruptions, and (2) Interference from co-located devices that increases VoIP loss rate and end-to-end delay. VoIP performance is determined by two metrics: loss rate and delay jitter. For VoIP, losses are due to packet losses as well as losses from packets that arrive too late at the client, i.e. those that exceed the delay budget for VoIP applications (typically 200ms). In this chapter, we present a system (dubbed 'Overcast') that addresses these problems, providing good Quality-of-Service (QoS) to continuously mobile VoIP clients even in the presence of interference.

The rest of this chapter is organized as follows. In Section 7.1, we motivate the problem of supporting continuous mobility for VoIP clients in enterprise WLANs. The design goals for our proposed approach are discussed in Section 7.2. In Section 7.3, we quantify mobile VoIP performance in present day enterprise WLANs. We then present the architecture of Overcast in Section 7.4 and study the impact of different AP selection algorithms in Section 7.5. We evaluate the performance of Overcast in Section 7.6, and end with related work and discussion in Sections 7.7 and 7.8, respectively.

[1]Note that we assume dense deployments and blanket wireless coverage in the enterprise, and thus ignore problems arising from coverage holes and poor signal quality

7.1 Motivation

Falling prices and demand for a mobile workforce have caused a proliferation of wireless LANs in modern enterprises [113]. In recent years, the emergence of new usage paradigms such as *continuous mobility* and an interest in applications such as Internet telephony (e.g. skype) and video are beginning to place additional demands on and create new challenges for IEEE 802.11 networks.

Recent growth in the use of smart-phones with large screens and greater processing power has spurred a demand for media rich applications that transmit voice, video, and other delay-sensitive content to mobile phones [91]. This has created a myriad of challenges for network designers as the IEEE 802.11 standard is not well-suited to such applications. While standards such as IEEE 802.11e aim to support real-time applications, they require changes to all devices and do not work in dense wireless environments with many co-located APs.

Continuous mobility also introduces a number of key challenges, such as the need to accurately track client link quality, the ability to measure interference on fast timescales, and the need to support seamless handoffs at little to no cost to the client. Because realtime applications are delay-sensitive, transient degradations in client performance severely hamper application execution.

Researchers have attempted to address the challenges of continuous mobility at all layers of the network stack, from the application layer [80], to the physical layer [91]. In this chapter, we study the design of a comprehensive system to support continuous mobility. We present the architectural requirements for supporting continuous mobility and show how they can be realized in today's enterprise WLANs.

Performing hand-offs efficiently between APs is necessary to support continuous mobility. Unfortunately, efficient hand-offs are challenging in 802.11 networks. In a typical 802.11 network, the client is responsible for associating and handing-off between APs. This requires proactively scanning for neighbouring APs, and re-associating to a new AP when required. This process has been known to cause excessive delays and prevent the correct operation of VoIP applications. This motivates a new approach to designing 802.11 networks that can support VoIP applications while the clients are on-the-go.

7.2 Design Goals

A large body of research has studied techniques for supporting VoIP on IEEE 802.11 networks [116, 58]. However, the assumptions and target scenario in these works are

significantly different from that addressed in our work. Our aim is to address the problem of continuous mobility for VoIP clients where interference can potentially impact the performance of realtime applications and can also change rapidly (due to client mobility) during the course of a VoIP session. In this section, we define the precise requirements for our target scenario.

- **Continuous Mobility** is defined as mobility where a client uses the network while on-the-go. This is becoming increasingly common for VoIP applications running on smartphones and is challenging to support on existing WLANs. Recent work on multi-channel enterprise WLANs indicates that handoff delays of up to 1.5 seconds are not uncommon [94], causing disruptions to VoIP applications on such networks (VoIP delay budgets are typically up to 200 ms). Standards such as IEEE 802.11k aim to reduce hand-off delays but require client modifications.

- **Robustness to Interference:** VoIP applications are sensitive to both delay and loss. While they can tolerate a small amount of loss (up to 10%), anything greater can disrupt service. Interference can severely impact client performance [48], making it important to design a WLAN system that is robust to interference in order to support VoIP clients in the enterprise.

- **Support for Legacy Clients:** Deploying new hardware and upgrading NIC software on the clients is expensive and impractical [94]. Therefore, by design, we require the WLAN system to support VoIP mobility without requiring client modifications and thus provide backwards-compatibility with existing IEEE 802.11 standards.

- **Scalability:** At any time, a large number of VoIP users may be simultaneously using the network (e.g. in a conference room setting). Therefore, scaling to a large number of VoIP users is crucial in such situations. Furthermore, non-VoIP traffic should not suffer severely as a result of supporting VoIP clients. In other words, the enterprise WLAN should maximize spectral efficiency.

As we discuss in Section 7.7, existing WLAN systems fail to meet the requirements outlined above, thus motivating the design of a new architecture to support mobile VoIP applications. In the next section, we characterize existing WLAN systems and highlight problems that lead to poor performance for mobile VoIP clients in the enterprise. In doing so, we come up with solutions to address these problems and subsequently use these insights to design the Overcast WLAN system in Section 7.4.

7.3 Quantifying Mobile VoIP Performance in Existing WLANs

The goal of this section is to elucidate those aspects of existing WLAN systems that lead to poor performance for VoIP clients in the enterprise. Note that we assume the enterprise WLAN operates in the presence of legacy clients, as per the requirements outlined in Section 7.2.

7.3.1 Methodology

In this study, we identify three challenges arising in existing enterprise WLANs, in the context of mobile VoIP clients. They are: (1) Inter-AP Handoffs, (2) Intra-AP interference, and (3) Inter-AP Interference[2]. We conduct experiments to isolate the effect of each factor on the performance of VoIP clients.

Experimental Setup

Experiments are conducted on the wireless testbed presented in Chapter 5. Testbed nodes act as APs and we use Dell Vostro 1400 laptops to serve as mobile clients. The laptops are equipped with an EMP 8602 (Atheros) card and we create two virtual interfaces using the MADWiFi 0.9.4 driver [3]. One interface acts as a client while the other acts as a sniffer to collect wireless traces on behalf of the client. These wireless traces are post-processed to obtain the statistics for the experiments. The mobility path chosen for these experiments is shown in Figure 7.1. The client starts at point A, moves along the rectangular black path and returns back to point A. We repeat each experiment five times to determine the mean performance for the VoIP client. All experiments are carried out on the 5.8 GHz band (using IEEE 802.11a), at a data rate of 6 Mbps.

As explained earlier, VoIP losses are a combination of packet losses and losses from packets arriving too late at the client. In all our experiments, we found that losses due to excessively delayed packets were almost negligible (we present some results in Section 7.6.3). Therefore, packet reception rate (or conversely, packet loss rate) are good indicators of VoIP performance. Hence, we choose packet reception rate as the metric for our experiments.

[2]Inter-AP interference implies both interference between APs as well as interference caused by APs on neighbouring clients

[3]Virtual interfaces allow us to simultaneously run the single physical radio in two wireless modes

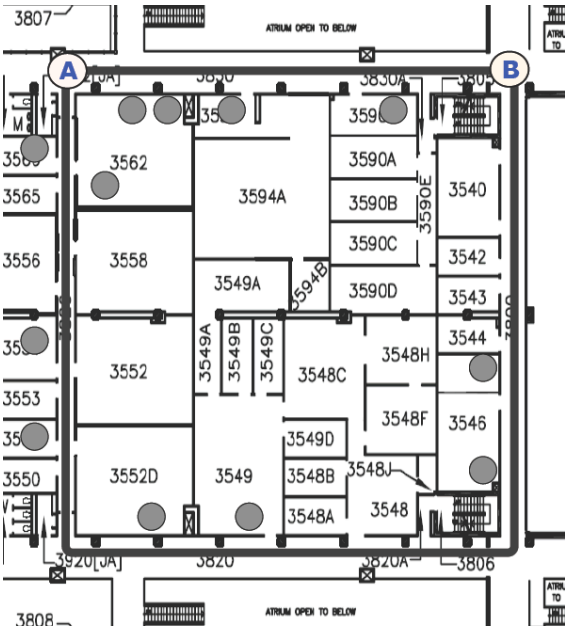

Figure 7.1: Mobility paths for the VoIP client. One path starts at point A, moves to point B, and then reverses back to point A. The second path starts at point A, follows the rectangular black line, and returns back to point A. Grey dots represent APs deployed in the neighbourhood of the paths.

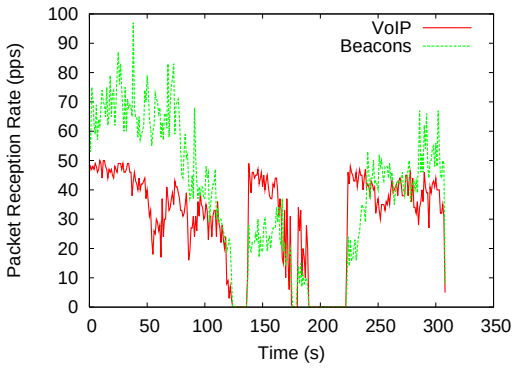

Figure 7.2: Multi-Channel Commercial Network

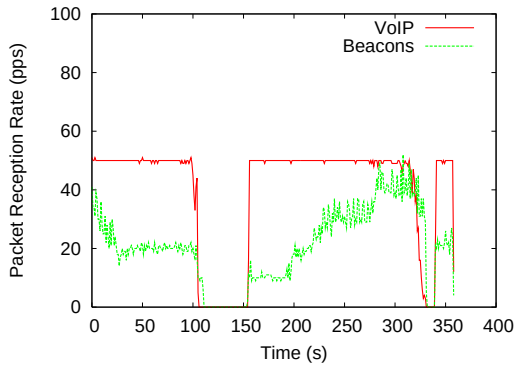

Figure 7.3: Multi-channel Wireless testbed

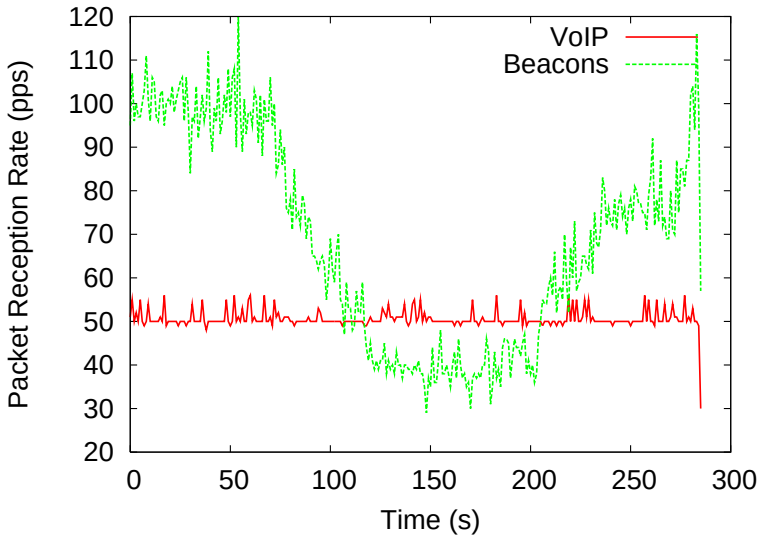

Figure 7.4: Single-channel Wireless testbed

7.3.2 Handoff Delay

The multi-channel design of WLANs has been widely adopted by many enterprise WLAN vendors [1, 3]. In the multi-channel design, APs have a single radio and are tuned to allow maximum channel re-use (thereby minimizing co-channel interference). In such a design, unmodified clients are required to scan for APs when handing-off and re-associate themselves as they move about in the enterprise. In this scenario, we would like to determine the impact of the handoff process on the performance of an on-going VoIP session at the client.

We start our investigation by studying VoIP performance on a commercial enterprise

network. We do this by associating to the department network deployed in our computer science building [1]. We choose the mobility path outlined in Figure 7.1 (the rectangular black path) and have a VoIP client walk along that path while it is connected to the network. We collect wireless traces on the client and plot the number of packets received per second from the source throughout the duration of the experiment (see Figure 7.2). The sending rate of the VoIP stream is $50pps$ with a packet size of 20 bytes.

Observe that the client initially gets a throughput of $50pps$ but it gradually drops as the client moves away from the AP. At a certain point, the throughput drops to zero and the client disconnects from the network. Once it re-establishes connectivity, this process repeats. Note that the duration of disconnection is between 30 and 50 seconds. We explain the underlying cause of this excessive delay a little later in this section.

To ensure that this behavior is not an artifact of using the commercial wireless network [4], we repeat this experiment with our testbed nodes that are on the mobility path of the client. We verify the absence of coverage holes and hand tune the channels to maximize channel re-use. Figure 7.3 presents the result of running this experiment. We no longer see the gradual degradation in performance observed in the previous case[5]. However, we still see long gaps in the traces where the client is not connected to the network. To investigate the gaps more thoroughly, we re-ran the experiment with mobile sniffers on each channel and moved the sniffers alongside the client[6].

Our analysis indicates the following behavior: When the client loses connectivity, it initially tries to re-associate with the same AP it lost connectivity with by transmitting Re-Association requests to that AP. After timing out (after ≈ 15 seconds), it sends Authentication requests to that same AP (for ≈ 1 second)[7]. Once that fails, it switches to the next closest channel and begins to send Probe requests on that channel (for ≈ 5 secs). It keeps repeating this until it associates to an AP with the same SSID. Since it does not scan only orthogonal channels, it suffers a larger delay in re-connecting to the network. Note that this behavior is independent of the frequency band being used and we are likely to observe the same behavior on other bands as well.

It is clear from the results discussed above that the multi-channel design is not well-suited to support VoIP mobility in the presence of unmodified clients. Even if some

[4]We ensure blanket coverage along the mobility path

[5]The gradual degradation occurred in the former case because the commercial APs are performing data rate adaptation that causes the channel quality to fluctuate as the client is moving around. Unfortunately, we are not aware of what algorithm the APs use for adjusting their data rate

[6]The client's virtual interface only sniffs traffic on the client's channel

[7]Note that this is the sticky behavior of clients that try to avoid the cost of switching between APs to prevent service disruption

client adaptors aggressively (or proactively) scan for neighbouring networks, assuming this behavior inhibits widespread application to all client platforms. This requires us to rethink the process of Association and Handoffs in enterprise WLANs.

Single channel WLANs are a different approach to designing enterprise WLAN networks [19, 8]. They assume APs with multiple radios, one for each orthogonal channel/frequency. By design, clients affiliate to the network on one channel and remain on that channel for the duration of their connection. APs advertise a common ESSID and MAC address and clients do not re-associate with the network. The infrastructure decides the AP through which the client communicates to the network. This architecture is compelling as it reduces the handoff cost to zero.

We are interested in determining how effectively single channel WLANs are able to support mobile VoIP clients in the enterprise. We instrument the AP version of the Intel 2915ABG wireless driver to broadcast Beacons with identical ESSID and MAC addresses. We also implement a controller running on a desktop machine that interfaces with the APs. When a client attempts to associate with the network, the AP sends the corresponding Association Request to the controller, which upon receiving requests from all APs chooses one of them to serve the client. Once associated, any re-association is handled seamlessly by the controller and the client is not required to scan for APs any longer. Thus, in theory, inter-AP handoffs are of zero cost to the client. To validate this, we perform the same experiment we did for the multi-channel case. The result of the experiment is shown in Figure 7.6. Observe that the client now gets the desired VoIP rate of $50pps$ from the source, indicating that handoffs no longer impact VoIP performance. This is a key feature of Overcast and we discuss its implementation details in Section 7.4.

7.3.3 Impact of Interference

As a VoIP client moves about in the building, it may encounter regions where there is a lot of wireless traffic. This is common in single channel WLANs where the APs share all the available orthogonal channels. In such a scenario, a VoIP client that performed well earlier may now suffer due to interference from co-located APs. Figure 7.5 illustrates this behavior[8]. When the client enters a congested region of the network, its throughput drops to zero and remains there until it moves out of that region. Our investigation reveals two key causes of poor client performance in these scenarios, which are discussed further.

[8]Note that the interferers are transmitting saturated backlogged traffic on-the-air

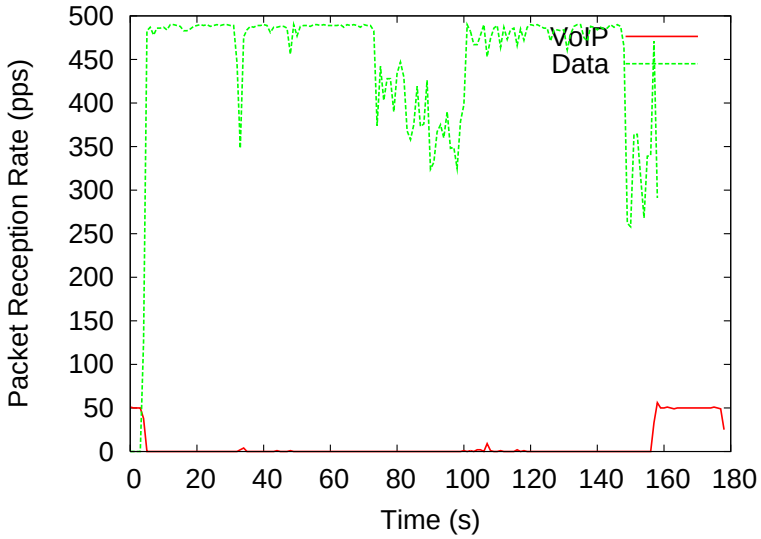

Figure 7.5: Performance of 802.11 client on a single channel in the presence of background interference

Intra-AP Contention

When a VoIP client is served by an AP that is also serving a large volume of non-realtime data traffic, it may experience congestion-related losses. In such cases, both realtime and non-realtime traffic share the same queue inside the kernel. Because the traffic is served in the order in which it arrives, realtime traffic has to wait for data packets ahead in the queue to be transmitted. This causes VoIP packets to suffer excessive queueing delay. In addition, when the kernel's queue becomes full, any subsequent VoIP packets arriving at the AP are dropped. This leads to poor client performance.

Solution: We address this problem by implementing an 802.11e-like mechanism where the AP driver uses separate queues for realtime and non-realtime traffic. The details of this approach are discussed in Section 7.4.4. Figure 7.6 shows the gains from this approach over the single queue case. However, note that the client still does not achieve the target rate of $50pps$. This is caused by inter-AP interference which we describe next. We also note that the sending rate for Data traffic falls by more then $50pps$, due to the presence of the VoIP stream. This is because of the additional air-time wasted (by the VoIP stream) in contending for the channel as well as the overhead of exchanging headers (e.g., PHY headers) per packet. Unfortunately, this cannot be avoided without modifying the functionality of the clients.

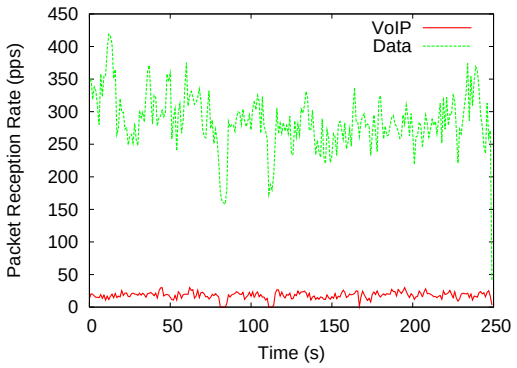

Figure 7.6: Prioritizing VoIP traffic improves performance but still does not yield the target reception rate of 50pps for mobile VoIP clients

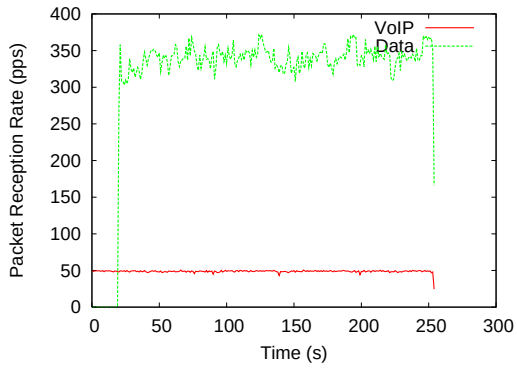

Figure 7.7: Conflict-graph based scheduling with AP prioritization yields the target (50pps) reception rate for the mobile VoIP client

Inter-AP Interference

In a single channel WLAN, interference from co-located APs is inevitable. Despite prioritizing VoIP traffic at the APs, traffic from neighbouring APs can still have a negative impact on VoIP client performance. This is why we still observe poor performance even after traffic prioritization is implemented at the AP.

Solution: We address this problem by scheduling interfering APs in separate time slots. This requires a priori information on interference patterns between APs. Methods to collect this information and details of the scheduler are discussed in Section 7.4. Figure 7.7 shows the performance of the VoIP client after running the scheduler and prioritizing VoIP traffic at the APs. The client now receives the desired rate of $50pps$ throughout the run of the experiment, illustrating that the combination of the two approaches discussed above make the system robust to background interference from co-located devices.

Summary

Supporting VoIP mobility in enterprise WLANs is challenging and requires systematically addressing the relevant problems in today's enterprise networks. We outlined three key challenges for mobile VoIP support in this section and presented techniques to overcome these challenges. A summary of the results is shown in Figure 7.8. In the next section, we describe the details of an enterprise WLAN system (dubbed Overcast) that uses the presented techniques to support VoIP mobility in the presence of legacy clients.

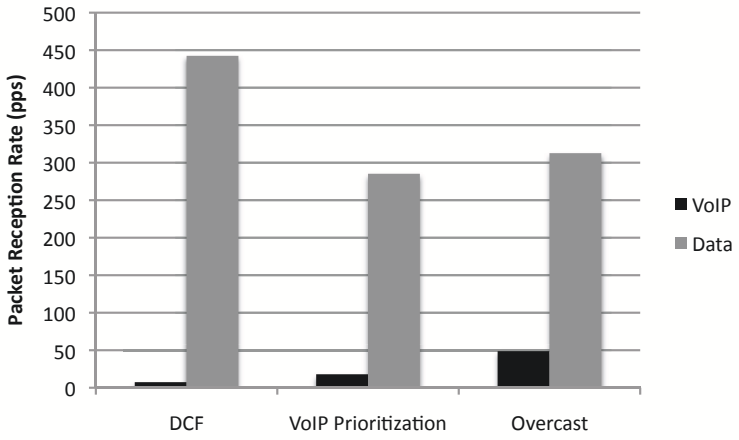

Figure 7.8: Mean packet reception rate across DCF, VoIP prioritization, and VoIP prioritization with conflict graph-based scheduling. For the last case, we observe that we achieve the target rate of 50pps for VoIP traffic

7.4 Architecture

Overcast is an infrastructure only solution, and therefore only requires modifications to the APs and use of a wired controller. It eliminates handoff latencies using a single channel design and manages interference with the help of a conflict graph. The conflict graph is used to schedule AP transmissions. Overcast currently operates on the downlink. The uplink can be handled using the approach proposed in [71], which is discussed in greater detail in Section 7.8. Note that Overcast APs properly implement the IEEE 802.11 standard and thereby do not introduce any unfairness to other co-located devices in the enterprise.

7.4.1 Overview

Overcast is a *single channel* centralized WLAN architecture. Orthogonal channels are used to add capacity instead of mitigate inter-cell interference. All APs broadcast Beacons with the same SSID and MAC address, emulating a single virtual AP cloud. Therefore, from the client's perspective, the entire network is a single virtual AP (as shown in Figure 7.9). The client does not attempt to re-associate because it observes a continuous stream of identical Beacons from all APs. We note that some commercial vendors such as Meru [8] employ a similar approach.

Using the single virtual AP architecture, we implement the following features to

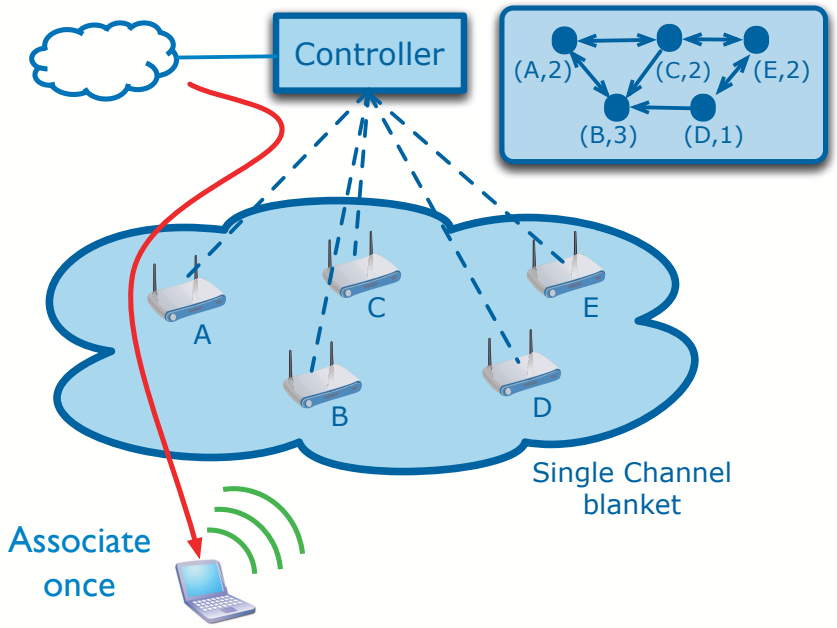

Figure 7.9: High-level view of the Overcast architecture. The client associates only once to the network (through AP A) and the controller seamlessly manages the AP-client link thereafter.

support seamless mobility for VoIP:

- **Centralized Client Association:** Overcast shifts the responsibility of client association to the network infrastructure. The infrastructure maintains statistics on each client and uses this information to choose the most suitable AP for the client. Associations may change at any point if the network determines a more suitable point of attachment for the client.

- **Online interference mapping:** To avoid inter-AP interference, Overcast uses a conflict graph that is measured using the micro-probing approach described in Chapter 6. This conflict graph is periodically re-measured to ensure it contains up-to-date information regarding interference in the network.

- **VoIP aware Scheduling:** Overcast coordinates packet transmissions at the APs to improve performance for VoIP clients suffering from interference and contention in their neighbourhood. As discussed earlier, this can exist due to intra-AP contention or inter-AP interference. For intra-AP contention, we implement traffic prioritization at the AP and for inter-AP interference, we implement centralized AP scheduling at the controller.

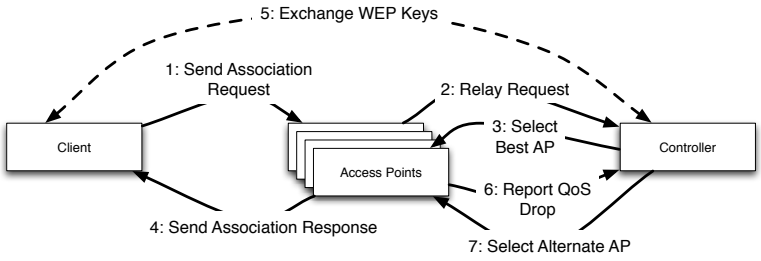

Figure 7.10: The sequence of events that occur when a client associates to the network. Steps 6-7 only occur when the client needs to switch to another AP.

We now describe each of Overcast's features in greater detail.

7.4.2 Client Association

Associating to the Network

The stepwise procedure for connecting to the Overcast system is shown in Figure 7.10. A client connects to the network by sending 802.11 Authentication Requests to the AP(s). All APs receiving the request forward it over the wire to the central controller. At this point, the controller has no information about the client aside from the signal strength seen by each AP that saw the Authentication Request. Therefore, as a first order approach, the controller instructs the AP who observes the strongest signal strength to service the client. It sends an ACK to this AP and a NACK to all other APs that also sent requests. The ACKed AP completes the 802.11 association process with the client and upon completion, sends all the context associated with the client (including encryption keys for WEP) to the controller. The controller stores this context for each client associated to the network[9]. Therefore, practically speaking, the client session terminates at the controller, not the AP. Note that Overcast makes an important distinction between client association and the process of deciding the best AP (i.e. AP selection) for the client. Traditionally, these tasks were combined and performed exclusively by the client.

Creating a path to the client: Aside from authenticating and associating the client to an AP, the controller must also setup a path to allow wired traffic to reach the wireless client. To allow for this, the controller adds an entry into its ARP cache that maps the client's IP address to the MAC address of the AP chosen to serve the client. Thus, any

[9]Once associated, the client requests an IP address that is provided by a DHCP server running on the controller. The client maintains this IP address throughout the time it is connected to the network

traffic destined to the client can now be forwarded to the correct AP. In the future, any changes to client-AP association also requires an update to the ARP cache, which is seamlessly handled by the Overcast controller at the time the client is switched to the new AP.

Collecting Client Statistics

An essential requirement for maintaining good Quality-of-Service (QoS) for VoIP clients, is the need to collect information on their performance. Several performance metrics may be collected for this purpose and which ones depend on the requirements of the optimization algorithms implemented for the Overcast system. We discuss individual metrics when we describe the optimization algorithms for Overcast. APs periodically report such statistical information to the controller. Using this information, optimization algorithms evaluate the current configuration of the client and decide whether a better configuration is possible. We describe two optimization algorithms that use such client statistics in later sections of this chapter. Note that we use EWMA for all our metrics with a weighting factor $\alpha = 0.9$, to give preference to more recently collected statistics.

Performing AP Selection

Overcast performs AP selection for a client once it has associated to the network and the controller has collected sufficient statistics on the client. There are a variety of AP selection algorithms that can be used with Overcast [111]. We discuss a few algorithms we evaluated in the next Section. Once an AP has been selected, Overcast uses the process of *make-before-break*, where the controller first sends the client's context to the new AP, instructing it to begin serving the client. Then, after a small delay (lasting $\approx$ 3-4 ms), it instructs the old AP to stop serving the client. During this entire process, the client is oblivious to the change and does not experience any delay that could potentially degrade VoIP performance.

7.4.3 Interference Mapping

At the heart of the Overcast system is an interference mapping (IM) engine. The IM engine is responsible for discovering downlink interference (or conflicts) between APs and between APs and clients in the network[10]. We choose the micro-probing approach

[10]We currently do not support uplink conflicts in Overcast

presented in Chapter 6 because of its ability to map interference in an online network without requiring client modifications.

Incorporating the IM engine into Overcast is challenging because it involves active measurements that can potentially interfere with the operation of the WLAN and care must be taken to avoid this from happening. Having said that, passive measurements (using Data traffic) can also be used to reduce the number of active measurements that need to be performed for mapping interference. However, in this work, we only employ active measurements to generate the interference map. We perform measurements for AP-AP and AP-client interference at system bootstrap time and periodically re-measure AP-client interference between links to ensure we have the most up-to-date interference information on the clients. The measurement interval α is a tuning parameter that determines how aggressively the system performs interference measurements.

7.4.4 Traffic Scheduling

We now describe the two techniques we implemented to manage intra-AP contention and inter-AP interference in the Overcast system. The first approach is referred to as *AP Prioritization* and the second approach is called *Centralized Scheduling*.

AP Prioritization

In Overcast, APs serve both realtime and non-realtime clients. As discussed in Section 7.3, realtime traffic can suffer large queueing delays and losses due to kernel buffer overflows. To avoid these problems for VoIP traffic, we implement a prioritization scheme at the AP. In particular, we use two queues, one for realtime traffic and the other for non-realtime traffic. Packets in the realtime queue are served before packets in the non-realtime queue. Moreover, the driver prevents the kernel from overwhelming the circular ring buffer sitting in-between the driver and firmware. The number of packets passed to the firmware (at any given point in time) is always less than the size of the circular ring buffer. Excess packets are stored in the driver's queues.

The above approach is similar in spirit to some of the mechanisms proposed for the IEEE 802.11e standard. However, there are two reasons we do not consider 802.11e for Overcast. First, 802.11e is not well-supported on commodity hardware. Some parts of the standard are optional and therefore may not even be available in 802.11e compatible hardware. Second, IEEE 802.11e requires client modifications and thus does not support legacy clients based on the 802.11a/b/g standards. Based on these observations, 802.11e is not a suitable approach for the Overcast system.

Algorithm 2 Greedy Scheduling Algorithm

```
1: A = {a_1, a_2, ..., a_i} /* Set of Active APs */
2: S = {s_1, s_2, ..., s_i} /* Set of Scheduler Slots */
3: CurrSlot = 0
4: for i = 1 ... | A | do
5:    if a_i is unmarked then
6:       CurrSlot = CurrSlot + 1
7:       Mark a_i
8:       Add a_i to s_CurrSlot
9:       for k = 1 ... | A | do
10:         if a_k is unmarked and a_k does not conflict with a_i then
11:            Mark a_k
12:            Add a_k to s_CurrSlot
13:         end if
14:      end for
15:   end if
16: end for
```

Centralized Scheduling

In a single channel WLAN, neighbouring APs are likely sources of interference[11]. Given that we have measured the interference map for the network, we now use a scheduler (co-located with the controller) to coordinate the transmissions at the APs. Our scheduling mechanism divides time into equal sized slots (of a certain size) and schedules APs such that no two conflicting APs (that interfere due to inter-AP or AP-client conflict) are scheduled in the same slot. The scheduler only considers APs that are actively carrying downlink traffic. APs periodically report their traffic load to the central controller which maintains an exponential average of such information. Using this information, the scheduler constructs a schedule for the APs using Algorithm 2. Note that an AP is 'marked' if it has already been assigned a slot.

Once a schedule is constructed, the scheduler implements it as follows. For each slot, it sends a broadcast packet with the identifiers of all APs assigned to that slot. It also adds the slot length (in ms) to the packet. The broadcast helps to synchronize the APs to the current scheduling slot[12]. APs upon receiving the broadcast packet, determine

[11]Note that the IEEE 802.11e standard discussed earlier cannot alleviate such interference because it does not handle interactions between BSSes

[12]Note that perfect synchronization is not required between APs for scheduling purposes. Nevertheless, our synchronization approach has been shown to be accurate on the order of tens of microseconds [30].

whether or not they are scheduled for the current slot (by searching for their id in the packet). If they are scheduled, they start sending any queued up packets and continue doing so until the slot duration comes to an end. If they are not scheduled, they block and wait for the next broadcast packet from the scheduler. Note that due to the small packet size of the broadcast packet and the data rates supported on the wired networks, the overhead of sending the broadcast packet is almost negligible. This is similar to the 'epoch' scheduling approach proposed for the CENTAUR system [109]. However, unlike CENTAUR, Overcast does not queue packets at the central controller, but instead lets the APs implement the queueing functionality. This inherently allows Overcast to scale to larger traffic volumes and more clients.

Traffic loads in the network are subject to change, as are interference patterns. Therefore, a new schedule will need to be periodically re-computed by the scheduler. How often this is done depends on how quickly these parameters change in the network. In practice, we find that the overhead of re-computing the schedule is negligible and we therefore re-compute the schedule after every iteration of the scheduler (which typically lasts 50 - 60 ms). An iteration of the scheduler is the amount of time it takes to completely execute the generated schedule exactly once.

Scalability of Overcast

In this section, we analyze the scalability aspects of Overcast. Our analysis serves to provide some intuition on how many clients the Overcast system is able to support. However, we also perform scalability experiments (as part of our evaluation) to practically study Overcast's scalability properties as well.

The parameters we consider in our analysis are:

- **C** : Number of orthogonal channels
- **J** : Jitter Buffer length (in ms)
- **S** : Scheduler slot length (in ms)
- **R** : Packetization interval of VoIP stream (in ms)
- **T** : Transmission duration of a VoIP packet at a fixed wireless data rate (in ms)

By design, we ensure real-time VoIP traffic gets priority over non-realtime Data traffic. Therefore, if an AP has even one outstanding VoIP packet, it is guaranteed to be sent

in the next time slot alloted to that AP. Our objective is to ensure that all VoIP packets received by the client are spaced apart by no more than the Jitter buffer length specified for the VoIP codec being used. In other words, the maximum number of slots an AP can wait before being scheduled is $J \div S$. To ensure that there is sufficient time to serve all clients before reaching the end of the jitter buffer interval, the number of slots an AP can wait is:

$$W = J/S - 1 \tag{7.1}$$

To simplify our analysis, we assume that the channel quality between the AP and client is good, thus ensuring that whenever the AP transmits a packet to the client, it is correctly received. Also, for the sake of simplicity, assume the conflict graph is a clique, so that only one AP occupies a scheduler slot at a time [13]. The packetization interval of the VoIP stream is R. Therefore, within an interval of W, the expected number of VoIP packets received for a single VoIP stream is

$$E = W * S/R \tag{7.2}$$

Given a slot duration of S, the maximum number of VoIP packets that can be transmitted within a slot duration is $P = S \div T$. Therefore, the maximum number of clients an AP can support on a single channel is:

$$M = P/E \tag{7.3}$$

If there are C orthogonal channels, a single AP can support up to $M * C$ clients using all available channels. Plugging in (J = 60ms, C = 3, S = 5ms, R = 20ms, T = 234us), we obtain $\approx 7 * 3 = 21$ clients per AP. This is significantly greater than the number of clients supported by 802.11's existing DCF mechanism, which is known to support only $2 - 3$ clients in interference-limited scenarios [71].

7.5 What is the impact of AP Selection?

There are many AP selection algorithms that can be designed for use with Overcast [111]. In this section, we are interested in answering the question, 'Does the choice of AP selection algorithm have a significant impact on the performance of the VoIP client?' We consider three metrics in order to answer this question. Two of them are popularly used

[13] This analysis extends to multiple APs occupying a single slot as well

in practice, while the third uses the conflict graph to make AP selection decisions, taking into account interference between links in the network. We briefly describe these metrics further:

RSSI-based Selection: In this approach, the algorithm selects the AP that sees the highest received signal strength (measured as RSSI) from the client. An exponential average of the RSSI observed for the client is maintained at the controller. This metric is popularly used in client NICs to decide which AP to select when associating to the network. Note that we use uplink RSSI as a predictor of client throughput in both the uplink and downlink directions. Given that we work with a dense AP deployment, prior work has shown that uplink RSSI is a good predictor of performance in both the uplink and downlink directions [94].

BRR-based Selection: In this approach, the algorithm selects the AP that provides the best downlink delivery ratio to the client. To measure downlink delivery ratio, all candidate APs are instructed to transmit a series of probes to the client (one after the other). APs report back the delivery ratio of these probes to the controller. The controller maintains an exponential average of the downlink delivery ratio values to the client and chooses the AP with the highest delivery ratio. Probe transmissions at the AP last $\approx 15-20$ ms, and therefore constitute a modest measurement overhead (there are typically 5 - 6 APs in the neighbourhood of the client).

Conflict-based Selection: In this approach, the algorithm also uses interference information available in the conflict graph. It assesses client performance along two axes: (1) Quality of the link to the AP, and (2) Degree of inter-AP interference at the AP. Link quality is assessed using the RSSI metric discussed above. Once a set of 'good' links are chosen, the algorithm then selects the AP that minimizes the sum total number of conflicts with neighbouring APs, with the goal of maximizing the amount of airtime a client gets from the AP. Note that this algorithm also requires load information from each AP in order to estimate conflicts.

Given that the VoIP client is already affiliated with the network and sending traffic every *t* seconds, where *t* is the packetization interval of the encoding scheme, we can collect almost all of our statistics without introducing any control traffic in the network. Since all access points can listen to all traffic on all channels, we passively collect statistics from all APs in the vicinity of a client. The only exception is the BRR algorithm where we must compute the downlink delivery ratio from each candidate AP to the client. Despite this overhead, our evaluation shows similar performance for BRR with no measurable gains over the RSSI or Conflict-based approach. Therefore, for conciseness, we omit presenting the results for BRR in this chapter.

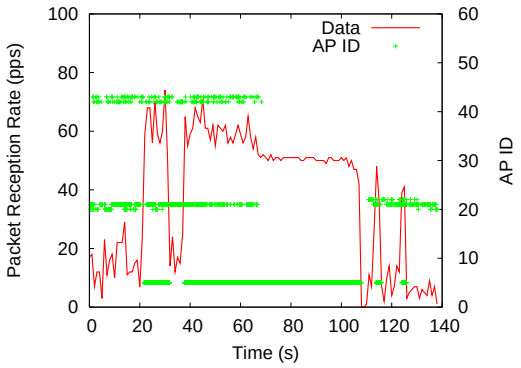

Figure 7.11: Conflict-based

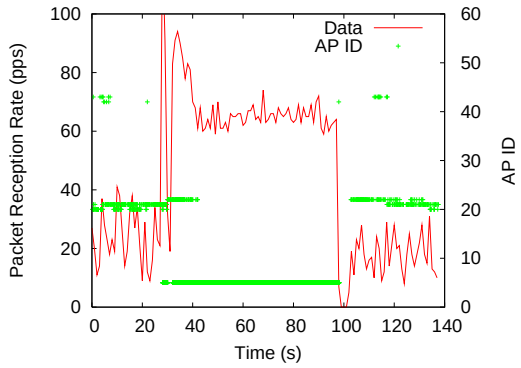

Figure 7.12: RSSI-based

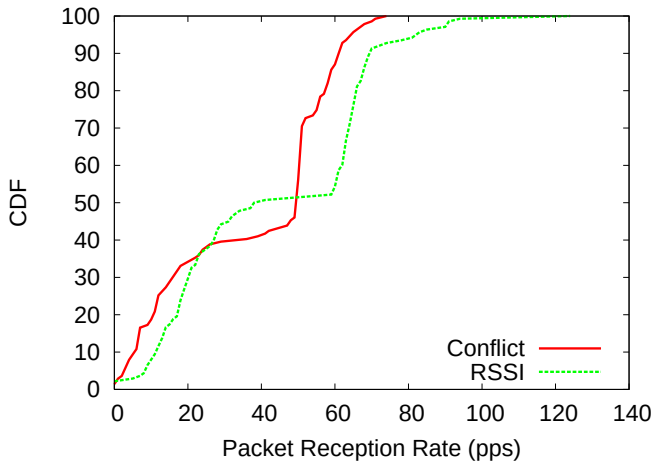

Figure 7.13: CDF of AP Selection Schemes

7.5.1 Experimental Evaluation

We now present experimental results to show how well each of the algorithms described above work in conjunction with Overcast. To isolate the impact of AP selection, we disable the scheduler and only perform AP selection. Furthermore, we generate background interference in our experiments to gauge the benefits of being interference-aware in the AP selection process. In our experiments, the VoIP client walks along the mobility shown in Figure 7.1. It starts at point A (region of high interference), moves to point B (region of low interference), and returns back along the same straight path to point A. We generate VoIP traffic at a rate of $50pps$ and measure the downlink packet reception rate at the client.

Figures 7.11 and 7.12 present time series plots of the throughput a client obtains using each of the AP selection schemes. Observe that on both time series graphs, the client

initially obtains low throughput (when it is in a high interference region). Its throughput increases as it moves away from its initial position (to a low interference region) and then drops down again as it returns back to its initial position. Looking at these graphs, there isn't any significant difference in the performance of either of the AP selection schemes. The CDF of the client throughput (Figure 7.13) validates this observation. We performed similar experiments with the client starting at different locations and moving along different mobility paths and obtained similar results.

From this extensive experimentation, we conclude that the choice of AP selection algorithm does not yield any measurable gains for VoIP traffic, in the context of single channel WLANs. This is because the degree of contention (or free air-time) at co-located candidate APs is almost the same. Given that the degree of contention and link quality are the two important criteria used in selecting the best AP, neither of the possible candidate APs offer significant advantages in these domains. Therefore, we obtain a strong *negative result* and find that a simple signal strength (or RSSI) metric is sufficient for AP selection. In the rest of this chapter, we use the RSSI-based AP selection algorithm to evaluate the Overcast system.

7.6 Evaluation

In this section, we evaluate Overcast on a number of different criteria. Our aim is to show the following:

- Overcast is able to provide consistent performance to mobile VoIP clients regardless of their location.
- Overcast provides the desired quality of service (QoS) to VoIP clients even as interference increases in their neighbourhood.
- Overcast scales to large numbers of VoIP clients that are simultaneously moving about in the enterprise.

7.6.1 Methodology

We evaluate Overcast on the 38 node wireless testbed described in Chapter 5. We use the Intel interface to act as the AP and modified the ipw-2200 driver (for the Intel card) to implement the features described in Section 7.4. For most of our evaluation, we use Dell Vostro 1400 laptops to act as clients. However, for the scalability experiments, we use

a variety of different platforms, from laptops (running Linux and Windows) to iPhones running the OS X iPhone operating system. We obtain similar results on all platforms.

For the mobility experiments, we choose the path shown in Figure 7.1. In this path, the client(s) start at point A, move(s) along the rectangular black path and return(s) to point A. The speed of movement is approximately $4kph$, mimicking the average human walking speed. We also evaluated other paths on our testbed and obtained similar results, indicating that Overcast is relatively insensitive to the mobility path chosen by the client. Unless otherwise stated, all our mobility experiments are repeated 5 times and we show 95% confidence intervals for each result.

We compare the performance of Overcast against two other schemes. The first is a multi-channel scheme (termed *M-channel*) that mimics a typical multi-channel enterprise network similar to what was discussed in Section 7.3.. For M-channel, we hand tune 3 orthogonal frequencies across the APs along the mobility path to maximize frequency re-use (to mimic real-world deployments). The second scheme (termed *No-Scheduler*) is identical to Overcast except that it does not use the centralized scheduler (discussed in Section 7.4.4) to mitigate inter-AP interference. The goal of this scheme is to elucidate how well VoIP clients perform in the absence of an optimization scheme that uses the conflict graph to optimize VoIP performance. This is analogous to a single channel WLAN that uses APs which implement only 802.11e-like enhancements to optimize VoIP traffic.

In our experiments, we generate VoIP traffic using UDP streams that mimic the popularly used G.729 VoIP codec. The packet arrival rate is 20 ms and packet size is 20 bytes, that results in a sending rate of 50 packets per second. This traffic originates at the controller and terminates at the client. By contrast, interferers are assumed to be backlogged, sending UDP traffic at the highest possible rate, with a packet size of 1400 bytes. The number of interferers varies (from 1-5) as the client moves along the mobility path during the experiment. This represents the worst case for Overcast and our results are therefore a lower bound on its performance.

All our experiments are conducted on the 5.8 GHz (IEEE 802.11a) band and use a fixed data rate of 6 Mbps. In other words, we disable auto-rate adaptation in our evaluation.

In our work, we consider metrics of packet reception rate (computed on a per second basis) and delay jitter to evaluate performance for VoIP traffic. Measuring packet reception rate is equivalent to measuring packet loss rate for VoIP because (as we show later) delayed-induced losses are negligible in our system. We also consider metrics such as the total connectivity time during an experiment and the number of disruptions to evaluate

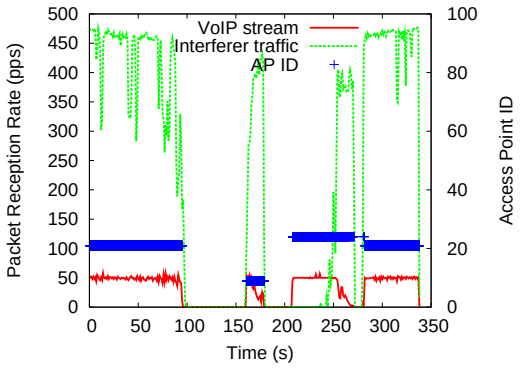

Figure 7.14: M-Channel

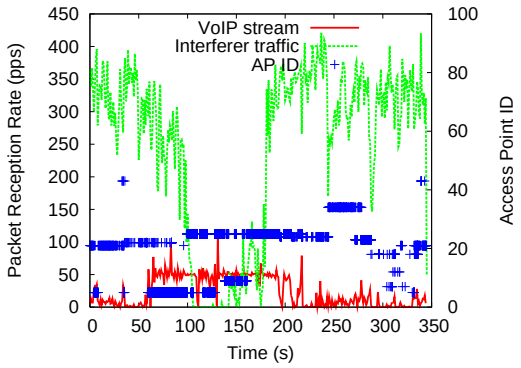

Figure 7.15: No-Scheduler

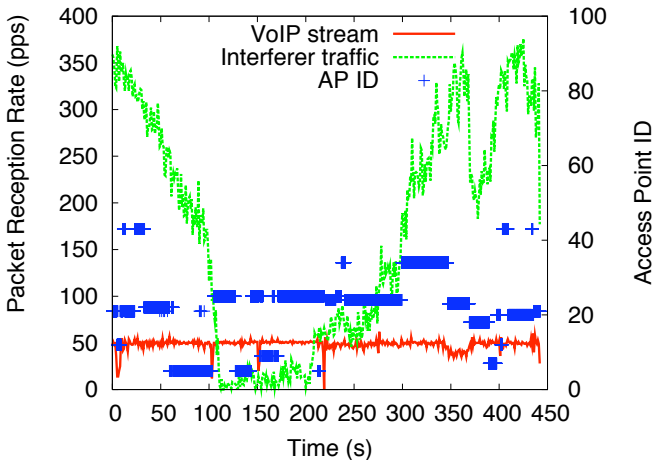

Figure 7.16: Overcast

VoIP performance. We discuss these metrics in detail in Section 7.6.3.

7.6.2 Overview

We start by comparing the performance of Overcast with the other schemes across a single mobility run. In this experiment, a client walks along a particular path and encounters interference from APs broadcasting (non-realtime) data traffic. This experiment is performed 5 times for all schemes and one run for each is shown in Figures 7.14, 7.15, and 7.16. The y-axis on the right of these graphs indicates the id of the AP to which the client is associated.

As discussed in Section 7.3, M-channel suffers frequent disconnections as the client attempts to maintain connectivity to the AP with which it is associated. Note also that M-channel rarely switches between APs, whereas No-Scheduler and Overcast switch

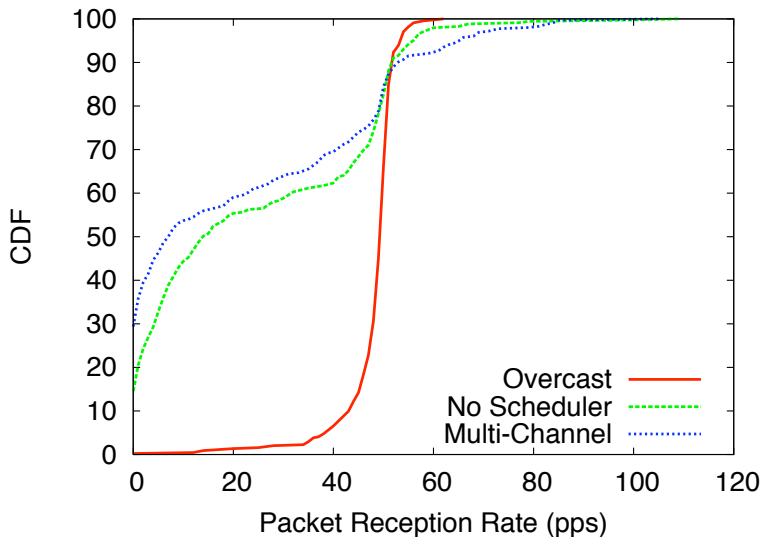

Figure 7.17: CDF of Packet Reception Rate for the three runs shown in Figures 7.14, 7.15, and 7.16

multiple times even within a 10 second interval. The *sticky* nature of the client in the M-channel case is the result of the client performing AP selection (instead of the infrastructure) which not only leads to periods of no connectivity but also periods where the client experiences poor performance and its throughput gradually degrades to zero (e.g. between 250-280s).

No-Scheduler performs the worst when there is a lot of interference traffic in the neighbourhood of the client. As the interference load drops, the client's performance begins to improve and eventually reaches $50pps$ close to the middle of the run. Note that between the intervals $50 - 100$ and $175 - 200$, the client manages to sustain $50pps$ despite the presence of interference traffic. This is because the client no longer suffers from inter-AP interference and instead only experiences intra-AP contention. Because AP prioritization is implemented for No-Scheduler, it does not suffer from intra-AP contention, allowing it to obtain the desired rate in the intervals discussed above.

Finally, the Overcast system performs the best and is able to sustain a packet reception rate close to $50pps$ throughout the mobility run. Notably, it performs an almost equivalent number of AP switches as compared to No-Scheduler (because the same AP selection algorithm runs on both schemes). However, use of the scheduler allows it to eliminate interference from neighbouring APs and provide consistent performance *regardless of location*.

The cumulative distribution function (CDF) of the packet reception rate (Figure 7.17)

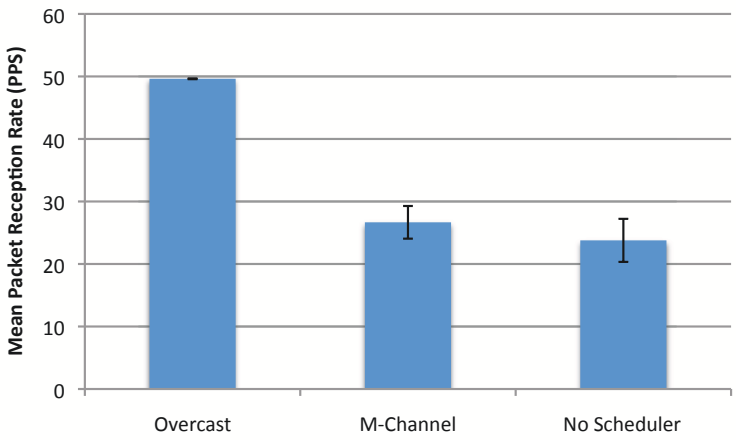

Figure 7.18: Mean Packet Reception Rate across all schemes

provides greater insight into the performance of the three schemes. Overcast operates close to the target packet reception rate for the VoIP stream. In contrast, the other two schemes perform quite poorly, where up to 70% of the traffic is below $40pps$. Note that although the time series results are qualitatively different for the other two schemes, the CDF indicates that the distribution of PRR for the schemes is almost equivalent. Nevertheless, these numbers are unacceptably low for VoIP traffic and these schemes do not provide the Quality of Service (QoS) necessary for supporting VoIP applications in interference limited environments.

7.6.3 VoIP Performance

In this section, we study VoIP performance based on the mean packet reception rate (PRR), delay jitter, and session-related metrics.

Mean Packet Reception Rate

A crucial factor determining VoIP performance is the packet reception rate (or conversely, the loss rate) on the link. For the G.729 codec, VoIP can tolerate losses of $\approx 10\%$ before the call quality becomes unacceptably low. The industry-standard for evaluating a voice call is the Mean Opinion Score (MoS), which ranges from 1 5. A value of 5 implies perfect call quality and a value of 1 implies the inability to communicate. Losses of up to 10% corresponding to an MoS value of 2. In this section, we characterize the

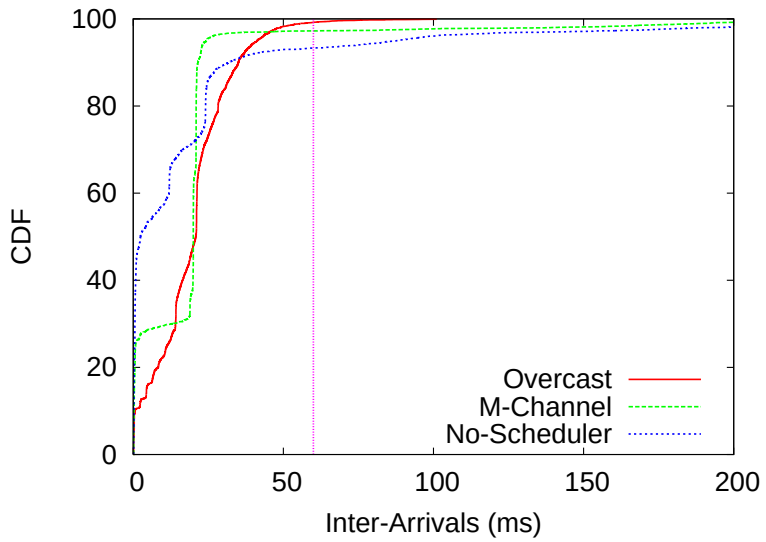

Figure 7.19: Cumulative distribution function of inter-arrival times shows that all packets arrive within the 60ms time specified for the jitter buffer size of the G.729 codec

loss rate for VoIP in terms of the mean packet reception rate for each of the schemes. We perform the same mobility experiments as those discussed in the previous section.

Figure 7.18 presents the mean packet reception rate across all schemes. As before, No-Scheduler performs the worst because the client experiences poor performance in high interference regions of the mobility run. M-Channel improves over No-Scheduler but suffers repeated disruptions in service followed by long periods of inactivity. Overcast is the only scheme that provides good overall performance to the VoIP client. Note that as shown in Figure 7.17, the majority of the mass lies in the $47 - 50pps$ range, across all experimental runs. Note that because our interferers are backlogged, this represents the worst case for Overcast. Given that it is able to maintain the target reception rate for VoIP in this scenario, we expect it to provide the same rate for less aggressive interference scenarios as well.

Delay Jitter

As discussed earlier, delay jitter is an important metric for VoIP applications. If the delay jitter is too high, VoIP clients could suffer from delay-induced losses. Delayed-induced losses are a function of the VoIP codec used. We use the G.729 codec implemented on most VoIP devices and as per convention, assume a jitter buffer length of 60 ms [52]. Therefore, our goal is to observe the span of the inter-arrival time distribution of VoIP

packets at the client. If the span is less than the jitter buffer length, delay-induced losses will be negligible.

Clock Re-synchronization Problems: We point out that in order to collect results for delay jitter, we were required to measure the arrival times of VoIP packets. Our initial goal was to use the high precision hardware clock of the wireless radio to measure these times. However, we observed an unusual behavior when using this clock. Specifically, we observed erratic changes in the values reported by this clock. Our in-depth investigation revealed that this was being caused by the client continuously re-synchronizing its clock based on the TSF time stamp it received in the Beacons of all in range APs[14] . Since all APs in Overcast broadcast the same MAC address and ESSID, the client adjusted its clock every time it received a Beacon from any in range AP. This made timestamps from the radio clock unusable for our experiments. Instead, we resorted to using the host time functionality provided by the Linux kernel. Host time is maintained by the kernel and provides accuracies of up to a millisecond. While not as accurate as the radio's clock, host time turned out to be sufficient for our purposes. We discuss the implications of the radio clock re-synchronization problem at the end of this chapter.

Figure 7.19 plots the CDF of the inter-arrival times of VoIP packets for the different schemes. This result corresponds to the mobility runs performed for the PRR metric shown in Figure 7.18. Note that we only show inter-arrival times for consecutive packets in the trace (which are identified by their sequence numbers). This leads us to omit packets not received during periods of disconnection in the M-Channel case. Therefore, while the CDF for M-channel is promising, it does not capture what happens when the client disconnects from the network. Therefore, in reality, M-channel performs even worse in terms of delayed induced losses, than what is observed in this result.

We draw a vertical line on the point corresponding to $60ms$ for the inter-arrival time. For Overcast, note that almost all packets arrive within $60ms$ of each other (with a span of $\approx 60ms$). In fact, $\approx 75\%$ of the packets arrive within the packetization interval of $20ms$. We repeat the experiment with different topologies and different configurations of interfering APs along the mobility path and obtain similar results. From this result, one can conclude that with Overcast, delay-induced losses for VoIP clients are almost negligible.

Surprisingly, we observe similar results for the M-channel and No-Scheduler schemes. In fact, No-Scheduler performs slightly better than Overcast. This improvement is attributed to the absence of the centralized scheduler in No-Scheduler. In Overcast, the

[14]The client uses the TSF timestamp to adjust its clock and remain in sync with the clock of the associated AP

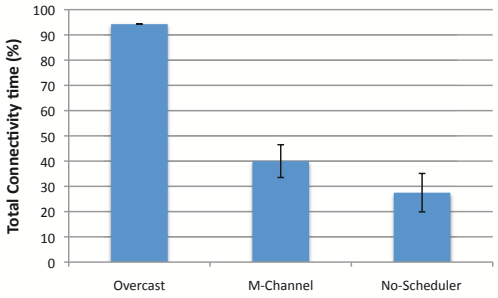

Figure 7.20: Total Connectivity Time

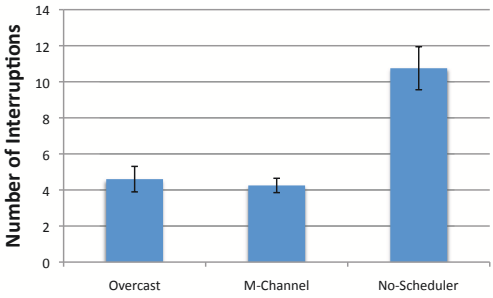

Figure 7.21: Number of Interruptions

scheduler introduces some delay to separate conflicting APs transmissions. This increases the delay between packets and therefore increases the inter-arrival time between packets. However, note that for both M-Channel and No-Scheduler, the span of inter-arrival times is up to $200ms$. This indicates that these schemes suffer from delay induced losses that further degrade VoIP performance. Given that our PRR results earlier do not factor in these losses for those schemes, we assume those results to be optimistic since the actual loss rate for VoIP using these schemes is in fact lower than that seen in the previous results.

Session Characteristics

The longer a VoIP client is able to connect to the network and obtain good service, the better. Metrics such as packet reception rate and delay jitter do not capture the length of a VoIP session. To quantify this, we introduce two metrics: Total connectivity time (as a percentage of the total experiment time) and the number of interruptions that occurred during the run. Total connectivity time is defined as the time the client was able to get acceptable quality of service from the network. Quality of service is defined as in [40]. In particular, a client obtains acceptable quality of service if its MOS value remains above 2. A disruption is said to have occurred if the MOS value falls below 2 for a period of at least three seconds (which is roughly the amount of time it takes to utter a short English sentence). Different MOS threshold values were tested (aside from 2) and for higher values, Overcast performed even better than M-Channel and No-Scheduler. Note that the total connectivity time metric also provides a way to lower bound the performance of the VoIP session and determine the amount of time the client was able to operate above this baseline. Thus, this metric provides us a better sense on the actual performance of the VoIP client across the entire mobility run.

Figure 7.20 presents the result for the total connectivity time of a mobile VoIP client

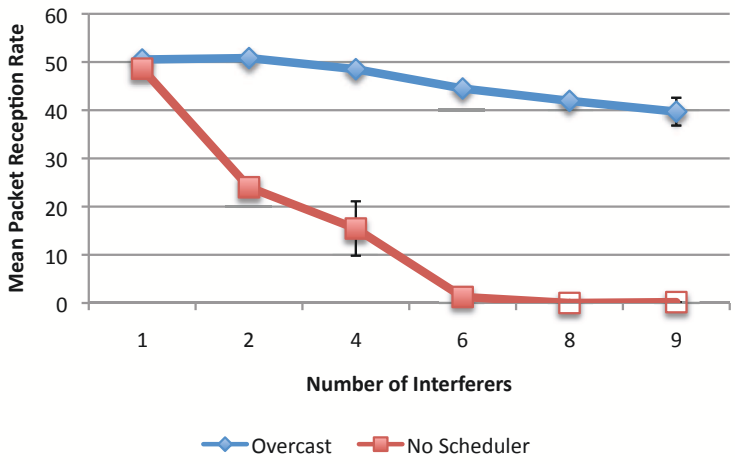

Figure 7.22: Interference/Contention has little effect on the performance of the VoIP client using the Overcast system. However, it suffers severely under the No-Scheduler approach which does not exploit information present in the conflict graph

walking along the same mobility path used in the previous experiments. Observe that No-Scheduler again performs the worst of all the schemes. M-Channel improves total connectivity time by almost 40% compared to No-Scheduler. Overcast yields the greatest total connectivity time, up to 130% greater than M-Channel. This indicates that *Overcast is able to more than double the overall talk time of a VoIP client as compared to multi-channel WLAN systems in widespread use today*.

Figure 7.21 presents results for the mean number of interruptions during the mobility run. No-Scheduler experiences the greater number of interruptions, which is approximately 150% higher than the other two schemes. The performance of M-Channel and Overcast is comparable for this metric, indicating that Overcast does not provide much gain in this dimension. However, note that disruptions in M-channel cause clients to lose connectivity and begin scanning for alternate APs, which is detrimental both in terms of performance as well as in terms of the energy consumed in sending probe requests and switching between channels while searching for an AP. This does not occur for Overcast as the infrastructure performs handoffs on behalf of the client.

Impact of Interference

Mitigating the impact of inter-AP interference is a key objective for Overcast. Therefore, it is important to understand the relationship between the amount of inter-AP interference in the neighbourhood of the client and how it affects VoIP performance. To isolate the

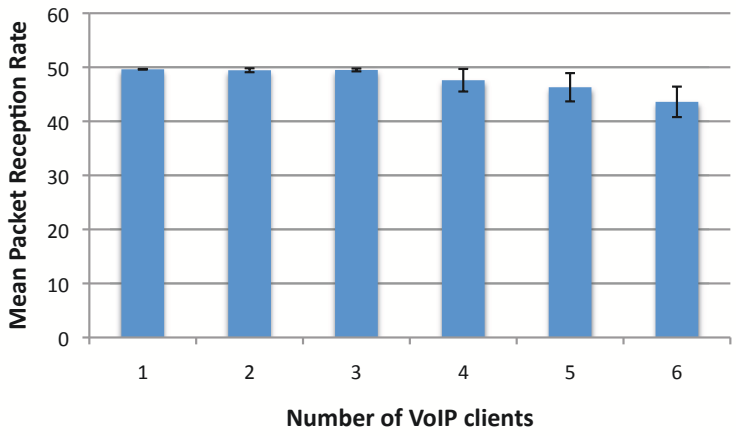

Figure 7.23: Mean packet reception rate for different numbers of VoIP clients

impact of interference from other factors that may also arise due to mobility, we perform experiments with a static client. To remove location-induced biases, we place the client at multiple locations and perform the same experiment. The results we obtain for different locations are similar and we therefore only present one of them in this section.

Figure 7.22 shows the mean packet reception rate for a VoIP client as the number of interferers is increased. We do not plot results for M-Channel, since they are similar to No-Scheduler but scaled up according to the number of orthogonal channels used in the experiment. Note that No-Scheduler's performance drops to almost half as the number of interferers increases to 2. This eventually goes down to 0, when the number of interferers increases to 6. However, Overcast provides near optimal performance for the VoIP client for up to 4 interferers, and falls only slightly as the number of interferers goes up to 9. This result illustrates the power of the Overcast approach. Even in high interference scenarios, Overcast delivers good quality-of-service by finely controlling transmissions using a centralized scheduler.

7.6.4 Scalability

We now turn our attention to the scalability properties of Overcast. We are interested in determining how many simultaneous VoIP clients Overcast can support. Because a key objective of Overcast is to support mobility for VoIP, we perform experiments by simultaneously moving multiple clients during an experimental run. There are a number of possible mobility scenarios that can be considered when conducting such experiments. However, note that moving clients along separate mobility paths does not stress the sys-

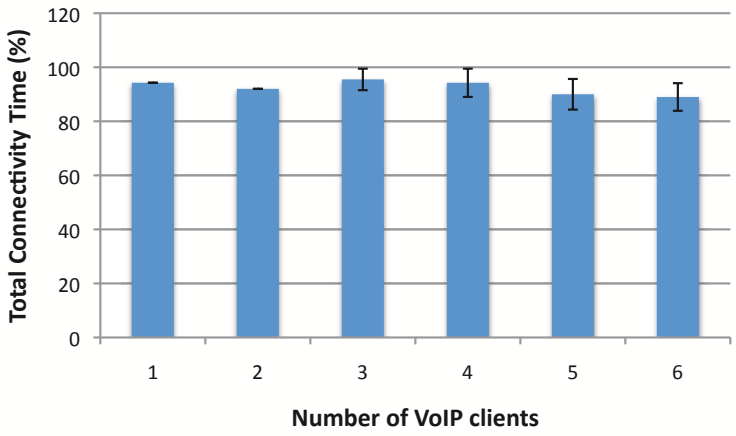

Figure 7.24: Total connectivity time for different numbers of VoIP clients

tem very much because VoIP traffic gets distributed across APs in different contention domains. Therefore, to stress test Overcast, we must perform experiments where we simultaneously (or at the same time) move multiple mobile clients along the same mobility path. For each client, we measure the mean packet reception rate during the mobility run and perform a total of 5 runs. We also plot the mean total connectivity time of the clients during these runs.

Figure 7.23 shows the mean packet reception rate for different numbers of clients. We see that Overcast is relatively insensitive to the number of clients and provides a mean packet reception rate of close to $50pps$ (although we observe some degradation for greater than four clients). While mean PRR is an aggregate statistic, Figure 7.24 shows the mean total connectivity time of the VoIP client, for different numbers of clients. To re-iterate, total connectivity time provides for us a way to bound the performance of the VoIP client and determine the amount of time the client operated above the baseline.

We observe that the total connectivity times are approximately the same for different numbers of clients. In these experiments, we used a variety of different hardware and software platforms for the clients, ranging from laptops running Linux, Windows XP and Vista, to iPhones running the OS X iPhone operating system. Therefore, our results are not an artifact of any particular platform used for the VoIP client[15]. In summary, our evaluation reveals that Overcast has good scalability properties even when multiple VoIP clients simultaneously walk along the same mobility path in the enterprise.

[15]Note that requiring no client modifications substantially eases switching between different client platforms as no configuration is necessary to allow them to interoperate with Overcast

7.7 Related Work

In this section, we discuss prior work on optimizing continuous mobility and providing realtime support for WLANs.

Continuous mobility has been studied in the context of minimizing hand-off latencies in wireless networks. Some prominent work, including [108] proposes neighbour graphs to reduce client scanning time, but requires offline computation of the graph, which is cumbersome and prone to inaccuracy. Ramani et al. [103] propose to synchronize beacon transmissions across neighbouring APs to reduce overall scanning time. However, this approach requires client modifications. In short, prior work attempts to minimize re-association overhead by reducing the scanning duration. In contrast, Overcast uses MAC address spoofing and a single virtual AP architecture to eliminate handoff delays altogether.

We now briefly discuss prior work on VoIP traffic support over 802.11 networks [58, 41, 69]. 802.11 networks are notorious for poorly supporting simultaneous VoIP connections [58]. Many proposals have been put forth to improve the dismal call capacity of WLANs [41, 69]. However, these require modifications to the clients MAC layer. Furthermore, most current approaches are designed to operate within a BSS. Recently, it was shown that multi-cell deployments support only 2 active sessions per AP on average [48]. This is a three times reduction compared to the single cell case, illustrating the poor support that existing multi-AP WLANs provide for realtime applications.

A recent paper proposes SoftSpeak [116], a distributed TDMA approach to supporting VoIP clients that both improves the number of simultaneous VoIP sessions as well as minimizes impact on Data traffic. However, SoftSpeak does not address handoffs and therefore cannot support continuous mobility. It also requires changes to 802.11 clients. These factors make it undesirable for the scenario we target in our work. Virtual PCF [71] has also recently been proposed to increase the number of VoIP users within a BSS, without requiring client modifications. In this scheme, an AP estimates when its VoIP clients will require access to the medium and uses CTS-to-self packets to reserve the medium for them. This approach is complementary to our work and can be integrated to provide uplink support for VoIP clients in Overcast.

We now move to works that propose architectures to support continuous mobility and VoIP. SMesh [38] proposes a system for fast, seamless handoffs in wireless mesh networks (WMNs). Each AP advertises a common gateway IP address and BSSID, avoiding DHCP overheads during handoff. However, SMesh requires clients to operate in ad hoc mode, which is not the default 802.11 client behavior. Another architec-

ture, DenseAP [94], uses dense AP deployments to improve performance in enterprise WLANs. Though interesting in principle, DenseAP is based on the multi-channel design and therefore incurs handoff delays (≈ 1.5 seconds), causing disruptions for realtime applications such as VoIP that have a delay budget of up to $200ms$. Furthermore, in congested scenarios, it does not prescribe any mechanism with which to manage interference in the neighborhood of the VoIP client. MDG [47] explores techniques such as channel assignment, power control and client association to improve enterprise network performance. However, MDG requires client modifications which makes it hard to deploy in practice. Trantor [96] was recently proposed as a clean-slate design to enterprise WLANs that also supports realtime applications. However, like other architectures, the benefits of Trantor are only realized with client modifications.

Some commercial vendors (e.g. Meru [8], Extricom [19]) also claim to support realtime traffic when clients are mobile. However, little is known about their solutions and there is no independent verification of their claims. Furthermore, our private discussions with one of them reveals that there are some fundamental differences between our approach and theirs. Finally, while these vendors use customized hardware for their solution, we develop Overcast on off-the-shelf commodity hardware that is deployed on the existing backbone infrastructure of our department's wired network.

7.8 Discussion

We now comment on the scope of Overcast. Overcast is designed to provide good QoS to legacy mobile VoIP clients in the enterprise. Having said that, there are a few points worth considering regarding the proposed approach:

- *Uplink Support:* While we focus on downlink interference (or conflicts) in Overcast, VoIP streams are bi-directional in nature and therefore require uplink support as well. To support uplink traffic, schemes such as the one proposed in [71] can be integrated into the Overcast system. We are currently investigating these techniques in greater detail.

- *Scheduling Overhead:* The current scheduling approach transmits a broadcast packet to synchronize APs at every time slot. While we have not observed any performance problems with this approach, it may become costly if the wired backbone is carrying a large amount of data traffic. Instead, after every few slot times, if we give the AP a schedule for the next few slots in a single broadcast packet, the overhead can be drastically reduced. On the other hand, errors due to clock

drift may occur between broadcast packets. Moreover, for a large number of slots (and long slot duration) the dynamics of the environment may also change (e.g. traffic patterns), which could result in wasted slot times. In practice, if the number of slots is small enough such that the environment can safely be assumed to be constant, we can get around the problem of network load and network dynamics.

- *Client Clock Synchronization:* While the virtual AP design is attractive since it removes the complexity of association from the client, there is a clock synchronization matter that could cause incorrect behavior at the client. Specifically, while each AP in the enterprise broadcasts the same BSSID and MAC address, the TSF timestamp is still unique to each AP. Every beacon received at the client causes it to update the hardware clock of its radio to reflect the newly received timestamp. Unfortunately, some 802.11 features that rely on accurate synchronization between the AP and the client (e.g. Power Save Mode) could experience problems as a result of this behavior. Addressing this matter is an interesting problem requiring further investigation.

- *Joint AP Selection and Scheduling:* In this work, we assumed that algorithms for AP selection and traffic scheduling operate independently of each other. While such an approach is feasible and provides gains, jointly deciding which AP (or path) to use for transmitting packets to a client and when they should be transmitted is another intriguing approach worth exploring.

- *Multi-Rate Support:* In our evaluation, we assumed that the link data rate for wireless transmissions is fixed. In real-world deployments, rate adaptation algorithms are commonly used to select the best rate based on current channel conditions. On our Intel platform, the rate adaptation algorithm was implemented in the firmware of the radio and would have required implementing an API to expose this information to the driver. However, the design of Overcast does not preclude multi-rate support for such traffic. In our current implementation, the AP estimates the number of packets it can transmit in a given slot, assuming a data rate of 6 Mbps. This can easily be replaced with a mechanism that dynamically picks a data rate based on the current mean signal strength observed from the client. This is similar to the 'rate-map' proposed in [94].

- *Network Security:* The single virtual AP design of Overcast presents some unique security challenges. For example the use of MAC address spoofing makes it harder to identify and isolate rogue APs that are deployed by malicious users trying to gain access to the corporate network. In these cases, using techniques such as WiFi-

based position estimation [49] may help to determine the positions of broadcasting APs and thereby identify devices that appear to be transmitting from unknown locations.

The objective of Overcast was to highlight the benefits of using *dynamic* conflict-graphs for fine-grained WLAN optimization. By exploiting such conflict graphs, accurate scheduling decisions can be made to provide consistent QoS to mobile clients. Apart from showcasing the usefulness of conflict graphs, this work contributes to the research literature in other ways as well. First, we show through detailed experimentation on both our wireless testbed and on a production WLAN, that the multi-channel design is poorly suited for delay-sensitive applications operating on legacy clients. Second, contrary to the prior literature that proposes sophisticated algorithms for AP selection [111, 94], in a single-channel design where hand-offs are network-controlled, simplistic AP selection algorithms work just as well as more sophisticated algorithms that account for multiple performance parameters when deciding the best AP for the client.

Chapter 8

Conclusions and Future Work

Modern wireless networks increasingly experience poor performance due to RF interference from devices operating on the same unlicensed frequencies. In the future, a growing user base and demand for high bandwidth applications will likely exacerbate interference in such networks. Enterprise WLANs are examples of 802.11 networks where user density and requirements for high throughput applications are common. Because these networks operate indoors, irregular RF propagation makes interference management an even greater challenge. To address these challenges, this dissertation takes a coordinated approach and proposes practical techniques for measuring and modeling RF interference in the form of conflict graphs. It applies these towards WLAN optimization problems to demonstrate significant gains in both network throughput and reliability. In this chapter, we conclude this dissertation by summarizing its main contributions, pointing out some of its limitations, and outlining remaining challenges for future work.

8.1 Contributions

The IEEE 802.11 standard was primarily designed for use in sparse network deployments with a few clients and APs. The MAC protocol in 802.11 was designed to provide distributed and fair access to the medium, and provide best-effort service to applications. Unfortunately, today's networks are characterized by dense deployments, heterogeneous traffic, and diverse usage patterns (from WiFi hotspots to long-distance WiFi networks). These characteristics violate the design principles upon which IEEE 802.11 was based, causing network performance to degrade. In what follows, we briefly list some of these key issues and subsequently describe how this dissertation fills in some crucial holes in the current design.

- The 802.11 design assumes interference to be an exception rather than the norm. As a result, simplistic techniques such as RTS-CTS and link-layer re-transmissions are proposed in the protocol to alleviate interference when it arises. In dense deployments with many APs and clients, interference is no longer an exception, and decentralized techniques such as those described above fail to address the interference problem.

- Management and coordination, using conflict graphs have been proposed for alleviating interference in dense 802.11 networks. However, existing techniques for building these conflict graphs are either too inaccurate or prohibitively expense for use in real-world networks.

- 802.11 networks are becoming increasingly dynamic, with factors such as obstacle movement and client mobility increasingly affecting network performance. As a result, interference patterns also rapidly change in such environments. Prior interference estimation techniques are ineffective in these scenarios because they assume static clients where links are stationary for the duration in which the clients are connected to the network.

- Existing interference estimation techniques assume complete administrative control of the network (APs and clients). However, clients in an enterprise WLAN operate independently of the network infrastructure and modifying them limits widespread applicability of the proposed techniques.

- Existing WLAN optimization algorithms operate on timescales of minutes to hours because interference information is only available at these timescales. Thus, fine-grained WLAN optimization is not possible with existing interference estimation techniques.

In the context of enterprise WLANs, this dissertation addresses the above problems in the following way:

- **Design:** It proposes a centralized enterprise WLAN architecture (dubbed 'SMARTA') that uses conflict graphs for WLAN optimization. SMARTA introduces a conflict graph construction framework that is not based on any RF propagation model and only requires modest modifications to the networking infrastructure. It is also the first architecture that caters to the unique requirements of enterprise WLANs (e.g. no client modifications, online interference estimation) and develops techniques that are easily deployable in existing WLAN systems. The gains from using the

SMARTA approach are illustrated by applying it to problems of centralized frequency selection and power control.

- **Implementation:**
  - With the goal of practically evaluating the proposed interference measurement framework, this dissertation also provides insights into designing enterprise WLAN testbeds for centralized control. It highlights the key factors and practical constraints that must be met when designing such enterprise WLAN architectures.
  - This dissertation implements the conflict graph construction framework for SMARTA (dubbed 'Micro-Probing') to demonstrate its practical application to real-world deployments. In doing so, it applies novel techniques such as MAC Service Time to discover carrier-sensing interference and CTS-to-self to silence the network. Techniques such as silencing not only facilitate conflict graph construction but also serve as generic tools that can be applied to other problems as well [71].
- **Application:** This dissertation applies the interference measurement system to the problem of supporting mobile VoIP clients in the enterprise. The proposed system (dubbed 'Overcast') uses the conflict graph to decide the best path (i.e. AP selection) and time (i.e. traffic scheduling) in which to transmit packets to each VoIP client. The resulting system is able to provide reliable performance to VoIP clients even in the presence of co-located backlogged interferers.

In addition, this dissertation makes the following key contributions:

- **Highlights the key requirements for conflict graph modeling and construction in enterprise WLANs**. Enterprise WLANs require a technique that *rapidly* discovers interference in an *online* network. Furthermore, to be widely deployable, the technique must require no modifications to wireless clients.
- **Proposes the first measurement approach that is able to discover interference in an online network.** The approach leads to a three orders of magnitude reduction in measurement time without sacrificing measurement accuracy. Furthermore, it is even able to capture interference in cases where the receiver is out of communication range of the interferer.
- **Opens up the space for new innovations in WLAN optimization**, because of its ability to measure the conflict graph at dramatically smaller timescales. With the

ability to measure conflict graphs quickly, optimization algorithms can account for changes caused by client mobility that were difficult to capture using traditional measurement techniques.

8.2 Limitations

In this section, we outline a few limitations of our work. We divide them into two categories. First are those that are fundamental to our proposed approach and represent design trade-offs. The second (discussed in Section 8.3) are open problems that are not necessarily limitations of our approach but enhancements to our system that we did not pursue in this dissertation. We discuss each of them in turn.

8.2.1 Lack of Client Control

In this dissertation, we focused on designing a system that allows: 1) Easy deployment into existing enterprise WLANs, and 2) Supports legacy clients that do not report any state information to the AP. While this greatly eases the deployment process, we recognize that such a design may likely be sub-optimal with respect to one which also uses client feedback (as discussed in [96]). However, note that our design does not preclude the ability to obtain feedback from the client and can be incorporated into our system, should that become feasible.

We therefore propose the following deployment path. Our current system can initially be deployed into existing enterprise WLAN infrastructures. Then, as clients are upgraded to the latest standards (e.g. IEEE 802.11k), the system can be modified to incorporate additional state information provided by these upgraded clients.

8.2.2 Use of Commodity Platforms

Commodity platforms are restrictive in terms of their functionality and flexibility, when compared to platforms such as software-defined radios (SDRs). In our work, we were able to gain access to the driver and firmware for a commodity radio, but were still not able to collect all possible metrics of interest. For instance, we were not able to measure energy spikes that could potentially be used to detect interference at the receiver (as discussed in [117]). Our choice of commodity platforms represents a tradeoff between ease of deployment and better interference measurement accuracy.

8.2.3 Non-Enterprise Interference

This dissertation focuses on interference arising from APs (and clients) that are part of the enterprise WLAN. Conflicts (or interference) from non-802.11 devices (e.g. microwave ovens) or devices that are part of a different enterprise network are not captured. Non-802.11 devices are difficult to detect using commodity radios that do not expose any information about non-802.11 RF signals. Furthermore, there is no way to control 802.11 devices that are under a different administrative domain[1]. However, as standards such as IEEE 802.11v are ratified, the ability to extend the interference measurement framework across different administrative domains shall greatly be eased.

8.3 Future Work

This dissertation lays the groundwork for an exciting set of research problems along multiple axes. We describe some of these research problems next.

8.3.1 Extending Conflict Graphs

The existing conflict graph representation and construction framework can be extended in various ways, as described below.

Upstream Conflicts: The current implementation of the interference measurement system supports detection of downlink conflicts, i.e. those arising from the APs. However, emerging realtime applications are bound to increase uplink traffic as well. Therefore, in the future, measuring uplink conflicts would also be necessary to ensure that interference is handled bidirectionally. While we have outlined some tests for uplink conflicts in Chapter 4, an implementation of such an approach is required.

Multi-Interferer Conflicts: In this dissertation, we focused on first-order conflicts, i.e. conflict between pairs of links. Second or third order conflicts are possible where multiple nodes combine to cause conflict on a link. Capturing the affect of these conflicts is possible by applying the ideas developed in [97] on the pairwise conflict graph computed using micro-probing. While this accounts for most higher order interference effects, it misses scenarios where pairwise interference is not observed between links, yet the combined interference from multiple interferers causes conflict on a link [54].

[1]There may be ways to passively sniff traffic to estimate interference, but the accuracy of such techniques is lower than the active measurements framework proposed in this dissertation

Extending the interference measurement system to handle these cases is an interesting area of future work.

Coordinated Silencing: An important design element of micro-probing is the use of silencing (i.e. unsolicited CTSes) to correctly execute the interference tests. While we were able to silence the medium in most cases, silencing may fail if devices use asymmetric powers levels across the enterprise. To alleviate this problem, multiple APs could perform silencing for a test and create a boundary around the link pair to be tested.

Scalability: In our work, we tested the interference measurement system on a 38 node wireless testbed. Today, enterprises can support up to 10,000 APs and many more clients [61]. In this case, a single controller may not be sufficient, requiring the use of a hierarchy of controllers where region-specific controllers manage individual regions of the deployment and compute the conflict graph for their own regions. These conflict graphs would then be combined at a master controller to generate the aggregate conflict graph for the entire network. Supporting conflict graphs for such deployments is an interesting area of future work.

Scheduling Interference Tests: In our work, we designed a set of tests that accurately and rapidly measure interference in enterprise WLANs. However, the manner in which these tests are scheduled in an online network is not covered. Instead, we periodically re-measure interference between links. While we did not encounter any performance problems with using micro-probing in conjunction with Overcast, with increasing traffic volume, designing intelligent scheduling strategies for the interference tests will become important.

Reducing Number of Interference Tests: So far, our work has focused on reducing measurement overhead per-interference test. However, at larger scales, the number of measurements could potentially become the bottleneck. Therefore, reducing the number of measurements will become important and is worth exploring for micro-probing as well.

Modeling Impact of Interference: In this dissertation, we use a simple linear model to capture the impact of interference, which is a function of the load of the interfering source (measured using packet level statistics that are readily available in the driver of most wireless cards). For a fixed data transmission rate, this modeling turns out to be accurate for 802.11 networks [97]. However, this model will likely not hold in cases where the network supports multiple transmission rates. For instance, a client with fewer packets to send at a low data rate could actually cause more interference than one with many packets to send at a higher rate. A better metric is the mean time that the client occupies the channel (i.e. the number of busy slots). This information is typically only

available in the firmware of most commodity cards.

Decentralized Construction of Conflict Graphs: We focused on addressing the problem of generating conflict graphs for enterprise WLANs. However, to measure the conflict graph for other wireless systems (e.g., wireless mesh networks (WMNs)), we require a distributed implementation. While the specific techniques proposed in this dissertation are not directly applicable to WMNs, the underlying principles to support interference measurements (i.e. synchronized probing, network silencing, etc) are common to both applications. Developing the interference measurement framework for a WMN is an exciting direction of future work.

8.3.2 WLAN Optimization Algorithms

Joint Parameter Optimization: Most WLAN optimization schemes tune AP parameters independently of each other [47]. In our work, we followed the same methodology. For SMARTA, we explored channel selection and power control independently. For Overcast, we performed AP selection and centralized scheduling independently. It would be interesting to explore algorithms that jointly optimize parameters for each of these systems.

Centralized Data Rate Adaptation: In today's WiFi networks, each transmitting node is independently responsible for deciding the best data rate to use, based on the observed signal quality to the receiver. Centralization of data rate adaptation, while proposed in prior work [96], has not been explored in depth. In such a scheme, conflict graphs could be used to select an appropriate rate. For links that have many potential conflicts, choosing rates that are more robust to collisions could improve performance for those links. Furthermore, choosing higher data rates that reduce the air-time per transmission could help alleviate the impact of exposed terminal interference between pairwise links.

Comprehensive Power Control: In Chapter 4, we presented a power control algorithm that only considered inter-AP conflicts when selecting transmission powers for the APs. A more comprehensive scheme would consider client conflicts as well. Designing a power control algorithm that accounts for all possible conflicts could improve performance over the weighted Iteration Reduction (wIR) algorithm proposed for SMARTA.

8.3.3 Studying Properties of Conflict Graphs

Being able to rapidly measure conflict graphs for 802.11 networks allows us to more closely examine the structural properties of these graphs. Understanding these properties can potentially enable the design of better optimization algorithms that further improve network performance. Below we describe two possible avenues in this space.

Graph-theoretic Properties of Conflict Graphs: While computing conflict graphs has been well-studied, exploring the graph theoretic aspects of such graphs has received less attention. In particular, determining the graph family (for example, interval graphs) to which most conflict graphs belong is useful as some graph problems (e.g. graph colouring) are easier on certain types of graphs. Consequently, this could lead to innovative algorithms that exploit such information to improve network performance.

Time-Space Properties of Conflict Graphs: A key aspect missing in prior work is a study on how the structure of the conflict graph evolves across time and space. Moreover, little is known about the impact of various tuning parameters on the shape of the conflict graph. Some of our preliminary work reveals similarities between different parameter configurations of an 802.11 radio [31], thereby allowing us to reduce the search space of possible configurations to test. However, a comprehensive study on the evolution of conflict graphs is required.

8.4 Concluding Remarks

This dissertation has a few underlying themes. First, it attacks an important problem in wireless systems that not only affects networks today but is expected to have an impact on future wireless networks as well. While a large body of prior work focuses on piecemeal solutions, this dissertation seeks to systematically address RF interference through the use of conflict graphs that globally model interference. Second, it develops solutions based on practical assumptions that allow for the easier integration of conflict graphs into existing wireless networking designs. This aspect is often ignored in academic research when designing and prototyping wireless systems. Third, it focuses primarily on practical implementation rather than theory. While theory provides us some intuition, practical implementation forces us to address real-world constraints. Finally, this dissertation adopts an evaluation methodology that involves experimenting on large-scale wireless testbeds. This is crucial as it allows researchers to test scalability aspects of the proposed approach. Put together, these themes make for a scientific method that not only allows for sound research contribution, but also real-world application of proposed

solutions that can be immediately deployed into existing systems.

Appendix A

Copyright Information

1. N. Ahmed and S. Keshav. SMARTA: A self-managing architecture for thin access points. In Proceedings of ACM CoNEXT, 2006

2. N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of RF interference. In Proceedings of ACM CoNEXT, 2008

Bibliography

[1] Aruba Networks: Advanced RF Management for Wireless Grids. http://www.arubanetworks.com/pdf/rf-for-grids.pdf. 24, 26, 33, 35, 36, 75, 89, 91, 99, 118, 119

[2] Belkin Wireless G Router. http://www.belkin.com/. 77

[3] Cisco Wireless Control System. www.cisco.com/en/US/products/ps6305/. 118

[4] Demands on Today's Data Communications Technologies. http://www.informit.com/articles/article.aspx?p=29249&seqNum=2. 85

[5] Engim Delivers Simultaneous Multi-Channel WLAN Switching Engine Silicon. http://www.tmcnet.com/enews/041103j.htm. 36

[6] ISO/IEC 11172-2:1993 Standard. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22411. 85

[7] Linksys WRT54G Router. http://www.linksys.com/. 77

[8] Meru Networks. http://www.merunetworks.com. 24, 26, 35, 36, 75, 120, 123, 145

[9] Network Time Protocol. http://www.ntp.org. 96

[10] OpenWrt – Wireless Freedom. http://openwrt.org/. 77

[11] Propagate Inc., AutoCell - The Self-Organizing WLAN. http://www.propagatenet.com/resources/index.html. 36

[12] Quick guide to IEEE 802.11 WG activities. http://tinyurl.com/ypojvc. 19

[13] Scalable Networks Inc., QualNet Simulator version 4.0. http://www.scalable-networks.com/. 5, 21, 44, 58, 74

[14] Soekris Engineering. http://www.soekris.com. 77

[15] VIA Technologies. http://www.via.com.tw. 77, 94

[16] Wi-Fi Alliance. http://www.wi-fi.org/pressroom_overview.php?newsid=770. 1

[17] WLAN Faceoff: Cisco Versus Meru. http://www.unstrung.com/document.asp?doc_id=111677&print=true. 35

[18] D-Link Wireless 108G MIMO Router (DI-634M), 2005. http://www.dlink.com/products/?pid=458. 17

[19] Extricom Inc., Wireless LAN Switch DataSheet, 2005. http://www.extricom.com/imgs/Uploads/PDF/SWDataSheet.pdf. 24, 26, 35, 36, 75, 120, 145

[20] Trapeze Networks Inc., RingMaster Datasheet, 2005. http://www.trapezenetworks.com/products/datasheets/\rm/rm.asp. 36

[21] Wireless-G Access Point with SRX, 2005. http://www.linksys.com/. 17

[22] Xirrus Inc, WLAN Array Architecture. 2005. http://www.xirrus.com/public/pdf/Xirrus_WLAN_Array_Architecture.pdf. 24, 26, 35, 36, 75

[23] Control And Provisioning of Wireless Access Points (CAPWAP). http://www.ietf.org/dyn/wg/charter/capwap-charter.html, August 2008. 35

[24] IEEE P802.11 - TASK GROUP k. http://grouper.ieee.org/groups/802/11/Reports/tgk_update.htm, June 2008. 34

[25] IEEE P802.11 - TASK GROUP V. http://grouper.ieee.org/groups/802/11/Reports/tgv_update.htm, July 2009. 35

[26] The ns Manual, 2009. http://www.isi.edu/nsnam/ns/ns-documentation.html. 61

[27] N. Ahmed. A Self-Management Approach to Configuring Wireless Infrastructure Networks, Master's Thesis, University of Waterloo (UW). 2006. `http://etd.uwaterloo.ca/etd/n3ahmed2006.pdf`. 51, 53, 54, 58, 65

[28] N. Ahmed, A. Allavena, and S. Keshav. A Mathematical Model for Optimal Coverage Planning in Wireless LANs. 2006. *Unpublished Manuscript*. 72

[29] N. Ahmed and U. Ismail. Designing a High Performance Wireless Testbed for Centralized Control. *In Proceedings of the 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom)*, 2009. 8

[30] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online Estimation of RF Interference. In *Proceedings of the 4th ACM International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT)*, 2008. 8, 87, 128

[31] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Abstract: Measuring Multi-Parameter Conflict Graphs for 802.11 networks. *In Mobile Computer Communications Review (MC2R)*, 2009. 155

[32] N. Ahmed and S. Keshav. SMARTA: A Self-Managing Architecture for Thin Access Points. In *Proceedings of the 2nd ACM International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT)*, 2006. 9, 37

[33] N. Ahmed and S. Keshav. Personal discussions with IT Administrators at the University of Waterloo. 2008. 39

[34] N. Ahmed and S. Keshav. Personal discussions with Technical Managers at Aruba Networks. 2008. 75

[35] N. Ahmed, V. Shrivastava, A. Mishra, S. Banerjee, S. Keshav, and K. Papagiannaki. Interference mitigation in enterprise WLANs through speculative scheduling. In *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2007. 75

[36] A. Akella, G. Judd, S. Seshan, and P. 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. 64, 72

[37] D. Akin. *Certified Wireless Network Administrator (CWNA) Official Study Guide (Exam PW0-100)*. McGrawHill/Osborne, Berkeley, California, 2003. 18, 25

[38] Y. Amir, C. Danilov, M. Hilsdale, R. Musăloiu-Elefteri, and N. Rivera. Fast handoff for seamless wireless mesh networks. In *Proceedings of the 4th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2006. 144

[39] P. Bahl, M. Hajiaghayi, K. Jain, V. Mirrokni, L. Qiu, and A. Saberi. Cell breathing in wireless lans: Algorithms and evaluation. *IEEE Transactions on Mobile Computing*, 2006. 72

[40] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan. Interactive wifi connectivity for moving vehicles. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 427–438, New York, NY, USA, 2008. ACM. 140

[41] R. O. Baldwin, I. Nathaniel J. Davis, S. F. Midkiff, and R. A. Raines. Packetized voice transmission using RT-MAC, a wireless real-time medium access control protocol. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):11–25, 2001. 144

[42] Y. Bejerano and R. Bhatia. MiFi: A Framework for Fairness and QoS Assurance in Current IEEE 802.11 Networks with Multiple Access Points. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1229–1240, 2004. 29

[43] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LAN's. In *Proceedings of the conference on Communications architectures, protocols and applications (SIGCOMM)*, pages 212–225, 1994. 17

[44] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005. 75

[45] R. Borndorfer, A. Eisenblatter, M. Grotschel, and A. Martin. Frequency assignment in cellular phone networks. In *Annals of Operations Research*, volume 76, pages 73–93, 1998. 72

[46] I. Broustis, J. Eriksson, S. Krishnamurthy, and M. Faloutsos. A Blueprint for a Manageable and Affordable Wireless Testbed: Design, Pitfalls and Lessons

Learned. *In Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom)*, 2007. 75, 76, 79, 83

[47] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre. MDG: measurement-driven guidelines for 802.11 WLAN design. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom)*, pages 254–265, New York, NY, USA, 2007. ACM. 33, 73, 145, 154

[48] A. Chan and S. Liew. VoIP Capacity over Multiple IEEE 802.11 WLANs. In *Proceedings of the IEEE International Conference on Communications (OCC)*, 2007. 115, 144

[49] R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-based Management System for Enterprise Wireless LANs. In *Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation (NSDI)*, 2007. 147

[50] H. Chang and V. Misra. 802.11 Link Interference: A Simple Model and A Performance Enhancement. In *Proceedings of the 4th International IFIP-TC6 Networking Conference (NETWORKING 2005)*, pages 1330 – 1333, May 2005. 44, 72

[51] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. *SIGCOMM Computer Communication Review (CCR)*, 2006. 75, 91, 96, 98, 112

[52] R. G. Cole and J. H. Rosenbluth. Voice over ip performance monitoring. *SIGCOMM Comput. Commun. Rev.*, 31(2):9–24, 2001. 138

[53] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom)*, pages 134–146, New York, NY, USA, 2003. ACM. 105, 106

[54] S. M. Das, D. Koutsonikolas, Y. C. Hu, and D. Peroulis. Characterizing multi-way interference in wireless mesh networks. In *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization (WiNTECH)*, pages 57–64, New York, NY, USA, 2006. ACM. 92, 152

[55] Y. Ding, Y. Huang, G. Zeng, and L. Xiao. Channel assignment with partially overlapping channels in wireless mesh networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet (WICON)*, pages 1–9, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 21

[56] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *Proceedings of the Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*, 2004. 77

[57] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Operating Systems Review (OSR)*, 36(SI):147–163, 2002. 50, 96

[58] S. Garg and M. Kappes. An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks. In *Proceedings of the IEEE Wireless Communications & Networking Conference (WCNC)*, 2003. 114, 144

[59] J. Geier. Assigning 802.11b access point channels. 2002. `http://www.wi-fiplanet.com/tutorials/article.php/972261`. 61

[60] A. Glassner. *An Introduction to Ray Tracing*. San Diego, California, 1989. Academic Press. 25

[61] J. Goldman. Aruba WLAN at Microsoft Exceeds 11,000 Access Points. 2008. http://www.wi-fiplanet.com/news/article.php/3753466. 153

[62] S. Gollakota and D. Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. *SIGCOMM Computer Communication Review (CCR)*, 38(4):159–170, 2008. 32

[63] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 networks. In *Proceedings of the International Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, New York, NY, USA, 2007. ACM Press. 19

[64] D. Hadaller, S. Keshav, and T. Brecht. MV-MAX: improving wireless infrastructure access for multi-vehicular communication. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks (CHANTS)*, pages 269–276, New York, NY, USA, 2006. ACM Press. 20

[65] M. M. Halldorson, J. Y. Halpern, L. E. Li, and V. S. Mirrokni. On spectrum sharing games. In *Proceedings of the 23rd annual ACM symposium on Principles of distributed computing (PODC)*, pages 107–114, St. John's, Newfoundland, Canada, 2004. 48

[66] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 339–350, New York, NY, USA, 2008. ACM. 33

[67] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. 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. 19

[68] A. Hills and B. Friday. Radio resource management in wireless LANs. *IEEE Communications Magazine*, 42(12):9–14, 2004. 26

[69] T. Hiraguri, T. Ichikawa, M. Iizuka, and M. Morikura. Novel multiple access protocol for voice over IP in wireless LAN. In *IEEE International Symposium on Computers and Communications*, 2002. 144

[70] S. hwan Yoo, J.-H. Choi, J.-H. Hwang, and C. Yoo. Eliminating the performance anomaly of 802.11b. 2005. In Lecture Notes in Computer Science, Vol. 3421. Springer: Berlin, 1055. 19

[71] U. Ismail. Virtual PCF: Improving VoIP Capacity in WLANs with Legacy Clients, Master's Thesis, University of Waterloo (UW). 2009. 8, 112, 123, 130, 144, 145, 150

[72] K. Jain, J. Padhye, V. N. Padmanabhan, and L. 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. 21, 72

[73] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth. Iqu: practical queue-based user association management for wlans. In *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom)*, pages 158–169, New York, NY, USA, 2006. ACM Press. 2

[74] T. Karhima, A. Silvennoinen, M. Hall, and S. Haggman. IEEE 802.11b/g WLAN tolerance to jamming. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2004. 18

[75] P. Karn. MACA - A New Channel Access Method for Packet Radio. In *Proceedings of the 9th ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990. 17

[76] A. Kashyap, S. Ganguly, and S. R. Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom)*, pages 242–253, New York, NY, USA, 2007. ACM. 22, 91

[77] B. Kauffman, F. Bacelli, A. Chaintreau, K. Papagiannaki, and C. Diot. Self-Organization of Interfering 802.11 Wireless Access Networks. Technical Report 5649, INRIA, 2005. 72

[78] V. Kawadia and P. R. Kumar. Principles and protocols for power control in ad hoc networks. *Special Issue on Ad Hoc Networks*, 1, 2005. 72

[79] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental Evaluation of Wireless Simulation Assumptions. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 78–82. ACM Press, October 2004. 5, 74, 87

[80] J. Kristiansson and P. Parnes. Application-layer mobility support for streaming real-time media. *Proceedings of the IEEE Wireless Communications & Networking Conference (WCNC)*, 1:268–273 Vol.1, March 2004. 114

[81] J. Lee, S.-J. Lee, W. Kim, D. Jo, T. Kwon, and Y. Choi. RSS-based Carrier Sensing and Interference Estimation in 802.11 Wireless Networks. In *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007. 91

[82] X. Liu, S. Seshan, and P. Steenkiste. A Practical Interference-Aware Power Management Protocol for Dense Wireless Networks. 2007. Unpublished Manuscript. 32

[83] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas. In *Proceedings of the Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*, 2009. 23, 32

[84] H. Lundgren, K. Ramachandran, E. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh. Experiences from the design, deployment, and usage of the UCSB MeshNet testbed. *IEEE Transactions on Wireless Communications*, 13(2):18–29, April 2006. 77

[85] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. *SIGCOMM Computer Commincation Review (CCR)*, 36(4):75–86, 2006. 91

[86] I. Mas, H. Velayos, and G. Karlsson. Distributed admission control for wireless LANs. In *Winternet Grand Finale Workshop*. KTH, Royal Institute of Technology, Stockholm Sweden, 2005. 112

[87] V. Mhatre, K. Papagiannaki, and F. Baccelli. Interference Mitigation Through Power Control in High Density 802.11 WLANs. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Anchorage, Alaska, USA, 2007. 33, 34

[88] V. P. Mhatre and K. Papagiannaki. Optimal design of high density 802.11 WLANs. In *Proceedings of the 2nd ACM International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT)*, pages 1–12, New York, NY, USA, 2006. ACM. 30

[89] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted Coloring based Channel Assignment for WLANs. *In Mobile Computer Communications Review (MC2R)*, 9(3), 2005. 21, 31

[90] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client Driven Approach for Channel Management in Wireless LANs. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, 2006. 31, 38, 42, 48, 49, 55, 72

[91] A. Mishra, S. Rayanchu, D. Agrawal, and S. Banerjee. Supporting continuous mobility through multi-rate wireless packetization. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications (HotMobile)*, pages 33–37, New York, NY, USA, 2008. ACM. 114

[92] A. Miu, G. Tan, H. Balakrishnan, and J. 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. 17

[93] A. Munaretto, M. Fonseca, K. Agha, and G. Pujolle. Fair Time Sharing Protocol: A Solution for IEEE 802.11b Hot Spots. 2004. In Lecture Notes in Computer Science, 3124:1261-1266. 19

[94] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing high performance enterprise Wi-Fi networks. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008. 34, 113, 115, 131, 145, 146, 147

[95] R. Murty, J. Padhye, A. Wolman, and M. Welsh. Dyson: An Architecture for Extensible Wireless LANs. In *SIGCOMM Poster Session*, 2009. 32

[96] R. Murty, A. Wolman, J. Padhye, and M. Welsh. An Architecture for Extensible Wireless LANs. In *Proceedings of thw 7th ACM Workshop on Hot Topics in Networks (HotNets-VII)*, 2008. 31, 145, 151, 154

[97] D. Niculescu. Interference map for 802.11 networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC)*, pages 339–350, New York, NY, USA, 2007. ACM. 25, 38, 46, 92, 105, 152, 153

[98] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2005. 5, 21, 23, 38, 72, 88, 90, 103, 109

[99] J. Park, D. kim, C. Kang, and D. Hong. Effect of partial band jamming on ofdm-based wlan in 802.11g. In *ICASSP*, 2003. 18

[100] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Fault Detection, Isolation, and Diagnosis in Multi-hop Wireless Networks. Technical Report TR-2004-11, Microsoft, 2003. 72

[101] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom)*, pages 171–182, New York, NY, USA, 2007. ACM. 22, 91

[102] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser. Symphony: synchronous two-phase rate and power control in 802.11 wlans. In *Proceeding of the 6th International Conference on Mobile systems, applications, and services (MobiSys)*, pages 132–145, New York, NY, USA, 2008. ACM. 32

[103] I. Ramani and S. Savage. SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005. 144

[104] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Pearson Education, 2002. 16, 29

[105] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. *SIGCOMM Computer Communication Review (CCR)*, 36(4):51–62, 2006. 22, 38, 91

[106] A. Seth, M. Zaharia, S. Bhattacharya, and S. Keshav. Policy Oriented Architecture for Opportunistic Communication on Heterogeneous Wireless Networks. 2006. Unpublished Manuscript. 1

[107] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker. MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs. In *Proceedings of the 4th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Uppsala, Sweden, 2006. 72

[108] M. Shin, A. Mishra, and W. A. Arbaugh. Improving the Latency of 802.11 handoffs using Neighbor Graphs. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004. 144

[109] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra. CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2009. 5, 27, 47, 88, 129

[110] K. Solie and L. Lynch. *CCIE Practical Studies, Volume II (CCIE Self-Study)*. Cisco Press, 2004. 85

[111] K. Sundaresan and K. Papagiannaki. The need for cross-layer information in access point selection algorithms. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC)*, 2006. 126, 130, 147

[112] G. Tan and J. Guttag. Time-based fairness improves performance in multi-rate WLANs. In *Proceedings of the USENIX Annual Technical Conference 2004 on USENIX Annual Technical Conference (ATEC)*, pages 23–23, Berkeley, CA, USA, 2004. USENIX Association. 19

[113] P. Thornycroft. The all-wireless workplace is now open for business: Using 802.11n as your primary network LAN deployments. 2007. Technical Report, Aruba Networks. 114

[114] Tristan Henderson and David Kotz and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. *Computer Networks*, 52(14):2690 – 2712, 2008. 75, 85

[115] A. Vasan, R. Ramjee, and T. Woo. ECHOS - Enhanced capacity 802.11 hotspots. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1–9, San Diego, CA, USA, 2005. 30

[116] P. Verkaik, Y. Agarwal, R. Gupta, and A. Snoeren. SoftSpeak: Making VoIP Play Fair in Existing 802.11 Deployments. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009. 114, 144

[117] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *Proceedings of the Annual Conference of the Special Interest Group on Data Communications (SIGCOMM)*, 2009. 151

[118] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing exposed terminals in wireless networks. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 59–72, Berkeley, CA, USA, 2008. USENIX Association. 92

[119] D. Wetherall. Management Chaotic Wireless Networks. Talk Delivered at Microsoft Research, Self-Managing Networks Summit, 2005. 26

[120] 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 *Proceedings of the 2nd European Personal and Mobile Communications Conference (EPMCC)*, pages 26–36, Bonn, Germany, 1997. 25

[121] X. Yang and N. H. Vaidya. Priority scheduling in wireless ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 71–79, New York, NY, USA, 2002. ACM. 4, 20

[122] J. Zhu, B. Metzler, X. Guo, and Y. Liu. Adaptive CSMA for Scalable Network Capacity in High-Density WLAN: a Hardware Prototyping Approach. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, 2006. 18, 23, 31