OPTIMIZING ENTERPRISE WIRELESS NETWORKS THROUGH CENTRALIZATION

by

Vivek Vishal Shrivastava

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2010

To my parents, Bijay Kumar Shrivastava and Rani Shrivastava.

ACKNOWLEDGMENTS

I arrived at the University of Wisconsin-Madison six years ago, intending to get a masters. However, six memorable years later, I am now finishing up my doctoral dissertation and I am very thankful for one of the biggest, and most importantly, best decisions of my life. And the credit must go to my advisor, Professor Suman Banerjee, whose guidance and unwavering support has been instrumental in my successfully reaching this juncture. His support and belief in my work has steered me through some of the most challenging phases of my PhD pursuit. Above all, he made himself available for all my problems, both professional and personal, and helped me look at the bigger picture of my work. Thanks, Suman!

Similarly, I am indebted to Dr. Dina Papagiannaki, my mentor at Intel Research, who was a source of constant encouragement and has been like a second advisor to me. Her tireless work ethic was a source of inspiration for me and her advice has helped shape a lot of my work. She is a world class researcher and I am fortunate to have the opportunity to work with her. Thanks, Dina!

Professor S. Keshav, my collaborator at the University of Waterloo, in addition to contributing to my work, has provided valuable feedback and support. His advice on publishing has been a key cornerstone for my research efforts. Thanks, Keshav! I would also like to thank my defense committee, Professor Paul Barford, Professor Aditya Akella, Dr. Dina Papagiannaki, Professor Shan Lu and Professor Stark Draper for their valuable advice and flexibility in facilitating my defense.

My research has been facilitated by some excellent colleagues and friends, whose constant support and encouragement helped me accomplish my goals in a timely manner. I would like to thank Arunesh Mishra, for mentoring me in the initial stages of my research and introducing me to the joys of experimental work. My initial experiences of working with Arunesh helped

shape my thinking and impacted my approach to solving practical research problems. Thanks, Arunesh! Likewise, I am indebted to my colleague and close friend Shravan Rayanchu for the long discussions and late night efforts, which helped refine my initial ideas into their final form. He was always available for me to bounce off my raw ideas and he helped me navigate many roadblocks that I hit in my research career so far. Thanks, Shravan!

Setting up the enterprise scale wireless testbed in the department was a big part of my research infrastructure and I am grateful to my dear colleague and friend Nabeel Ahmed for helping me set up the testbed. Nabeel's passion for his work is contagious and we have experienced the pains and eventual joy of successfully setting up big infrastructure, which eventually served as the bedrock of our experiments. Thanks, Nabeel!

I am fortunate to have been at University of Wisconsin-Madison for six years. During my stay here, I have interacted with and received valuable feedback from many fellow students. Sayandeep Sen, Jongwon Yoon, Ashok Anand, Asim Kadav, Dheeraj Agrawal, Sharad Saha, Cindy Rubio, Jayaram Bobba and Siddarth Barman deserve credit for their help and support at different points in my PhD. Thanks to Neil Klingensmith and Vishnu Katreddy for making the last few months so exciting and enjoyable.

PhD requires sustained efforts for a long period of time and I wish to acknowledge my close group of friends and family who helped me navigate many difficult situations in the past six years. I am especially indebted to Nipun, Gaurav, Sharad, Awlok, Vikram, Mithilesh and Geetika, who have supported me whole heartedly in some of my most difficult times, ever since my undergraduate days right through my graduate years. I would also like to thank Namita, Ranga, Rajeev and Emma for their love and support, which made my stay in Madison enjoyable and fun-filled.

Special thanks to Nidhi, who provided the much needed support and encouragement, especially during the closing stages of my PhD. Her knack for cheering me up, even in the most stressful of times, has been priceless!

I will be forever indebted to my parents, whose sacrifices made it possible for me to pursue my dreams. I am blessed to have such understanding and caring parents. Thanks, Mom! Thanks,

Dad! My brother, Rohit, has supported me through all my tough times while taking care of his own academics. Thanks, Rohit!

I would like to extend a special thanks to all my other friends who I might not have mentioned here. I am grateful for having known you and thanks for all your help and support at any point in my life.

So long Madison, it has been a special ride !

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

Appendix

Page

Appendix

Page

DISCARD THIS PAGE

LIST OF TABLES

Table Page

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

DISCARD THIS PAGE

NOMENCLATURE

IEEE	Institute of Electrical and Electronics Engineers
802.11	Standard for physical and mac layer mechanisms as defined by IEEE
WLAN	Wireless Local Area Network
HT	Hidden Terminal links
ET	Exposed Terminal links
DCF	Distributed Coordination Function, the default channel access for 802.11
DET	A downlink deterministic scheduler
SPEC	A downlink speculative scheduler
CENTAUR	A downlink hybrid scheduler, that combines DET with DCF
PIE	A passive interference estimation tool

OPTIMIZING ENTERPRISE WIRELESS NETWORKS THROUGH CENTRALIZATION

Vivek Vishal Shrivastava

Under the supervision of Associate Professor Suman Banerjee

At the University of Wisconsin-Madison

As the number of wireless devices continues to grow in offices and enterprise environments, Wireless Local Area Networks (WLANs) have emerged as an important part of an enterprise network. Most enterprise WLANs tend to have a centralized architecture, which facilitates management and better design of various control and security functions. In spite of significant progress made in planning, deploying, and managing such enterprise WLANs, radio interference remains a core concern among WLAN users, network administrators, and operations staff alike.

This dissertation explores the design and implementation of data and control plane mechanisms that leverage the centralized architecture of enterprise WLAN's to manage interference effectively in such environments. We first explore the design of a centralized data plane for enterprise WLANs and present CENTAUR, a centralized scheduling framework that mitigates interference by intelligently scheduling downlink packets to avoid simultaneous transmissions on conflicting links. Next, we explore control plane mechanisms that manage contention by configuring the operating parameters for the wireless APs. As a part of our efforts into the control plane, we present Model-TPC, a mechanism that facilitates robust and practical transmit power control in enterprise WLANs by determining the set of feasible power levels for different APs in the system. Both CENTAUR and Model-TPC require real-time interference estimates to function efficiently in dynamic wireless environments. Towards that end, we propose PIE, an online interference estimation mechanism for enterprise WLANs that merges traffic reports from wireless APs to generate dynamic interference estimates. Finally, we believe that the mechanisms presented in this dissertation are simple, yet effective and can be used as building blocks for designing more sophisticated tools towards managing interference in enterprise WLANs.

Suman Banerjee

ABSTRACT

As the number of wireless devices continues to grow in offices and enterprise environments, Wireless Local Area Networks (WLANs) have emerged as an important part of an enterprise network. Further, such enterprise WLANs tend to have a centralized architecture, which facilitates management and better design of various control and security functions. In spite of significant progress made in planning, deploying, and managing enterprise WLANs, radio interference remains a core concern among WLAN users, network administrators, and operations staff alike. This dissertation explores the design and implementation of both data and control plane mechanisms that leverage the centralized architecture of enterprise WLANs to manage interference effectively in such environments.

We first explore the design of a centralized data path for enterprise WLANs and present, CENTAUR , a centralized scheduling framework that mitigates interference by intelligently scheduling downlink packets to ensure that transmissions on conflicting links do not proceed simultaneously. It leverages the fact that in a enterprise WLAN with centralized architecture, most downlink traffic passes through a centralized controller, which is at a unique vantage point to schedule that traffic. In CENTAUR, we take a fresh, implementation and deployment oriented view in understanding data path choices in enterprise WLANs. We perform extensive measurements to characterize the impact of various design choices, like scheduling granularity on the performance of a centralized scheduler and identifying regions where such a centralized scheduler can provide the best gains. Our detailed evaluation with scheduling prototypes deployed on two different wireless testbeds indicates that although distributed channel access is quite robust in many scenarios, centralization can play a unique role in hidden and exposed terminal scenarios. Motivated by these observations, CENTAUR combines the simplicity and ease of distributed channel access with a limited

amount of centralized scheduling from a unique vantage point to improve client performance in such hidden and exposed terminal scenarios. Moreover, CENTAUR is implemented primarily by the centralized controller, with minimal modifications to the operating parameters of wireless APs in the WLAN.

Next, we explore mechanisms that manage contention by explicitly configuring the operating parameters for wireless APs. These mechanisms leverage the fact that the centralized controller has the global view of the system and can dynamically compute and update the operating parameters of APs over a fast wired backplane. As a part of our efforts in exploring such mechanisms, we present Model-TPC, a practical transmit power control mechanism for enterprise WLANs. In Model-TPC, we study the feasibility of using fine grained power control in enterprise WLANs and show that multipath, fading, shadowing, and external interference from wireless devices, make the implementation of fine grained power control challenging in practical settings. We then build an empirical model that determines appropriate number and choices of power values that are feasible in a given indoor setting and show how we can leverage such a model into implementing effective transmit power control for enterprise WLANs.

Both CENTAUR and Model-TPC are efficient interference mitigation mechanisms, but they require accurate, real-time interference estimates to adapt dynamically to constantly changing interference patterns in practical WLAN deployments. Towards that end, we propose PIE , an online interference estimation mechanism for enterprise WLANs that collects and merges traffic reports from wireless APs to generate interference estimates, which is in turn passed as input to interference mitigation mechanisms. Further, the most attractive feature of PIE is that it imposes no measurement traffic, and yet provides an accurate estimate of WLAN interference as it changes with client mobility, dynamic traffic loads, and varying channel conditions. Our experiments conducted on on two different testbeds show that PIE is able to not only provide high accuracy but also operate beyond the limitations of prior tools, providing a true solution to performance diagnosis and real time WLAN optimization, as manifested through its use in multiple WLAN optimization applications, namely channel assignment, transmit power control, and data scheduling.

Overall, the contributions of this dissertation are numerous. First, we collect detailed measurements in real wireless deployments to characterize and validate the key problems addressed in this dissertation. We show that interference can negatively impact the performance of clients in production WLANs, although the exact impact of interference varies dynamically with time and depends on the client's location and traffic patterns of other competing wireless transmitters. Second, we present key interference mitigation and estimation mechanisms, that adhere to practical design constraints like support for legacy clients, which makes them attractive for use in current deployments. Finally, we present a prototype implementation and evaluation of these mechanisms on real enterprise scale wireless testbeds and show that they provide consistent gains under diverse topology and traffic patterns. We believe that the mechanisms presented in this dissertation are simple, yet effective, and can be used as building blocks for designing more sophisticated tools towards improving client performance in enterprise WLANs.

Chapter 1

Introduction

Wireless networks have seen unprecedented growth and WiFi is increasingly being used for last mile Internet access under various settings, from unplanned hotspots and home networks to planned enterprise deployments and city-wide wireless mesh networks. Such ubiquitous WiFi proliferation has led to strong sales of WiFi-enabled mobile devices and smart phones, which will exceed $100 billion in 2010 [12]. Moreover, as the number of wireless devices continues to grow in offices and enterprise environments, Wireless Local Area Networks (WLANs) have emerged as an important part of an enterprise network.

1.1 Enterprise WLAN architecture

A typical enterprise WLAN consists of a set of Access Points (or APs) that are connected through a wired backplane. Such WLANs have been rapidly deployed in recent years by businesses and university campuses. There are two main approaches used for deploying and managing such enterprise WLANs:

Distributed WLAN architecture: Distributed WLAN architecture consists of a set of wireless APs connected over a wired backplane, where each AP has autonomy over access, security, and operation. Such distributed WLANs usually do not require a wireless controller and most access control and management functionalities are implemented by different APs in the system independently. Earlier WLANs were mostly distributed in nature and still there are some key advocates of this architecture, as it does not require a centralized wireless controller that can be expensive

in practice. Such distributed WLANs are currently been provided by vendors like Xirrus [20] and Aerohive [1].

Centralized WLAN architecture: In a centralized WLAN architecture, key configuration and management functionalities are offloaded from the APs to the central control element (i.e, the WLAN controller). Such centralized WLANs comprises of lightweight APs, under the control of a centralized WLAN controller, that observes the entire network and can potentially configure parameters, such as the channel of operation and the transmit power level, of each AP in the system. Further, in a centralized WLAN, all the downlink and uplink traffic passes through the centralized WLAN controller and hence provides a unique opportunity for the WLAN vendors to implement various security and access control policies at one location. Hence, most commercial grade WLAN controllers come equipped with advanced security and management software like Intrusion Detection Systems (IDS) [6, 3] for detecting malicious traffic and RADIUS Servers [19] for session authentication and access control. Such ease of management has prompted major WLAN vendors, e.g., Cisco [7], Aruba [2], and Meru [15], to move to a centralized WLAN architecture. In order to understand the key implications of choosing a particular architecture, we discuss their relative advantages (and disadvantages).

1.1.1 Distributed vs. Centralized WLANs

We compare the pros and cons of distributed and centralized architecture over several key metrics important for an efficient WLAN deployment and management.

Security: Enterprise WLANs routinely deploy mechanisms to detect rogue wireless transmitters and malicious traffic that may be harmful for the enterprise network. In a distributed architecture, such security mechanisms must be implemented and managed separately at individual APs, which can quickly become overwhelming for a large number of APs. On the other hand, in a centralized architecture, the wireless controller collects periodic feedback from the APs and performs analysis in one location. Such a centralized analysis allows system administrators to manage security alerts in one location and also helps detect a wider range of events (e.g. rogue AP detection), as compared to purely distributed analysis at the APs [167].

Policy enforcement: Creating and enforcing access control and other policies is an integral part of an enterprise WLAN management. Many such policies (e.g. load balancing and quality of service) require tight coordination between APs in the system and are harder to realize in a distributed architecture. Further, it is typically easier for system administrators to implement policies at one location (wireless controller in centralized WLAN) than multiple APs, reducing the chances of human error in centralized deployments.

Reliability: In a centralized architecture, the wireless controller monitors the entire network using feedback from the lightweight APs and can detect failed or faulty APs dynamically. The wireless controller can also increase the transmit power levels of neighboring APs to compensate for failed APs. However, the wireless controller is also a single point of failure in centralized WLANs, which can disrupt service in the entire WLAN. On the other hand, in a distributed architecture, failure of an autonomous AP can cause significant service disruption for the clients in the coverage area of the failed AP, but there is no single point of failure that impacts the entire WLAN.

Troubleshooting: Troubleshooting wireless performance is important for supporting enterprise clients. In centralized WLANs, the wireless controller periodically collects reports from the lightweight APs in the system and can correlate these traffic reports to diagnose the performance problems for end clients. Such coordinated troubleshooting efforts are significantly more difficult to implement in distributed WLANs, where fault diagnosis must be performed separately by each AP.

Scalability: In a distributed architecture, adding an autonomous AP requires significant effort, as it needs to be configured with the right parameters and policies upon installation. In contrast, addition of wireless APs is more seamless in a centralized architecture as most of the functionalities reside at the wireless controller, which can configure the APs remotely after deployment. However, in a centralized architecture, each new AP puts more processing load on the wireless controller, due to which multiple wireless controllers may be required for the larger deployments.

Radio-Frequency(RF) management: Wireless being a broadcast medium, the control settings, like channel of operation or power level of one AP, may impact the performance of other APs in

the system. As enterprise WLANs become larger in size and density, it is important to have a coordinated way of configuring APs to improve resource utilization and enhance client experience. In centralized architecture, the wireless controller has a unique vantage point to configure enterprise APs in a coordinated fashion, which is difficult to realize in a distributed WLAN architecture.

Cost: Typically, the cost of autonomous APs in distributed WLANs is higher than the cost of lightweight APs in centralized WLANs, but centralized WLANs must also incur the cost of a wireless controller to manage the lightweight APs. Further, the wireless controller in a centralized architecture often includes security features, like firewalls and intrusion detection, that may need to be additionally purchased for individual APs in a distributed architecture.

Apart from capital expenses to deploy the WLAN, there are operational expenses associated with each architecture. Such operational expenses include time and manpower required for configuring system-wide policies, troubleshooting, and upgrading the network. In a distributed architecture, such operational expenses increase significantly as the number of autonomous APs increases. On the other hand, in the centralized architecture, the operational costs are less impacted by network size, as an increasing number of lightweight APs can be seamlessly managed by implementing system-wide policies at the wireless controller.

1.1.2 Summary

Both centralized and distributed architectures have their pros and cons. However, given the current state-of-the-art in WLAN technologies, centralized architecture provides a better framework for large enterprise WLANs because network policies, security settings, and radio interference can be managed from a single device. Moreover, centralized WLANs also facilitate better troubleshooting and client mobility, and are easier to upgrade to evolving wireless standards. Such key benefits of centralized architecture have prompted major WLAN vendors to shift to a centralized architecture and current trends show that centralized WLAN architecture is the dominant choice for deploying enterprise WLANs [5].

1.2 Thesis goals

Key advantages and popularity of centralized architecture have led to significant efforts in planning and deploying centralized WLANs in enterprises. Still, enterprise WLANs typically operate in the unlicensed spectrum band and consequently need to address radio interference, which remains a core concern among WLAN users, network administrators, and operations staff alike. In addition, enterprise environments present a unique set of problems due to an increasingly mobile workforce and the inherent challenges of an indoor wireless environment (for example, varying signal attenuation depending on the shape of the building and materials used in its construction). Moreover, emerging applications, like audio and video streaming, put more stress on such enterprise wireless networks, which need to ensure high performance even under heavy traffic loads and interference from competing wireless devices. These factors, coupled with the robust performance requirements expected in an enterprise environment, make the handling of radio interference critical in enterprise WLANs. Motivated by these challenges in enterprise WLANs, this dissertation specifically tries to answer the question:

How can we exploit the centralized structure of enterprise WLANs to ensure efficient utilization of the wireless medium and provide a robust framework for managing and mitigating radio interference under varying client density, mobility, and traffic loads ?

We believe that the centralization of WLANs can be leveraged in solving this problem by providing fine grained control on managing interference in the system. The thesis of this research is that the centralized management of the data and control planes for 802.11 based enterprise WLANs is both necessary and desirable in achieving efficient operation of these networks.

We explain this further with a simple example. Consider two Access Points (APs) X and Y shown in Figure 1.1, each with one wireless client (C1 and C2 respectively), located in such a manner that the APs cannot sense the presence of each other, i.e., they are outside mutual carrier sense ranges. However, the clients are in carrier sense range of each other (and hence, in mutual interference range). Let us assume that the traffic is downstream along both these links, i.e., from

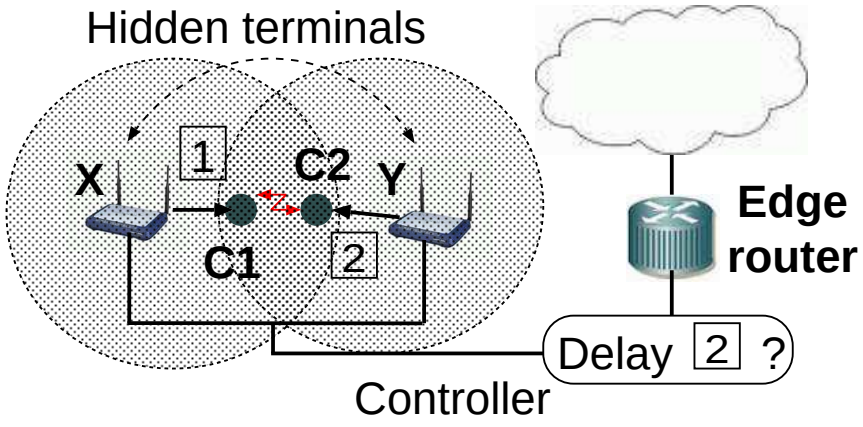


Figure 1.1: Centralization opportunity in the data path to avoid potential interference effects. The wireless controller can delay the packet 2 for client C_2 to avoid potential collision with packet 1 of client C_1.

the APs to the clients. It is easy to demonstrate that in such a scenario, depending on traffic patterns, one AP-client traffic can completely starve the other AP-client traffic, in spite of deploying existing QoS mechanisms, such as the 802.11e standards. On the other hand, a central traffic controller co-located with the edge router can make intelligent scheduling decisions for all such downlink traffic (Figure 1). In particular, the central controller can be placed on the data path and can be used to delay one downlink packet (say, packet 2 to be transmitted by AP Y to client C2) so that it does not interfere with a previously forwarded packet on an interfering link (say, packet 1 being transmitted by AP X to client C1). As shown in this example, managing the common resource – the wireless medium – can be much more efficient if we exploit the natural centralized structure of enterprise WLANs.

The goal of this dissertation is to study the challenges in efficiently managing the wireless spectrum and designing robust techniques to manage interference in an enterprise-wide wireless network setting. We exploit the inherent centralized architecture of an enterprise WLAN, where a central control element is wired to all the APs in the wireless network (as illustrated in Figure 1.1). Such a central network element (the WLAN controller) provides a natural platform to centrally configure and globally optimize channels and power levels at the APs. Further, the WLAN

controller, has a clear view of all downstream traffic [1] that will be transmitted across the entire enterprise WLAN. If the WLAN controller can infer something about different interference domains in the wireless network, it can perform data plane optimizations, like centralized scheduling, to meet desired objectives of traffic engineering and also mitigate interference in the system.

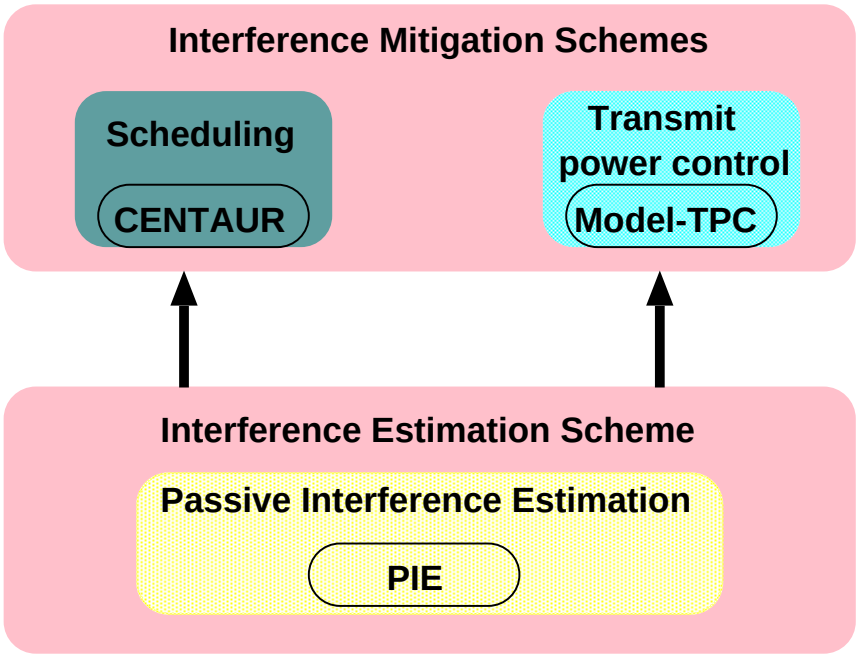


Figure 1.2: Thesis components.

Based on these observations, in this dissertation we have investigated mechanisms that leverage the centralized infrastructure to measure and manage interference effectively in enterprise WLAN deployments. The key difference from past work is that this takes a more practical, implementation, and deployment-driven approach in designing and evaluating these mechanisms. Consequently, our mechanisms show consistent gains when evaluated on diverse topologies and under realistic traffic patterns. We believe that our emphasis on practical design and rigorous implementation-driven evaluation makes our mechanisms especially attractive for deployment in current enterprise WLANs.

[1] About 80% of total enterprise traffic [51, 162] is downstream (from the Internet to the wireless clients).

Figure 1.2 presents a pictorial overview of the three key components of this dissertation. We first explore the design and implementation of a centralized scheduling framework, CENTAUR , that mitigates interference by scheduling downlink packets in an enterprise WLAN. The scheduling functionality can be implemented by the centralized WLAN controller that sits in the data path of downlink traffic and functions without explicitly modifying any key operating parameters of wireless APs. Next, we explore mechanisms that manage interference by explicitly configuring the operational parameters of wireless APs. We present the design and implementation of Model-TPC, a practical transmit power control mechanism that determines the minimum feasible transmit power levels for the wireless APs that sustains high throughputs for the wireless clients, while reducing the overall interference in the system. Both CENTAUR and Model-TPC are sound interference mitigation mechanisms, but they require accurate, real-time interference estimates to adapt dynamically to constantly changing interference patterns typical of real WLAN deployments [44]. We propose PIE, an online, passive interference estimation mechanism for enterprise WLANs. PIE collects and merges traffic reports from wireless APs to generate interference estimates, which is in turn passed as input to interference mitigation mechanisms. All the three mechanisms were implemented and thoroughly evaluated on multiple wireless testbeds and our experimental results demonstrate that they provide consistent gains under diverse wireless scenarios. We describe each of them briefly below.

1.2.1 CENTAUR – Hybrid data path for enterprise WLANs

In the first part, we design and implement, CENTAUR (Chapter 3), a hybrid data path for enterprise WLANs, that combines the simplicity and ease of distributed channel access [2] with a limited amount of centralized scheduling from a unique vantage point. In designing CENTAUR , we take a fresh, implementation and deployment-oriented view in understanding data path choices in enterprise WLANs. We perform extensive measurements to characterize the impact of various design

[2]Today, the primary mode of channel access in enterprise WLANs is the Distributed Coordination Function (DCF), as defined by the 802.11 standard. As the name suggests, it is a *distributed* technique, which employs a random access mechanism to resolve contention between multiple competing transmitters. We briefly describe the functioning of DCF in Chapter 3.

choices, like scheduling granularity on the performance of a centralized scheduler, and identify regions where such a centralized scheduler can provide the best gains. Our detailed evaluation with scheduling prototypes deployed on two different wireless testbeds indicates that distributed channel access is quite robust in many scenarios, but centralization of the data path can play a unique role in 1) mitigating hidden terminal scenarios, which may occur infrequently, but become pain points when they do and 2) exploiting exposed terminal scenarios, which occur more frequently, and limit the potential of successful concurrent transmissions. We show that CENTAUR not only delivers significant performance gains for scheduled traffic, but also improves the performance of the network as a whole due to the improved utilization of the wireless medium. Importantly, CENTAUR can be implemented by any individual WLAN vendor without any changes required for clients.

After exploring the design of a centralized data path, we investigate mechanisms that explicitly modify the operating parameters of wireless APs, like transmit power, to mitigate interference in the WLAN. Such mechanisms can be efficiently implemented by the centralized WLAN controller that has a global view of the network and can quickly modify the operating parameters of the wireless APs over the fast wired connection between the WLAN controller and the APs. As a part of our efforts into such control plane mechanisms, we explore the design of a practical transmit power control mechanism for enterprise WLANs.

1.2.2 Model-TPC – Practical transmit power control for enterprise WLANs

In this piece of work, we investigate the feasibility of using fine grained power control that minimizes contention in the enterprise WLAN by configuring the APs with the lowest possible power levels without sacrificing the performance of clients associated with those APs. Such an assignment can significantly increase transmission concurrency and can also reduce the impact of interference that an enterprise wireless AP has on the clients associated with other APs in the system. However, we observe that multipath, fading, shadowing, and external interference from wireless devices, make the implementation of such a fine grained power control challenging in practical settings. Our measurements show that, due to such practical challenges, only 3-5 power

levels are distinguishable for any realistic indoor setting and a failure to identify the correct set of power levels can negatively impact the performance of any power control mechanism by increasing its convergence time. In order to overcome these challenges, we propose a practical transmit power control mechanism, Model-TPC (Chapter 4), that determines the appropriate number and choices of power values adequate for any setting. We perform detailed experiments on an enterprise scale wireless testbed to show that Model-TPC allows the WLAN controller to quickly converge on the desirable power settings for wireless APs, providing significant throughput gains, especially in dynamic client mobility scenarios.

Finally, we explore the design of a practical interference estimation mechanism that can provide dynamic interference estimates to CENTAUR and Model-TPC, allowing those mechanisms to react efficiently to change in interference patterns.

1.2.3 PIE – Online, passive interference estimation for enterprise WLANs

In the third and final part of this dissertation, we investigate the design and implementation of a Passive Interference Estimator (PIE) (Chapter 5) that can dynamically generate fine-grained interference estimates across an entire WLAN by passively observing the wireless traffic at the APs and merging the local reports from different APs to construct a global view of the system. As discussed before, such dynamic interference estimates are critical for the robust performance of interference mitigation mechanisms. The most attractive feature of PIE is that it imposes no measurement traffic, and yet provides an accurate estimate of WLAN interference as it changes with client mobility, dynamic traffic loads, and varying channel conditions. Our experiments conducted on on two different testbeds show that PIE not only provides high accuracy but also operates beyond the limitations of prior tools, providing a true solution to performance diagnosis and real time WLAN optimization, as manifested through its use in multiple WLAN optimization applications, namely channel assignment, transmit power control, and data scheduling.

1.3 Thesis contributions

The high-level contribution of this dissertation is to design, implement, and evaluate practical control and data path mechanisms that leverage the centralized infrastructure of an enterprise WLAN to measure and manage contention in such deployments. The specific contributions of this dissertation are described as follows:

1.3.1 Practical Interference Mitigation Techniques

This dissertation explores the design and implementation of two key interference mitigation mechanisms, CENTAUR and Model-TPC, and shows how the centralized infrastructure can be leveraged to practically deploy these mechanisms. For example, CENTAUR leverages the fact that most downlink traffic in an enterprise WLAN passes through the centralized wireless controller, which can perform intelligent scheduling on those packets to mitigate downlink interference in the system. In Model-TPC, the centralized controller uses periodic feedback from the APs to determine feasible transmit power levels for different APs and then computes the globally optimal power level assignment that minimizes the overall contention in the system. It dynamically computes these power levels for all APs in the system and updates the power setting of the wireless APs in real time over the fast Ethernet backplane. Note that these mechanisms are complimentary to each other and can be used in conjunction to provide even better interference mitigation in challenging enterprise environments.

1.3.2 Practical Interference Measurement Technique

Accurate, fast, and scalable interference estimation is critical for the interference mitigation mechanisms to be effective in real settings. This dissertation also explores the design and implementation of an interference estimation mechanism, PIE , that leverages the fast Ethernet backplane of enterprise WLANs to merge traffic reports from wireless APs at the centralized controller and generate real-time interference estimates for the entire network. PIE is a completely passive mechanism and does not introduce any additional wireless traffic in the system. We show that it

can provide accurate estimates in the presence of mobility and varying traffic patterns of wireless clients. We integrate PIE with the interference mitigation mechanisms proposed in this dissertation and show that it can facilitate robust performance of such mechanisms in dynamic wireless environments.

1.3.3 Measurement-driven Problem Validation

The key problems tackled in this dissertation have been carefully validated using detailed measurements from real WLAN deployments. Such measurement-driven validation allows us to identify the real pain points and challenges for enterprise WLANs and also provides critical insight for designing effective mechanisms to overcome those challenges. Below we outline the measurement studies undertaken by us as a part of our problem validation exercise:

- As a part of our exploration into the usefulness of a centralized data path in mitigating interference, we identify hidden and exposed terminals as two key scenarios, where distributed channel access performs poorly and centralization can play a unique role in improving client performance. Prior to designing the centralized path to solve these problems, we first validate that these are important problems to begin with. We show that downlink hidden and exposed terminals are prevalent in multiple enterprise WLANs through analysis and measurement of two production WLANs, as well as measurements on our wireless testbeds. We quantify the performance loss observed due to hidden and exposed terminals in such settings. We observe that about 34% of the links in the two production WLANs experience some form of hidden terminal interference from other APs in the same WLAN. Further, a few links in those measured WLANs experience severe interference, i.e. a throughput loss of greater than 80% in presence of hidden terminals. On the other hand, our exposed terminal measurements on an enterprise scale wireless testbed shows that about 41% of the wireless link pairs are exposed terminals that could double their throughput using a more intelligent channel access mechanism to share the medium simultaneously. These measurements clearly indicate significant potential gains for an intelligent channel access mechanism that can overcome the shortcomings of DCF in these hidden and exposed terminal scenarios.

- As part of our efforts to design a practical mechanism to configure the transmit power of enterprise APs and minimize interference, we identify significant fluctuations in Signal to Noise Ratio (SNR) of indoor wireless signals as a key challenge in practically realizing fine-grained power control mechanisms. Fluctuations in SNR introduces errors into the feedback loop of the power control mechanism, which in turn leads to poor performance of fine-grained power control mechanisms when deployed in indoor settings. To validate this problem, we first collect detailed wireless traces at multiple WLAN deployments and analyze those traces for SNR fluctuations. Our analysis shows signal strength fluctuations are present in most WLAN deployments; however, the extent of fluctuation depends on the type of building (shape, material) in which the WLAN is located and also the exact nature of wireless interference in that building. We observe that SNR can typically show a variation of upto $\pm 2dBm$ in line-of-sight scenarios, while in non line-of-sight scenarios, the fluctuations can be as high as $\pm 4dBm$, thereby limiting the usefulness of fine grained power control, which typically relies on varying transmit power in units of $1dB$ or smaller. These measurements show that SNR variations are a real phenomenon and we show how to take them into account when we present the design and implementation of Model-TPC in Chapter 4.

1.3.4 Evaluation on Enterprise-scale Wireless Testbeds

All three systems presented in this dissertation have been deployed and evaluated in actual wireless testbeds. Since our focus is on enterprise WLANs, our testbeds consist of a set of APs that are connected over a fast Ethernet backplane to a centralized WLAN controller. Below we summarize the evaluation setup for the mechanisms presented in this dissertation:

- We evaluate CENTAUR and PIE over two different testbeds, each with a different wireless platform, NIC, and wired backplane. (i) Testbed 1: deployed across five floors of a building, consisting of 30 Soekris 4826 nodes (266 MHz) equipped with Atheros-based 802.11 abg NICs, interconnected with a 100 Mbps Ethernet backplane, and (ii) Testbed 2: deployed across a single floor of a building, consisting of 20 VIA nodes (1.2 GHz) equipped with Intel 2915 802.11 abg NICs, interconnected with a Gigabit Ethernet backplane.

- We evaluate Model-TPC over a 802.11 wireless testbed spread across one floor of a building consisting of 7 VIA nodes (1.2 GHz) equipped with Atheros-based 802.11 abg NICs, interconnected with a 100 Mbps Ethernet backplane.

1.3.5 Practical Evaluation Showing Consistent Gains

Testbed evaluations indicate that our mechanisms perform consistently under diverse scenarios, with the exact performance gain dependent on the topology and traffic scenarios. Below we summarize the key experimental results from this dissertation:

- We evaluate the performance of CENTAUR on two wireless testbeds through a combination of controlled experiments as well as by playing back real traffic traces on these testbeds. Our results indicate up to $1.48\times$ improvement in data throughputs, $1.38\times$ reduction in web transaction completion times, and $1.21\times$ improvement in MOS for VoIP-like traffic for CENTAUR .

- Similar to CENTAUR , we evaluate the performance of Model-TPC by running web traffic for mobile clients in enterprise WLAN setting. We show that using Model-TPC for joint transmit power and data rate adaptation can yield up to 45% throughput gains as compared to naive mechanisms that explore all power levels for adaptation. Further, we observe that the set of feasible power levels for all seven receivers in the testbed was between two to four, reinforcing the usefulness of using Model-TPC, which narrows the search for power level adaptation, resulting in quicker convergence to the right power levels. Such reduction in convergence time can provide significant performance gains, especially in high mobility scenarios.

- We evaluate the accuracy of PIE as compared to state-of-art bandwidth tests on multiple links. We show that for 80% of the links, the interference estimate of PIE is within $\pm 10\%$ of the estimate produced by bandwidth tests. We show that in high traffic scenarios, PIE can converge on the correct interference estimate within 100ms and even in low traffic scenarios (where interference events are infrequent), it converges within 600ms. This represents

an order of speed up over state-of-the-art bandwidth tests. Further, integration of PIE with interference mitigation mechanisms, like data scheduling, transmit power control, and channel assignment, indicate that using PIE can provide throughput gains up to 40% in dynamic wireless scenarios, like high mobility and rapidly changing traffic patterns.

1.4 Relation to previously published work

CENTAUR was published in the proceedings of the 15th Annual International Conference on Mobile Computing and Networking [149], along with co-authors Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Dina Papagiannaki, Srinivasan Keshav, and Arunesh Mishra. This dissertation describes the system in greater detail, including more discussion on alternative design choices for fine grained scheduling in enterprise WLANs, specifically the potential of speculative scheduling under different scenarios.

Model-TPC was published in the proceedings of the 8th Internet Measurement Conference 2008 [148], along with co-authors Arunesh Mishra, Dheeraj Agrawal, Suman Banerjee, and Tamer Nadeem. This dissertation provides a greater motivation for the benefits of using an empirical model for determining feasible power levels in an indoor wireless environment.

The work on passive interference detection is currently under submission.

1.5 Dissertation structure

Chapter 2 discusses related work on capturing and mitigating interference in wireless networks. Chapter 3 presents CENTAUR, a centralized scheduling framework for enterprise WLANs, and describes its design, implementation, and detailed evaluation on on two large scale wireless testbeds. Chapter 4 details our modeling efforts in determining the set of feasible power levels in indoor wireless environments and using those power levels to implement a robust transmit power control mechanism suitable for use in enterprise WLANs. Chapter 5 describes the design and implementation of PIE , a tool to capture interference in real time for enterprise WLANs. It also presents the results from our efforts to integrate PIE with multiple interference mitigation mechanisms like

power control, channel assignment, and centralized scheduling. Chapter 6 concludes and presents avenues of future research.

Chapter 2

Related Work

In this chapter, we discuss prior research efforts towards efficient contention management in wireless networks, broadly classifying them into data and control plane mechanisms. Section 2.1 provides a brief overview of data plane mechanisms, like packet scheduling, and error recovery, for wireless networks. We also discuss prior approaches for real-time scheduling in operating systems and wired networks, and their relevance for the scheduling framework proposed in this dissertation. Section 2.2 discusses the key control plane mechanisms, channel assignment and transmit power control, that are typically used for managing interference in wireless environments. We also discuss prior approaches for estimating interference in wireless networks, broadly classifying them into active and passive estimation mechanisms.

We identify three key properties of such prior data and control plane mechanisms: 1) centralized or distributed framework, 2) support for legacy clients (are client modifications required ?), and 3) the evaluation methodology (theoretical, simulation, or real implementation). These properties are central to the design and implementation of different mechanisms proposed and evaluated in this dissertation. Identifying such properties allows us to put our work in perspective in terms of other prior approaches for interference mitigation and estimation in wireless networks.

2.1 Data plane

Data plane mechanisms try to improve client performance by intelligently manipulating different parameters of the client traffic, like modifying per-packet delay in scheduling, and adding redundancy to packets for error correction. We categorize such data plane approaches into data

scheduling, differentiated services (to provide Quality of Service (QoS)) and error correction mechanisms. Table 2.1 shows the key implementation and evaluation properties of such data plane mechanisms. Next, we discuss prior approaches in each category of data plane mechanisms.

Domain	Solution	Properties		
		Infrastructure	Compatibility	Evaluation
Wireless Scheduling	Vaidya et al. [160]	Distributed	Client modifications needed	Simulations
	Kanodia et al. [85]	Distributed	Client modifications needed	Simulations
	TBR [156]	Distributed	No client modifications	Experiments
	Hadaller et al. [66]	Distributed	Client modifications needed	Experiments
	MiFi [34]	Centralized	Client modifications needed	Simulations
	MIM [119]	Centralized	Client modifications needed	Experiments
	CMAP [161]	Distributed	Client modifications needed	Experiments
	TDMA style [55, 83]	Distributed	Client modifications needed	Experiments
Differentiated services	AMC [105]	Distributed	Client modifications needed	Simulations
	CRS [154]	Distributed	Client modifications needed	Simulations
	802.11 [11, 59, 166]	Distributed	No client modifications	Experiments
	Kse et al. [97]	Distributed	Client modifications needed	Simulations
	Tan et al. [158]	Distributed	Client modifications needed	Simulations
Error correction	Zigzag [61]	Distributed	Client modifications needed	Experiments
	SIC [68]	Distributed	Client modifications needed	Experiments
	PPR [77]	Distributed	Client modifications needed	Experiments
	Soft [165]	Distributed	Client modifications needed	Experiments

Table 2.1: Identifying key properties of prior data plane mechanisms as relevant to this dissertation. We identify three key properties of prior approaches: 1) centralized or distributed framework, 2) support for legacy clients (are client modifications needed ?), and 3) evaluation methodology (simulations or experiments).

2.1.1 Wireless Scheduling

In the research community, people have thoroughly studied scheduling-based channel access and there is a large body of literature dealing with efficient scheduling in wireless networks. We provide a brief overview of some key data scheduling mechanisms proposed in literature.

Vaidya et al. [160] propose a distributed fair scheduling algorithm for WLANs, such that different flows are allocated bandwidth in proportion to their weights. They propose a Distributed Fair Scheduling (DFS) approach obtained by modifying the Distributed Coordination Function (DCF) in IEEE 802.11 standard to dynamically choose the backoff interval on a per flow basis, thereby facilitating more fine-grained bandwidth shaping than possible using simple DCF. They evaluate their mechanism using simulations and numerical analysis.

Kanodia et al. [85] suggest distributed scheduling mechanisms in multi-hop wireless environments with specific delay and throughput constraints. They develop two key mechanisms. First, they facilitate exchange of QoS information between nodes by piggybacking priority information onto handshake packets (like RTS). Second, they adapt the priority of packets with the varying channel conditions so that it can meet delay and throughput constraints under dynamic environments. They develop an analytical model to evaluate their mechanism and also perform NS-2 [16] simulations to study its performance in some scenarios.

TBR (Time Based Regulation) [156] is a simple scheme that works in conjunction with any MAC protocol to provide long-term time-based fairness in AP-based WLANs by appropriately scheduling packet transmissions. They show that such time-based fairness can solve rate anomaly issues in 802.11 networks, where one slow client can potentially degrade the performance of the entire network. They implement their scheme on a single AP system with multiple clients operating on diverse rates, and show that their scheme can be useful in the presence of rate diversity. However, their scheme is limited to a single AP system and is evaluated using small scale experiments.

Similarly, other mechanisms have been proposed that use frame size modifications [170, 21] and time division access to allow time based fairness in the presence of rate diversity. The idea here 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, once it wins in the channel contention period. Nevertheless, in this approach, the client is still free to choose the most suitable transmission data rate and corresponding packet size for its transmission, to meet the deadline requirements specified by the channel access protocol. However, the majority of such mechanisms have been evaluated using NS-2 [16] simulations and analytical models.

Recently, researchers have also explored the usefulness of data scheduling in wireless mesh networks. It has been well documented that state of the art random access mechanisms using carrier sensing work poorly for both short and long distance wireless mesh settings [55, 145, 146, 40]. As a result, TDMA-style scheduling has been popular for both long distance 802.11 links [135, 134, 127, 120] and wireless mesh networks [55, 83]. Most of these mechanisms estimate traffic load and interference in the system and accordingly adopt the operating parameters of the TDMA scheduler, like dynamic slot sizing in [120], to provide throughput and latency improvements. Most such mechanisms are evaluated using testbed experiments and trace driven simulations. However, they use a naive hop distance-based approach to characterize interference between neighboring nodes, and also do not have fine-grained control over frame transmissions.

MIM [119] allows a receiver to disengage from an ongoing reception and engage onto a stronger incoming signal. Links that otherwise conflict with each other can be made concurrent with MIM. However, the concurrency is not immediate, and can be achieved only if conflicting links begin transmission in a specific order. Their work presents the design and implementation of a scheduling framework that orders the transmissions such that they are received correctly at their respective receivers even when they are transmitted in the same slot. While their mechanism has merit, it requires disabling of carrier sensing, which can make it difficult to coexist with other unscheduled traffic.

CMAP [161] presents a system that infers interference between links on the basis of packet reception probability and opportunistically disables carrier sensing whenever possible. Each node individually builds an interference map of the system, depicting its view of the conflicts in the system and takes such conflicts into account while transmitting packets to any particular destination. While such mechanisms can provide substantial throughput gains in interference prone WLANs,

they require firmware changes to the client's wireless NIC, which makes it difficult for them to be readily deployed in current WLAN scenarios with legacy devices. Also, since CMAP depends on the clients to decode headers of interfering packets, it cannot detect conflicts that are outside the transmission range but inside the interference range of the client.

The centralized scheduling technique most closely related to our work, is by Bejerano and Bhatia [34]. They proposed an architecture, called MiFi, that uses PCF-style polling based channel access control for APs and clients. MiFi requires clients to inform APs and the infrastructure about their traffic requirements. Unlike our work, MiFi focused more on the efficient design of fair algorithms, and was evaluated through simulations.

In [93], the authors propose an epoch based scheduling framework, where scheduling decisions are taken at the granularity of an epoch. They use epochs to aggregate knowledge about traffic demands in a distributed environment, while we use epochs to hide inaccuracies in scheduling due to variable latencies on the path for downlink scheduling in enterprise WLANs. More importantly, their mechanisms required inherent changes to the clients, while one of our key goals in data plane centralization is to keep the clients unchanged.

2.1.2 Differentiated services

Quality of service (QoS) is an important consideration in networking, but it is also a significant challenge. Providing QoS guarantees becomes even more challenging when you add the complexities of wireless and mobile networks, where there are multiple contention domains that change dynamically due to varying external interference and client mobility patterns.

In [105], an architecture is proposed that combines QoS reservation and scheduling at the MAC layer with Adaptive Modulation and Coding (AMC) at the physical layer. In AMC, the method for transmission changes when the link quality changes. For example, if the link quality degrades, the physical layer may start transmitting using BPSK instead QAM-16. As a result, more time will be required to send the same amount of data, so the MAC layer must adjust its schedule accordingly. Using this scheme, throughput performance closely matches the performance of the channel. When the link quality is good, it will take less time to transmit the QoS-guaranteed traffic

than it will take when the link quality is bad. At these times, there will be more resources available to transmit best-effort traffic.

In a contention-based network, it is possible to reduce the probability that a collision will occur for any given packet simply by reducing the size of the packet. In [171], the authors suggest a way to make use of this phenomenon in order to aid in making QoS guarantees. In the proposed scheme, real-time traffic has a smaller window size than best effort traffic. Also, the real-time data is not re-transmitted by ARQ. As a result of these changes, the real-time traffic has a much better chance of being transmitted without a collision and therefore receives a better level of service. When combined with admission control, the proposed scheme can make guarantees on throughput, delay or loss, and on some combinations of the three.

802.11e [11] is an extension of the popular 802.11 wireless LAN protocol. This extension enables multiple service levels and can provide service guarantees for network traffic. This extension enables audio and video over home and office 802.11 networks. Though 802.11e is a vast improvement over 802.11 in terms of the potential for QoS, many people have identified improvements [59, 166, 105, 171, 154] that can be made to 802.11e in order to make it even more capable of satisfying QoS requirements.

In [97], the authors extend the Enhanced Distributed Coordination Function (EDCF) in 802.11e so that the Inter Frame Spacing (IFS) is modified when the link quality changes. When link quality degrades, the IFS for high priority traffic increases and the IFS for best effort traffic decreases. Back-off periods are similarly modified. The effect of the change is that when the link quality decreases, best effort traffic is sacrificed in order to continue to meet the guarantees of the higher priority traffic.

2.1.3 Real time scheduling

There is a large body of work that focuses on real-time scheduling of jobs to optimize different metrics like average response time and throughput. Such scheduling frameworks have been proposed and implemented in various contexts, most notably in the realms of real-time operating systems [133, 104, 94, 33], web-servers [142, 70, 46], job scheduling in multi-processor systems [91],

and packet scheduling in wired networks [47, 126]. We broadly classify such real-time scheduling mechanisms into the following categories:

2.1.3.1 Clock-driven scheduling

This class of scheduling mechanisms [30, 41, 104] is used mainly for hard real-time systems where the processing time for each job is known a priori, such that offline scheduling techniques can be applied to optimize the target metric. In this framework, scheduling decisions are taken periodically and computing the schedule offline reduces the overhead of scheduling in run time. Although simple to implement, they are not flexible and hence cannot handle dynamic changes to execution times or job priorities [113].

2.1.3.2 Priority-driven scheduling

In priority driven scheduling framework, each job is assigned a priority on the basis of some criteria, like job length, deadline for completion, etc. According to their respective priorities, jobs are placed in different queues and each time the resource becomes free, the job with the highest priority is scheduled for execution. One key example of such priority based scheduling is Earliest Deadline First (EDF), where jobs are assigned priorities on the basis of their respective deadline to finish execution. The job with the earliest absolute deadline has the highest priority at any instant. This scheme requires the knowledge of deadlines for all processes that are scheduled.

2.1.3.3 Round-robin scheduling

These mechanisms are typically used for scheduling real-time processes in time-shared systems. In this framework, the scheduler iterates over all processes that are ready to execute and schedules them to run for one time slice each. Accordingly, when there are n jobs competing for the resource, each job will get one time slice of every n time slices (n time slices is called a round). Many variations of such round-robin scheduling framework have been proposed for packet

switched networks [147, 153, 122, 47, 126]. Such mechanisms in packet switched networks provide some key advantages over First In First Out (FIFO) scheduling, in terms of fair bandwidth allocation and lower delay for processes that use less than their full share of bandwidth.

Summary: In one of the key pieces of this dissertation, we have explored the design and implementation of a scheduling framework for managing interference in enterprise wireless networks (CENTAUR - Chapter 3). In our work, our focus was to implement and test the feasibility of a simple centralized data path and accordingly we adopt a simple scheduling framework. We wanted to first explore the extent of gains from a simple design and the exact regions where such gains can be realized in a WLAN Hence, we schedule each incoming packet in FIFO fashion such that its completion time is minimized. Still, our scheduling framework can be extended to incorporate more sophisticated scheduling approaches, like EDF or other round-robin scheduling mechanisms. Instead of scheduling packets as they arrive, the scheduler can cache some packets and then determine the best order of scheduling them for transmission, depending on their relative priorities or transmission durations (similar to execution times in real-time scheduling frameworks).

2.1.4 Error Recovery

Zigzag [61] 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 cancel interference from erroneously received frames and 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) [68] 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 needs access to the PHY layer of the radio and requires requires client side modifications.

Partial Packet Recovery (PPR) [77] and Soft [165] are approaches proposed recently to tackle high losses in the wireless medium. Both of these mechanisms exploit physical layer hints to recover bits at the MAC layer. PPR uses physical layer hints to find incorrect chunks in the packet and then retransmits only those chunks instead of the entire packet as in conventional 802.11 mechanisms. On the other hand, Soft uses this physical layer information from multiple erroneous packets to reconstruct the correct packet. These mechanisms require access to the PHY layer of the radio and modifications to both the wireless transmitter and receiver.

2.1.5 Commercial solutions

Centralized controllers are commercially available, from vendors such as Cisco [7] and Aruba [162], but they typically operate only in the control plane. Centralization of data, though recognized as providing more control, is harder to implement, and therefore less common. A few examples of such a design exist. For example, Meru Networks has proposed cellular-like coordination of various APs and scheduling mechanisms to provide a certain degree of deterministic channel access in enterprise WLANs [163, 164]. The proprietary nature of Meru's solution makes it difficult to present a detailed comparison with CENTAUR. However, through private communication, we have established that Meru's solution has some fundamental differences from CENTAUR's approach of hybridization and in the specific mechanisms implemented to detect and handle the exposed and the hidden terminals.

2.1.6 Summary

While these techniques are intuitively appealing, to the best of our knowledge there are no careful studies of the feasibility of data plane centralization in an enterprise WLAN through actual prototype design and implementation. In this dissertation, we present a first-of-its-kind implementation-based evaluation of challenges associated with such data plane centralization. Our focus has been on understanding and evaluating the impact of various challenges, including latency and jitter on control paths, uncertainties of the medium, impact of frame losses, re-transmissions,

Mechanism	Target problem	Approach	Changes to clients or NIC firmware?	Evaluation testbed
CMAP [161]	ET	Conflict graph and DCF	Yes	802.11
ZigZag [61], SIC [68]	Collisions (HT)	Symbol/signal manipulations	Yes	USRP
CENTAUR (Chapter 3)	HT and ET in enterprise	Conflict graph, DCF, and scheduling	No	802.11

Table 2.2: Comparing CENTAUR with recently proposed mechanisms of mitigating interference.

and data rate adaptations, and the need for some limited amount of centralized scheduling for solving hidden and exposed terminal problems in enterprise WLANs.

Since the main focus of our data-plane centralization is to mitigate interference in enterprise environments, we outline the key differences between CENTAUR and other recent approaches to interference mitigation in Table 2.2. While such interference mechanisms [61, 68, 89, 161] can provide substantial throughput gains in interference prone WLANs, they require firmware changes to the receiver's wireless NIC, which makes it difficult for them to be readily deployed in current WLAN scenarios with legacy devices. In contrast, CENTAUR only requires a software update to the wired Ethernet driver at the centralized controller, making it an attractive approach for current enterprise WLANs that want to support legacy wireless devices. This ease of deployment was a critical factor that enabled us to implement and test our system on two different testbeds with relative ease.

2.2 Control plane

We classify all mechanisms that tune the operating parameters for wireless devices like transmit power and channel of operation as control plane mechanisms. Table 2.3 classifies control

plane mechanisms into three main categories of channel assignment, transmit power control and interference estimation, and outlines their key properties to contrast them with the control plane mechanisms proposed and evaluated in this dissertation.

2.2.1 Channel Assignment

We first discuss different mechanisms proposed for assigning wireless channels to competing transmitters in a wireless environment. Intelligent channel assignment can significantly enhance system performance as it minimizes interference in the wireless medium and also facilitates multiple competing transmissions to proceed simultaneously. Below we describe some prominent channel assignment schemes proposed in the literature.

The CFAssign methods in [110] address the joint problem of channel assignment and load balancing in centrally managed WLANs. Their work shows that client-driven mechanisms are important in capturing interference accurately. By comparing against prior vertex coloring based approaches [109], they also show that vertex coloring approaches tend to be inefficient for centrally managed WLANs. However, CFAssign requires client participation to accurately estimate the interference sets and also to perform load balancing.

In [28], Bahl et al. propose a protocol that uses channel hopping for capacity improvement in wireless ad-hoc network where nodes have a single interface. The primary application of channel hopping in their work was to improve the capacity of the network as a whole by enabling these single-interface nodes to utilize multiple non-overlapping channels. Their mechanism has been evaluated using detailed simulations and requires both wireless transmitters and receivers to be modified for their scheme to work.

Channel assignment in cellular networks is a well-studied problem [136]. The *cells* in a cellular network have very regular properties, compared to 802.11 APs. Each cell has a relatively large coverage area and a high powered base station is used to connect to the cellular phones. Studies such as [96, 107], focus on centralized optimization schemes, including a mixed linear integer programming based model to determine an efficient channel assignment for cellular networks. These centralized schemes work well in cellular networks as the channel assignment is computed

Domain	Solution	Properties		
		Infrastructure	Compatibility	Evaluation
Channel Assignment	CFAssign [110]	Centralized	Client modifications needed	Experiments
	SSCH [28]	Centralized	Client modifications needed	Simulations
	Ko et al. [81]	Distributed	Client modifications needed	Simulations
	MDG [38]	Centralized	No client modifications	Experiments
	Cellular networks [136, 96, 107]	Centralized	Client modifications needed	Simulations
Transmit power control	PCMA [75]	Distributed	Client modifications needed	Simulations
	SHUSH [144]	Distributed	Client modifications needed	Simulations
	COMPOW [118]	Distributed	Client modifications needed	Simulations
	PCM [82]	Distributed	Client modifications needed	Simulations
	Subbarao [155]	Distributed	Client modifications needed	Simulations
	MiSer [130]	Centralized	Client modifications needed	Simulations
	PERF [26]	Distributed	Client modifications needed	Experiments
	Symphony [132]	Centralized	Client modifications needed	Experiments
	MDG [38]	Centralized	No client modifications	Experiments
Interference Estimation	Reis et. al [53]	Centralized	Client modifications needed	Experiments
	Kashyap et al. [88]	Centralized	Client modifications needed	Experiments
	Padhye et al. [125]	Centralized	Client modifications needed	Experiments
	Niculescu et al. [123]	Centralized	Client modifications needed	Experiments
	Smarta [24, 23]	Centralized	No client modifications	Experiments
	Yeo et al. [79, 169]	Centralized	No client modifications	Experiments
	Jardosh et al. [78]	Centralized	No client modifications	Experiments
	Mahajan et al. [106, 138]	Centralized	No client modifications	Experiments
	Cheng et al. [44, 43]	Centralized	No client modifications	Experiments

Table 2.3: Identifying key properties of prior control plane mechanisms as relevant to this dissertation. We identify three key properties of prior approaches: 1) centralized or distributed framework, 2) support for legacy clients (are client modifications needed ?), and 3) evaluation methodology (simulations or experiments).

once and changes rarely. However, most of the work in cellular channel assignment is done using simulations and mathematical modeling.

2.2.2 Transmit power

Adaptive transmit power has been an active area of research in the wireless community. Researchers have explored applications of transmit power control for both conserving battery power of the mobile devices [130, 87, 128, 82, 62, 137] and improving wireless network capacity [132, 75, 152, 114, 103, 115, 116, 129, 108, 37, 118]. Below we first discuss transmit power control mechanisms proposed for improving wireless capacity and spatial reuse, followed by a discussion of mechanisms targeted for optimizing the energy consumption of wireless devices.

2.2.2.1 Improving wireless capacity

The schemes that fall into this category apply transmit power control to improve spatial resuse in wireless networks. The key idea in these mechanisms is to find the minimum power level at which the wireless nodes can operate without compromising their throughput. These mechanisms allow the wireless nodes to settle at transmit power levels that minimize overall interference in the system, as a result improving overall system throughput.

Monks et al. proposed a power controlled multiple access wireless MAC protocol (PCMA [75]). PCMA generalizes the transmit-or-defer "on/off" collision avoidance models to a more flexible "variable bounded power" collision suppression model. Using PCMA, the transmitter-receiver pairs can be more tightly packed into the network by adjusting the power level of the transmitter to the minimum required for a successful transmission, thereby allowing a greater number of simultaneous transmissions (spectral reuse). While the mechanism is very promising, it has been evaluated using simulations and hence does not take into account the practical challenges of implementing a transmit power control mechanism in a wireless medium.

Seth et al. propose a reactive transmit power control mechanism, called SHUSH [144], where nodes operate on the optimum (minimum) power required for communication. On detecting interference, SHUSH calculates the exact power required to send an RTS to the interferer and hence

optimizes the "floor space" acquired by any flow. Unlike PCMA, however, SHUSH transmits at a higher power only when a flow is interrupted by external interference. However, similar to PCMA, SHUSH has been evaluated using simulations and requires client modifications.

Kawadia and Kumar argue that power control should be a network layer function and develop the COMPOW protocol [118], in which routing layer agents are used to converge to a common power level for all nodes. However, as pointed out in [90], their scheme can be too conservative, especially when the wireless nodes are clustered rather than uniformly distributed.

Akella et al. [26] also discuss some power control mechanisms in their work on wireless hotspots. They propose that APs should use the minimum transmit power required to support the highest transmission rate. In their scheme, the receiver sends the value of observed RSSI, averaged over some small number of packets, as a feedback to the transmitter. The transmitter, on receiving the average RSSI value on the receiver side, decides the optimal power level suitable for use in the current channel conditions.

Ramachandran et al. propose Symphony [132], a two-phase rate and power control mechanism, that adjusts the power levels of the wireless transmitters to maximize overall network capacity. In Symphony, all wireless transmitters cycle through two phases, REFERENCE phase and OPERATIONAL phase. In the REFERENCE phase, Symphony estimates the best achievable performance for each link, and in the OPERATIONAL phase, it tunes the link to the lowest transmit power possible such that its performance is the same as in the REFERENCE phase. This two-step synchronous mechanism ensures that the tuning of transmit power for any wireless node does not degrade the performance of the system and hence the overall capacity always increases (or at least remains the same) at the end of OPERATIONAL phase. The authors have implemented and evaluated their scheme on a real wireless testbed that uses a centralized server to synchronize the two phases between different wireless nodes in the system. But their scheme requires modifications to both wireless transmitters and receivers, which may be difficult to realize for legacy wireless clients.

Finally, Broustis et al. propose MDG [38], which performs joint power control and channel assignment for wireless networks. MDG is a measurement-driven framework that determines the

optimum order of applying different interference mitigation mechanisms when they are to be used in conjunction. Their framework allows system administrators to efficiently deploy different interference mitigation mechanisms in conjunction and is a practical tool for improving the performance of current WLAN deployments.

2.2.2.2 Improving battery life

In this category, transmit power control is used for conserving the battery power of mobile devices. The key idea here is to determine minimum power level at which a wireless transmission can be successful, thereby minimizing energy consumption per transmission.

Many mechanisms in this category [87, 128, 151] exchange RTS/CTS frames at maximum power, while DATA and ACK frames are sent at lower power levels. Most recently, Vaidya et al. [82] point out a key weakness of such mechanisms, where reduced power levels for DATA-ACK packets reduce the corresponding carrier sensing zone for other nodes listening to the transmission and can lead to potential collisions. They propose a variation of this scheme in Power Controlled MAC (PCM), which modifies the aforementioned scheme by occasionally sending the DATA-ACK packets at a higher power level (once per Extended Inter Frame Spacing (EIFS)) to allow other neighboring nodes to carrier sense and defer to the transmission. Their mechanism has been evaluated using detailed simulations on NS-2 [16].

Subbarao [155] and Gomez et. al [62] have proposed a dynamic power-conscious routing mechanism that incorporates link layer and physical layer properties in routing metrics. They route the packet on a path that requires the least amount of total power expended and each node transmits with the minimum power required to ensure reliable communication. Such schemes require per packet power control and also need feedback from the destination regarding SNR on a per packet basis.

In MiSer [130], the authors consider the joint rate adaptation and transmit power control with the objective of minimizing energy consumption per wireless transmission. Their proposed mechanisms computes an offline rate-power table, which is used by wireless nodes to determine the most energy efficient rate-power combination to use for each data frame depending on signal strength,

interference and noise floor values for a given transmitter-receiver pair. Their mechanism is quite promising but has been evaluated primarily using simulations and also requires client-side modifications.

Mechanism	Objective	Granularity	Joint power-rate adaptation	Evaluation	Practical deployment
PCMA [75]	Capacity	Fine-grained	No	Simulations	Difficult
SHUSH [144]	Capacity/Energy	Fine-grained	No	Simulations	Difficult
MiSer [130]	Energy	Fine-grained	Yes	Simulations	Difficult
COMPOW [118]	Capacity	Fine-grained	Yes	Experiments	Difficult
Model-TPC (Chapter 4)	Capacity	Model-based	Yes	Experiments	Robust for indoor and outdoor settings

Table 2.4: Comparing Model-TPC with prior transmit power control mechanisms.

Summary: Most of these power control mechanisms (with the exception of COMPOW [118] and Symphony [132]) have been mainly evaluated using simulations. Such network simulations fail to capture the inherent complexity in realizing power control mechanisms for real deployments. For example, such mechanisms do not take into account significant SNR variations for indoor wireless environments that can significantly impact the performance of these mechanisms. We discuss the design and implementation of a practical transmit power control mechanism (Model-TPC) in Chapter 4 that overcomes such challenges and provides a robust way of implementing transmit power control in indoor environments. Table 2.4 compares Model-TPC with some key transmit power control mechanisms discussed in this chapter.

2.2.3 Interference estimation

There is a large body of work on estimating wireless interference. In particular, there has been a lot of work on characterizing and evaluating the capacity of wireless networks in the presence of interference [65, 88, 99, 57, 100, 102, 63, 58]. The seminal work in modeling wireless network capacity was presented by Gupta et. al in [65], which has been extended for node mobility ([63]),

network coding ([58]), and other traffic patterns ([102]). Further, researchers have also focused on modeling the performance of 802.11 DCF mechanism under multiple competing transmitters. However, such efforts either assume that all competing nodes are in communication range of each other [36, 99, 57], or the number of competing transmitters is restricted to two flows [56]. While these models provide helpful insights into the capacity of wireless networks under target scenarios, they are abstract and make many simplifying assumptions (like using a binary interference model), due to which they cannot be directly used in real settings. Such limitations have motivated much work towards measurement based modeling [53, 22, 92, 106, 44, 43, 143, 88, 131, 24, 123, 125, 161], where the models are seeded (and validated) through real measurements.

We classify such measurement-driven interference detection mechanisms into active and passive mechanisms. Active mechanisms [53, 125, 123, 92, 24, 23, 64, 88] comprise of those techniques that require active probes to estimate interference in the wireless system. On the other hand, passive mechanisms [161, 44, 43, 39, 106, 78, 143, 169, 79] comprise of techniques that do not inject any additional traffic into the system and estimate interference by passively observing the existing traffic.

2.2.3.1 Active mechanisms

Active mechanisms introduce control traffic into the system to estimate the link interference. Below we discuss some key active mechanisms for interference modeling.

Reis. et al. [53] propose an interference model that is seeded using pairwise signal strength measurement. In their scheme, each node (turn by turn) broadcasts probe packets and every other node in the system notes the signal strength of the probe packet received from the broadcasting node. Thus, using O(n) measurements, they can determine the link quality for all transmitter-receiver pairs and can also infer physical layer deferral and collision properties. However, their model works only for single interferers and requires receiver support to report the signal strength measurements.

Kashyap et. al [88] and Qiu et. al [131] both extend the model proposed in [53] for multiple interferers. Their models still require O(n) measurements, but use analytical methods to estimate

the impact of multiple interferers. Further, the model presented in [131] also handles unicast transmissions, which has a different underlying transmission procedure than broadcast. The accuracy of both of these mechanisms has been evaluated using simulations and testbed experiments. Similar to the model presented in [53], these models also require receiver participation in reporting the signal strength values for the probes.

Padhye et al. [125] proposed bandwidth test, a measurement-intensive approach for determining conflicts. The key idea is to systematically transmit a simultaneous burst of traffic along each pair of AP-client links and observe how the aggregate throughput differs from the throughput achieved by each link operating in isolation. The notion of conflict is a continuous value ranging from no conflict to very high conflict, depending on the degradation of throughput under interference. The authors in [125] use measurement bursts of 30 seconds in length, and show how O(n^2) measurements are needed to determine conflicts between n links, which is a significant overhead for real deployments.

The bandwidth test mechanism proposed in [125] was extended by Niculescu et. al. in [123] to compute a conflict graph for the entire WLAN. They show that the interference from different 802.11 sources is additive, and hence O(n^2) bandwidth tests are required to compute the conflict graph for a n node WLAN. However, this mechanism requires significant network downtime.

Ahmed et al. [24, 23] propose the use of micro-experiments, each lasting less than a millisecond, to detect different kinds of conflict between WLAN nodes. Each micro-experiment typically involves one or two APs, and requires them to transmit a packet each to specifically chosen clients. Based on the clients' normal 802.11 standard defined response (either with a valid ACK or its lack thereof), the APs infer the existence or absence of conflict between the specific AP-client links under test. Although, these micro-experiments can be performed with legacy clients, they require network downtime, which may be a significant deterrent for its application in real wireless deployments.

Summary: Although active mechanisms capture interference accurately for the control traffic, their interference estimate is a function of control traffic parameters, like packet size and data rate. It is difficult to extrapolate the interference estimate measured using control traffic to practical

scenarios where actual traffic parameters like packet size and rate may differ significantly from the control traffic parameters. Such a limitation, coupled with the requirement of network downtime, makes active mechanisms less likely to be used in realistic settings.

2.2.3.2 Passive mechanisms

In contrast to active mechanisms that introduce control traffic to measure interference, passive mechanisms observe the real traffic in the wireless environment to derive performance and interference measures. Such passive mechanisms have been an active area of research in the wireless community, ranging from low-level channel access studies for a pair of nodes [48, 49, 121, 143] to large scale passive measurements for understanding the performance of wireless users [71, 72, 95, 31, 157, 79, 169]. Such studies mainly focus on the application level performance of the users, deriving high level measures like application workloads, user mobility patterns, session durations, etc. Recently, there has been a significant push in the wireless community to compute low-level interference estimates for wireless environments using only passive measurements [79, 169, 78, 106, 138]. Below we discuss some key passive mechanisms to measure interference that have been proposed in the literature.

Yeo et al. [79, 169] were the first to investigate the use of multiple monitors to passively capture wireless traffic and process the traces collected by multiple monitors to derive interesting measures for the wireless network. As pointed out in their work, one of the key challenges in the use of such a monitoring infrastructure lies in merging the traces collected from multiple monitors to generate a consistent global view of the events taking place in the network. They propose the use of common beacon frames as a mechanism for synchronizing the traces and show that the use of multiple monitors can significantly enhance the monitoring accuracy. However, they do not explicitly discuss the mechanisms to derive interference measures from the synchronized traces and most of their experiments are reported on two to three monitors.

Similar to the efforts of Yeo et al. to monitor wireless traffic using multiple monitors, Jardosh et al. [78] also collect passive measurements from a large Internet Engineering Task Force (IETF) meeting using three monitors to capture traffic in the three orthogonal 802.11g channels.

Additionally, they perform detailed analysis on the collected traces to understand the link-level performance of wireless nodes under different levels of traffic loads. They also analyze the impact of 802.11 parameters, like backoff windows, and mechanisms, like rate adaptation, on the end user performance.

Mahajan et al. [106, 138] also capture traffic in a large conference using five monitors distributed across three orthogonal channels. They also use the common beacon packets to merge traces from multiple monitors and derive a global view of the wireless network. Their key contribution is a state-machine based learning approach that can infer missing wireless events that may be missed by the monitors and provide formulations for deriving low level interference measures like, packet delivery probability and channel utilization.

Cheng et al. have systematically tackled the problem of monitoring a large scale wireless environment using a dense deployment of monitors [44, 43]. They provide valuable insights into the challenges of realizing such a large scale monitoring infrastructure consisting of 192 monitors spread across five floors of a production WLAN deployment. They present detailed results outlining the challenges and their potential solutions in synchronizing and merging the traces collected from such a large number of monitors in real time. They also propose the idea of timestamping every packet in the wireless network at different points in the network (at wireless gateway, different points in the AP queue) and using those timestamps to infer low level measures, like the average time a packet spends in the AP's queue and the average backoff window for different wireless transmitters. Similar to [106], they propose formulations for deriving interference measures like packet delivery probability for different links in the system.

Summary: As discussed above, passive mechanisms are non-intrusive and do not require client modifications. However, most passive mechanisms require deployment of additional monitors in the target environment to capture wireless traffic efficiently. The cost and maintenance of these monitors makes such passive mechanisms less attractive for many realistic deployments. Also, most of these passive mechanisms to date perform offline trace merging and analysis and may not be suitable for capturing rapidly changing interference patterns. We discuss the key requirements

Mechanism	Changes to client NIC or firmware	Speed	Network downtime	Real traffic	Wireless control traffic
Interference maps [123]	Yes	Offline	High	No	High
Microprobing [24]	No	Online	Low	No	Low
CMAP [161]	Yes	Online	Zero	Yes	High (per-packet header)
PIE (Chapter 5)	No	Online	Zero	Yes	Zero

Table 2.5: Comparing PIE with other interference estimation mechanisms

of a practical interference estimation tools in Chapter 5, and propose PIE, an interference estimation mechanism that meets those requirements in practice. Table 2.5 presents a comparison of PIE with approaches in interference estimation as discussed in this chapter.

Chapter 3

CENTAUR : A hybrid data path for enterprise WLANs

The key goal of this dissertation is to design practical interference mitigation mechanisms that leverage the centralized structure of enterprise WLANs to improve client performance in such environments. Towards that end, we first exploit the property that downlink traffic in a centralized enterprise WLAN passes through the WLAN controller, which has a unique vantage point to intelligently schedule this traffic and mitigate interference for downlink transmissions. In this chapter, we present the design and implementation of CENTAUR, a hybrid scheduling framework that combines limited amount of centralized data scheduling with state-of-the-art distributed channel access to improve performance for clients suffering from hidden and exposed terminal problems, without incurring significant performance penalty for normal (non-hidden, non-exposed) clients in the system. In the next chapter, we explore the design of a centralized control plane mechanism that configures the wireless APs with suitable power levels to minimize the overall contention in the WLAN, thereby improving performance for the end clients. Today, the primary mode of channel access in enterprise WLANs is the Distributed Coordination Function (DCF) as defined by the 802.11 standard. As the name suggests, it is a *distributed* technique which employs a random access mechanism to resolve contention between multiple competing transmitters. Given the wasted airtime incurred by random backoff in DCF and the potential for collisions due to uncoordinated access it has been argued that centralization of data transmission decisions can improve network capacity [160, 117, 34, 85]. However, conventional wisdom also suggests that the overhead of centrally scheduling each data packet transmission can be prohibitive, while the DCF approach

is simple and has been shown to be adequate for most common scenarios. Therefore, the main question we pose in this chapter is the following:

Is there a useful role for a centralized data path in enterprise WLANs in which a central control element makes scheduling decisions about when individual frames should be transmitted by APs that are part of the enterprise?

After detailed experiments, we found significant merit in the conventional wisdom — despite its many known failings, DCF is particularly robust across a large range of scenarios, often more so than a carefully engineered centralized scheduling approach implemented on commodity 802.11 hardware. However, there exist two challenging scenarios, *hidden terminals* and *exposed terminals* where DCF performs poorly and centralization can play a unique role.

In this chapter, we present CENTAUR - the first contribution of this thesis. CENTAUR leverages a limited amount of centralization and explicitly mitigates the performance loss experienced by *downlink traffic* in enterprise WLANs, while indirectly also improving the performance of uplink traffic. More specifically, CENTAUR implements a centralization function for all hidden and exposed terminal links that are identified on the downlink wireless path. All remaining wireless traffic, e.g., uplink enterprise traffic as well as downlink traffic not experiencing hidden or exposed terminal interference, accesses the medium using the standard DCF mechanism. Thus, CENTAUR can be viewed to be half-centralized and half-DCF in nature [1]. We show that such a structure not only helps improve the performance of the downlink hidden and exposed terminals, but also provides an aggregate improvement for the entire WLAN across all uplink and downlink paths. An important property of CENTAUR is that it requires *no changes in the 802.11 clients.* In fact, the entire centralization functionality is implemented in a single central controller, and only requires a small amount of configuration changes in APs. Hence, CENTAUR can be independently implemented and deployed by a WLAN vendor.

The rest of the chapter, we first discuss the motivation for a centralized data path in enterprise WLANs (Section 3.1). We present the design and implementation of a simple deterministic

[1] It is analogous to the mythological creature, Centaur, which is supposed to be half-human and half-horse

centralized scheduler, DET, in Section 3.2 and discuss the key advantages and disadvantages of a purely centralized channel access. Motivated by the performance penalty incurred by DET due to the feedback messages, we discuss the design of a speculative scheduling framework (SPEC) in Section 3.3. We describe how SPEC can pipleline packets to mask the delays associated with feedback messages, but at the same time it performs poorly in dynamic wireless environments, where the probability of mis-speculation is high. In Section 3.4, we show how the hybrid design of CENTAUR carefully combines distributed channel access with limited centralization to provide significant gains under realistic settings. We first evaluate the performance of such a hybrid scheduling framework using targeted microbenchmarks in Section 3.5. We then evaluate the performance of CENTAUR on two large scale wireless testbeds using realistic traffic traces (Section 3.6). We finally conclude the chapter in Section 3.7, by discussing some lessons that we learned during our efforts towards implementing and evaluating CENTAUR.

3.1 Motivation

We first provide some background into the working and limitations of the state-of-the-art distributed channel access mechanism in IEEE 802.11.

3.1.1 Distributed channel access in 802.11

IEEE 802.11 uses a distributed channel access scheme, Distributed Coordination Function (DCF), which is a variant of Carrier Sense Multiple Access (CSMA) / Collision Avoidance (CA) protocol that allows the wireless devices to share the medium. A wireless device that wishes to transmit data, first senses if the medium is free and if it senses the medium to be free for a certain period of time (called Distributed Inter Frame Spacing (DIFS) in 802.11 standard), it transmits the data. On the other hand, if the medium is sensed to be busy, the wireless device selects a random backoff time that is upper bounded by the current contention window of the device. It then waits till the medium becomes free again, and then counts down the backoff timer. It finally transmits the data once the backoff timer expires.

Further, 802.11 also provides link layer reliability using retransmissions in the event the data packet is not correctly decoded by the receiver. To indicate successful data reception, wireless receivers send an acknowledgment back to the transmitter within a small period (defined as Small Inter Frame Spacing (SIFS) in 802.11), of correctly decoding the data transmission destined for them. If the data transmission is not successful, the transmitter doubles the contention window and chooses a new (possibly longer) backoff time. It again waits for the backoff timer to count down to zero before retransmitting the data packet. Although IEEE 802.11 DCF performs well in a wide range of scenarios, there are two key scenarios where such a distributed channel access falls short:

Hidden Terminals: This scenario occurs when two wireless transmitters *cannot* physically carrier-sense each other, and hence while contending for the wireless medium, they both transmit simultaneously. In hidden terminal cases, such simultaneous transmissions leads to collisions at one or both the intended receivers. This problem was first outlined in seminal papers by Karn [87] and Bhargavan et al. [35]. Distributed channel access scheme like DCF is unable to solve this hidden terminal problem efficiently, which can lead to severe performance degradation for the victim clients.

Exposed Terminals: This scenario occurs when two wireless transmitters *can* physically carrier sense each other, but they do not interfere with each others receivers. In this scenario, the transmissions of the two transmitters is incorrectly serialized, even though they could have successfully transmitted data to their respective receivers simultaneously. Exposed terminal problems can reduce the capacity of wireless networks by reducing transmission concurrency in the system.

In this chapter, we explore the possibility of leveraging the centralized data plane in enterprise WLANs to solve such downlink hidden and exposed terminal problems. Prior to describing our approach in solving performance problems in enterprise WLANs that occur due to downlink hidden and exposed terminals, we first validate that these are important problems to begin with. Intuitively, it may appear that both hidden and exposed terminal problems can be eliminated by carefully planning the AP locations, and efficiently assigning channels. However, in practice, both these scenarios occur due to arbitrary location of the clients in the system. Fig. 1.1 shows a scenario where APs X and Y are placed far enough apart that they cannot carrier sense (CS) each

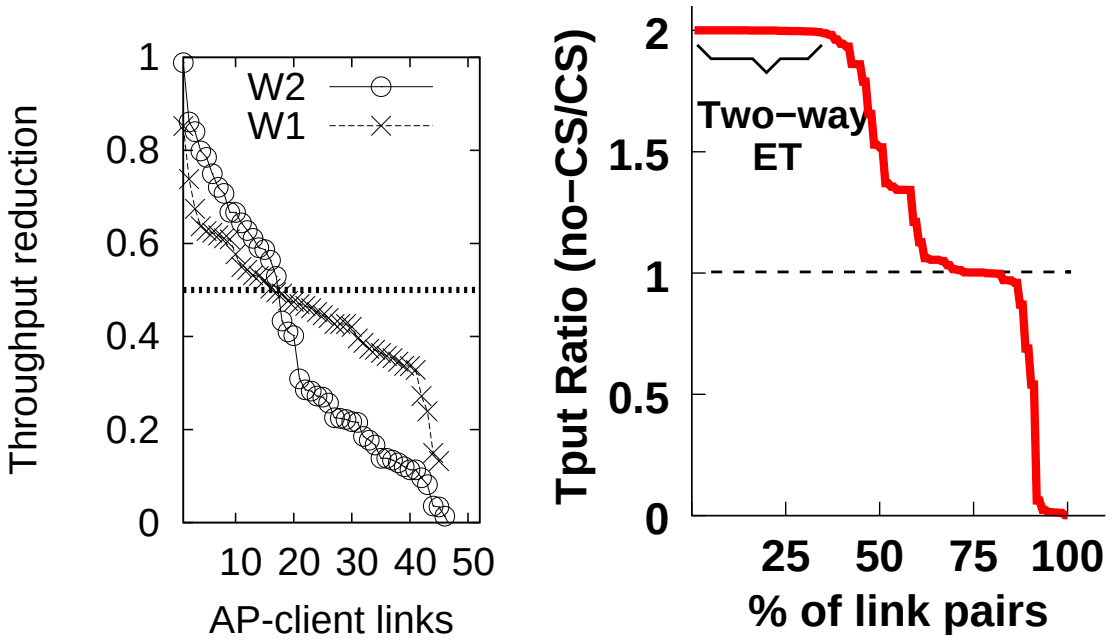
other. However, if two clients C_1 and C_2 get positioned as shown in the figure, and associate to the AP with the strongest signal strength, then X and Y are hidden terminals to clients C_2 and C_1 respectively. One might expect that such close-by APs are likely to be on different 802.11 channels mitigating the entire problem. Unfortunately, as analysis in this section shows there are frequent occurrences of these problems even in carefully-deployed enterprise WLANs, even when adaptive channel assignment schemes are used to mitigate interference. Note that if the enterprise WLAN operates in the 802.11b/g mode then the scarcity of orthogonal frequencies is bound to lead to an imperfect channel assignment.

3.1.2 Quantifying downlink hidden terminals

The Jigsaw effort [44] presented a detailed performance study of a building-wide WLAN in the UCSD campus, consisting of 45 APs and used regularly for Internet access by faculty, staff, and students. It was reported that "co-channel interference from hidden terminals is the likely cause of interference" and for 56% of all interfered traffic, the sender was the AP (i.e., interference was downlink in nature).

Two production WLANs: Motivated by this observation, we conducted our own measurements of two production 802.11b/g WLANs (W_1 and W_2), each in a different building, each serving hundreds of users daily. These WLANs differ from each other in many significant ways as follows. W_1 spans 5 floors of a building and uses 9 APs manufactured by vendor A. The network administrator was responsible for conducting Radio Frequency (RF) site surveys [2] , identifying locations to place the APs, and manually assigning the channel of operation of each AP to minimize interference. Exactly 3 APs were placed on channels 1, 6, and 11 in W_1 to make the level of inter-AP interference relatively low. In contrast, W_2 occupies a single floor of a different building, uses 21 APs manufactured by a different vendor, B, and features a controller in charge of dynamic channel assignment. The number of APs on each channel, thus, varies over time. In W_2 the vendor was responsible for conducting the RF site surveys and making AP placement decisions.

[2] RF site survey is the process of planning and deploying a WLAN. It usually involves measuring wireless coverage that can be achieved from different prospective AP locations in the building and hence helps determine good locations for AP placement while deploying a WLAN.



(a) Impact of hidden terminals

(b) Impact of exposed terminals

Figure 3.1: (a)Throughput reduction due to hidden terminals in production WLANs, W_1 and W_2. Throughput reduction is defined as the ratio of throughput achieved by an AP-Client pair under interference from its strongest interfering AP, to the throughput achieved in isolation. Reduction in excess of 0.5 implies hidden terminals. Severity of hidden terminals increases as throughput reduction approaches 1. (b) Throughput gain for link pairs in CS range (thr without CS/thr with CS). 41% of the link pairs doubled their throughput (two-way exposed terminals), 10% of the link pairs lost throughput (hidden terminals), 20% of the link pairs observe a gain between 1 and 2 (intermittent or one-way exposed terminals). The rest of the links are unaffected.

We placed 45 and 51 nodes in different offices of these two buildings to operate as regular clients to W_1 and W_2 respectively, emulating positions where users typically are located. Once each client associated to a single best AP in each WLAN, we conducted "bandwidth tests" for each pair of AP-client links to identify all occurrences of downlink hidden terminals.

In Fig. 3.1(a) we show the reduction in throughput due to interference of each AP-client link from its strongest AP-client interferer (relative to the throughput achieved when it operates in isolation). A reduction of throughput around 0.5 is expected if the two links are in carrier-sensing range of each other. However, a reduction in excess of 0.5 implies hidden terminals, with the most severe hidden terminals approaching a throughput reduction of 1 (i.e., zero throughput). This is further confirmed by the increased loss rates for these links. We observe that 16 and 17 AP-client links in W_1 and W_2 respectively (out of 45 and 51) experience some form of hidden terminal interference from other APs in the same WLAN. Further, a few links experience severe hidden terminal interference, i.e. reduction in excess of 0.8. In any production WLAN, even if the number of such hidden terminals is small, the persistent, drastic reduction in throughput for these unfortunate clients makes the WLAN unusable for them.

Further experimentation and analysis revealed that such performance degradation would not be prevented even if the Ready-To-Send/Clear-To-Send(RTS/CTS) mechanism were to be enabled. This was primarily because RTS/CTS itself incurred significant airtime overhead [159].

Summarizing, downlink hidden terminals occur relatively infrequently (about 10% of clients) in enterprise WLAN scenarios but *when they occur they do so with devastating consequences for the clients.* Existing mechanisms, like DCF and RTS/CTS, are unable to address the resulting performance degradation.

3.1.3 Quantifying downlink exposed terminals

Unlike hidden terminals, exposed terminal occurrences are hard to observe in production WLAN systems. This is because the only real way to identify if a pair of AP-client links are exposed is by disabling carrier sensing at the APs and testing for loss-free simultaneous communication. Unfortunately, it was not feasible to disable the carrier sensing behavior of the APs in these production WLANs. Hence, we evaluate exposed terminals using our own nodes in Testbed 1. We organize the testbed nodes to mimic the structure of production network W_1 (the closest testbed node to each W_1 AP was chosen to operate as an AP, while the rest of the nodes operated as clients).

Using backlogged UDP traffic we compare the throughput achieved by each pair of links in Testbed 1 with and without CS. We then compute the relative gain obtained in the absence of CS. A value of 1 implies that both experiments led to the same throughput. A value of 2 means that the link doubled its throughput without CS - it was exposed to another link. Fig. 3.1(b) shows the distribution of throughput gain across all link pairs in the network that were in carrier sense range of each other. We observe that around 41% of the links are exposed terminals that could double their throughput. These observations are consistent with observations in CMAP [161] where exposed terminals were found often in their topologies.

Summarizing, DCF mechanisms miss significant opportunities of throughput improvements when exposed terminals occur. While mechanisms such as CMAP [161] can help, they do not meet our objective of requiring no change in 802.11 clients, and hence cannot be implemented independently by an enterprise WLAN vendor. Motivated by these shortcomings of distributed channel access in key hidden and exposed terminal scenarios, we explore the opportunity of leveraging the centralized architecture of enterprise WLANs to solve such hidden and exposed terminal problems.

3.1.4 Why centralization is feasible (and how it can help)?

Enterprise WLANs have a useful construction that facilitates significant gains of centralization without much of its overheads. This is because all traffic to this network typically enters through a single edge router (Fig. 1.1).

Consider the case of two downlink packets (1 and 2) for the two clients C_1 and C_2, associated to APs X and Y respectively. In the traditional DCF mode of operation, the edge router receives these packets and forwards them immediately to the respective APs. Both these packets may get transmitted on the wireless medium simultaneously, leading to interference and packet loss due to the hidden terminal scenario. However, if a wireless controller (co-located at the edge router) realized that such a hidden terminal conflict exists, it might be able to delay packet 2 to a later "time slot," thereby avoiding the collision and packet loss. The key advantage in this design is that by knowing the conflicts in the wireless environment and by observing the previously scheduled

downlink traffic, a controller would have a fair estimate on when to transmit a new downlink packet for interference-free reception. Furthermore, given that a dominant fraction of traffic in an enterprise WLAN is downlink in nature (as observed by analyzing traces of [44, 139, 52] and as reported in [162]), such a mechanism can mitigate a significant fraction of potential interference in the enterprise WLAN and improve the levels of contention in the environment as a whole.

Based on these observations, we first present a simple deterministic central scheduling algorithm (called DET) for managing downlink traffic in an enterprise WLAN, that has some performance advantages, but incurs performance penalty under heavy loads due to the overhead associated with feedback messages from the AP to the controller. In Section 3.3 we attempt to mask such feedback overhead by exploring the design of a speculative scheduling framework (called SPEC) that pipelines packet transmissions to mask delays associated with feedback messages. However, as we show in 3.3.2, such a speculative scheduling framework is difficult to realize in dynamic wireless environments where estimating the the completion times of wireless transmissions is difficult, which can negatively impact the performance of the wireless client under SPEC framework. Finally in Section 3.4 we present the design of an epoch-based scheduling framework that refines DET and SPEC to obtain the CENTAUR system.

3.2 A Simple Deterministic Centralized Scheduling Approach (DET)

Assume that the controller can obtain a *conflict graph,* $G = (L, E)$, where L is the set of (AP-client) transmission links and E is the set of conflict edges defined as $E = (L_i, L_j) \mid L_i, L_j \in L$, such that L_i and L_j interfere with each other. Let us assume that a set of packets $P_1, P_2, \ldots, P_r$ have already been scheduled for transmission but are not yet transmitted. Let $\lambda(P_i) \in L$ denote the link on which packet P_i will be transmitted, $t(P_i)$ the corresponding transmission time, and $\tau(P_i)$ the transmission duration. Now consider a new packet P_{r+1} that arrives at the central controller. We use P to denote the entire packet set $\{P_1, P_2, \ldots, P_{r+1}\}$. For DET we define a simple central scheduling decision where we minimize the time at which the next packet P_{r+1} gets scheduled:

$$\textit{minimize} \quad t(P_{r+1}) \tag{3.1}$$

with the constraint that any two packets to be transmitted on interfering links should not be scheduled together, i.e., if $(\lambda(P_j), \lambda(P_k)) \in E$, then, $P_j, P_k \in P, t(P_j) \geq t(P_k)+\tau(P_k) \bigvee t(P_k) \geq t(P_j) + \tau(P_j)$.

Further, DET is applied to downlink packets only. Uplink packets from clients to APs continue to use the DCF mechanism for channel access. Therefore, uplink transmissions will interfere with centrally computed schedules. We accept this penalty in our design but still expect significant improvements over DCF.

3.2.1 Design and Implementation of DET

We describe the design and working of centralized scheduling framework in detail.

3.2.1.1 Controller

The following are the important components of the controller's logic.

(i) *Conflict graph generator* — This module is responsible for periodically (re-)computing changes to the conflict graph. The conflicts between different AP-client links are detected through small "interference tests" that involved participation of the different APs, as proposed in [23]. Further, we describe a passive mechanism to detect interference (PIE) in Chapter 5, that dynamically generates system wide interference estimates. We also discuss the integration of PIE with the centralized scheduling framework in that chapter. However, in this chapter, we use the conflict graph generated through interference-tests as described above.

(ii) *A packet DAG manager* – This module implements the solution to the optimization objective presented in Equation 3.1, by attempting to schedule each arriving packet in the earliest conflict time slot available, such that it does not interfere with any previously scheduled packets on conflicting links. Packets destined for the wireless clients are enqueued into a Directed Acyclic Graph (DAG), which is maintained by this module. The vertices in the DAG are packets scheduled for future transmission, and there is a directed edge between any two packets that belong to conflicting links. The two core functions of the DAG, packet insertion and removal are discussed next.

Algorithm 1 DAG : packet insertion

$L = \{L_1, L_2, L_3, \ldots, L_n\}$ is the set of n transmission links in the system

$E = \{(L_i, L_j) \mid L_i, L_j \in L$, such that L_i and L_j conflict with each other$\}$

$G = (L, E)$, where L is the set of transmission links and E is the set of conflict edges defined above

$P = P_1, P_2, P_3, \ldots$ is the stream of packets arriving at the router, where each packet P_i is associated to a link L_j

We define $\mathcal{S}$ as the collection of sets $\mathcal{S} = \{S_1, S_2, \ldots\}$ where each S_i is a set of packets from the DAG, that are transmitted in slot i. For any i, j, $i \neq j \Rightarrow S_i \cap S_j = \phi$. Also if $i < j$, packets in slot S_i are transmitted before packets in slot S_j.

We define a layered acyclic graph $DAG = (\cup S_i, E')$, where $S_i \in \mathcal{S}$ and E' is the set of directed edges defined as: $E' = \{(P_m, P_n) \mid$ iff $P_m \in S_i, P_n in S_i + 1$, and P_m, P_n is to be transmitted on links L_t, L_u respectively, and $\exists$ $(L_t, L_u) \in E$, *i.e.* L_t and L_u conflict$\}$

Procedure $DAG - insert\ (P_i, G, DAG, \mathcal{S})$:

$k = \min_j S_j$ s.t. $\forall P_m \in S_j$ and P_m, P_i is associated to L_t, L_u respectively, $\nexists (L_t, L_u) \in E$

if no j exists

Create new $S' = \{P_i\}$ with index $|\mathcal{S}| + 1$

$\mathcal{S} \leftarrow \mathcal{S} \cup S'$

end if

for all $P_j \in S_{k-1}$ **do**

$E' \leftarrow E' \cup \{(P_j, P_i) | P_j, P_i$ is associated to L_t, L_u respectively, and $\exists (L_t, L_u) \in E\}$

$indegree(P_i) \leftarrow indegree(P_i) + 1$

end for

for all $P_j \in S_{k+1}$ **do**

$E' \leftarrow E' \cup \{(P_i, P_j) | P_i, P_j$ is associated to L_t, L_u respectively, and $\exists (L_t, L_u) \in E\}$

$indegree(P_j) \leftarrow indegree(P_j) + 1$

end for

$V \leftarrow V \cup \{P_i\}$

if $indegree(P_i) = 0$

$SendOut(P_i)$

end if

Algorithm 2 DAG : packet removal

For initialization see 1.

Procedure $DAG - remove\ (P_i, DAG, \mathcal{S})$:

Define child set $C(P_i) = \{P_j \mid (P_i, P_j) \in E'\}$

$P \leftarrow P \setminus \{P_i\}$

$E' \leftarrow E' \setminus \{(P_i, P_j) \mid \forall P_j \in C(P_i)\}$

for all $P_j \in C(P_i)$ **do**

$indegree(P_j) \leftarrow indegree(P_j) - 1$

if $indegree(P_j) = 0$

$SendOut(P_j)$

if $P_j \in S_k$, $S_k \leftarrow S_k \setminus \{P_j\}$

if $S_k = \phi$, $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_k\}$

end if

end for

Procedure $DAG - sendOut\ (P_i)$:

if $Speculative$

$Timer(T(P_i))$, where $T(P_i)$ is the expected

transmission time of packet P_i

end if

Put packet P_i on Ethernet

Procedure $DAG - timer\ T(P_i)$:

Wait for time $T(P_i)$

$DAG - remove\ (P_i, DAG)$

- **Packet Insertion:** The psuedocode for packet insertion in the DAG is shown in Algorithm 1. As shown in the code, whenever a packet X is received, it is inserted in the DAG and there is a edge from packet X to all the other packets that are already scheduled on conflicting links (same AP packets are considered as an conflict as they cannot be scheduled simultaneously). We also considered alternative approaches such as K-packet lookahead, where we can simultaneous schedule K packets together in an attempt to minimize the overall transmission time for all the active clients in the system. However, solving such an optimization problem

can be difficult at the small time scales (microsecond granularity) at which a which a typical scheduling formulation needs to operate for high speed wireless networks.

- **Packet Removal:** Semantics of the Traffic DAG enable the scheduler to quickly identify the next set of packets that should be scheduled if a particular packet completes its transmission. The psuedocode for packet removal in the DAG is shown in Algorithm 2. When an AP successfully transmits a frame on the wireless medium, it issues a special *wired acknowledgment* to the controller indicating the completion of that packet. On receiving this wired acknowledgment, the corresponding packet is removed from the DAG along with all its incoming edges, freeing up other potentially conflicting packets for transmission. Hence, any packet enqueued in the DAG with no outgoing edge is scheduled for transmission immediately. This helps the scheduler to determine the next set of packets to be scheduled in a bounded $O(n)$ time, where n is the number of wireless clients in the system. This property of the DAG prevents exhaustive search for identifying the next set of packets to be scheduled.

(iii) *Link Statistics Manager* – In order to estimate the amount of time required for completing a packet transmission on a link, the centralized scheduler maintains a link statistics table, that records the average wireless transmission time and delivery probability for different links in the system. Link statistics are updated by the wired feedback from the AP. The estimate of wireless transmission times is maintained as an Exponentially Weighted Moving Average(EWMA) [3] with a weight γ assigned to the most recent value of transmission time reported by the wired feedback (and $1 - \gamma$ to the historical value in the table). Our experimentation shows that using a gamma value of 0.95 works well for our system. Using this high value of γ can be attributed to the bursty nature of wireless links, due to which the most recent value of transmission time is a good indicator of link quality. This table is build over the period of time the link is active.

(iv) *High Resolution Timer* – Scheduling in wireless networks requires microsecond granularity to efficiently schedule packets at the precise time instants. The most commonly available kernel

[3] EWMA is used with time series data to smooth out short-term fluctuations and highlight long-term trends [50]. In EWMA, the weighting of older data points decrease exponentially to reflect changes in the data pattern. The exact weight given to the old data points determines how quickly the moving average changes with the change in data patterns.

timers have a resolution of $1ms$, which is inadequate for our scheduling framework, that may need to schedule multiple packets within $1ms$ window, especially so when the packet sizes are small and data rates are high. In order to achieve microsecond granularity for centralized scheduling, we made use of High-Resolution (HR) timers [10] available for the 2.6.20 version of the Linux kernel. HR timers are part of an effort to implement a separate kernel timer sub-system where timing events are decoupled from the rest of the kernel functionality to ensure high-precision timing. Figure 3.2 shows the error in HR-timer accuracy for 10000 instances of timer expiration under heavy and light loads at the wireless controller. Error in HR-timer accuracy indicates the gap between the time for which the HR-timer was scheduled and the actual time after which it expired. As shown in the Figure, HR-timer is accurate to the microsecond granularity and the error remains within $20\mu s$ for majority of cases under both heavy and light traffic loads at the controller.

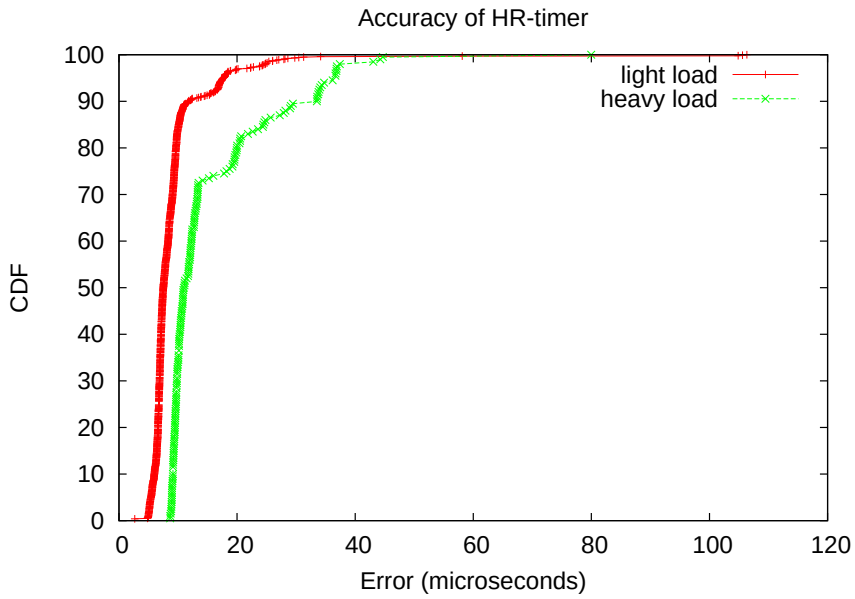


Figure 3.2: HR-timer accuracy for heavy and light loads. Error is defined as the offset between the time for which the HR-timer was scheduled and the actual time after which it expired. We compute error over 10000 instances of timer expiration. HR Timer is more accurate at light loads, where in about 90% of the cases the error is within $20\mu s$. In heavy load scenarios, in about 90% of the cases, the error is less than $40\mu s$, which is still reasonable for our scheduling purposes.

Putting it all together: We implemented the centralized scheduler on a standard Linux PC (3.33 GHz dual core Pentium IV, 2 GB DRAM). Further, the scheduling functionality is implemented as a kernel module that hooks into the Ethernet driver of the host machine. The module intercepts packets being sent out on the wire and inserts them into DAG for appropriate queueing. The controller was implemented using 3,000 lines of C code and a few hundred lines of Perl scripts for generating and updating the conflict graph. The scheduler interfaces with the conflict graph generation daemon running in user-space through sysctl calls, where it periodically requests an up-to-date version of the conflict graph for scheduling purposes.

3.2.1.2 Access Point

Our AP runs on a Soekris 4826 box (266MHz MIPS, 32MB RAM) running a customized version of Linux operating system [18]. We implemented a direct *driver-to-driver communication path* to allow packets received on the wired interface to be immediately forwarded to the wireless interface, bypassing the kernel network queue. This also helped us minimize unpredictable delays from other in-kernel events. We also instrumented the Intel wireless device driver and firmware (ipw2200 [13]) (i) to report retransmissions, contention window size and data rate used for each frame, and (ii) to issue the wired ACK to the controller for each successful frame transmission reported by the firmware. Such wired acknowledgments allow the centralized controller to manage accurate statistics for transmission times on different links in the system.

3.2.2 Where DET helps and where it does not ?

To evaluate DET's functionality, we experimented using three different simple canonical topologies involving two AP-client links, where the downlink paths are: i) hidden terminals (HT), ii) exposed terminals (ET), and (iii) normally interfering, but neither hidden nor exposed terminals (non-HT/non-ET). Figure 3.3 shows the throughput gains of DET (normalized to DCF) for these three topologies, under low, medium and high traffic loads on the two downlinks. For the HT case, DET achieves 2-4$\times$ throughput gains over DCF in medium and high loads. Unfortunately,

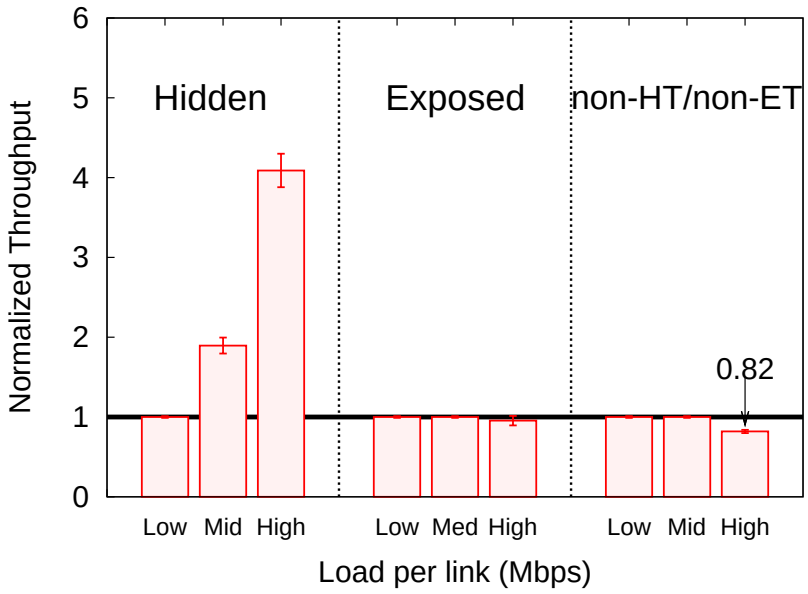


Figure 3.3: Throughput achieved using DET (normalized to DCF throughputs) on a two-link topology for three different scenarios of HT, ET and non-HT/non-ET in a 802.11g wireless network. Low, Mid and High represent loads of 1.2 Mbps, 2.4 Mbps and 6 Mbps respectively. Performance gains of DET over DCF increases with increase in traffic load for HT and ET, while the throughput decreases for non-HT/non-ET links under heavy loads due to path latencies.

DET provides no advantage for ET and normal terminal cases. In fact, there is a slight loss in performance when compared to DCF in the normal case, especially under high loads.

Limitations of DET : The above results made clear that even a simple centralized scheduling technique can provide significant performance gains when downlink hidden terminals occur. However, the performance penalties in the normal interference case, and the lack of gains in exposed terminals need further investigation. It turns out that much of this inefficiency stems from overheads and inaccuracies in scheduling downlink packets from the controller. Through careful instrumentation of the Atheros wireless driver (Testbed 1) and the Intel ipw2200 wireless driver (Testbed 2), we obtained these delays for different parts of the downlink path (Figure 3.4). Despite our considerable effort to minimize the latency for the wired acknowledgment (Wired-delay) by optimizing the scheduler and the driver-to-driver (D2D) communication path, we found this delay

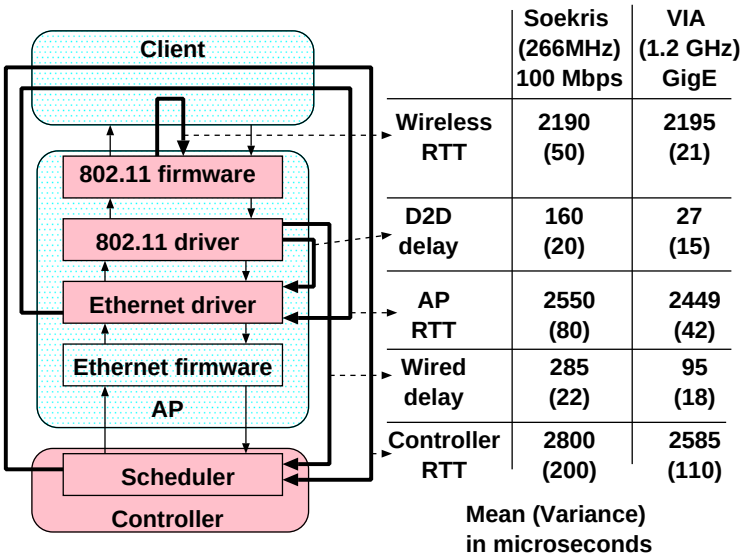


Figure 3.4: Latencies on Controller-AP-client path that impacts centralized scheduling decisions. Note that Controller RTT = Wired delay + AP RTT .

to be 285 μs on the Soekris/100 Mbps Ethernet platform and about 92 μs on the VIA/GigEthernet platform. Such delays can lead to wasted airtime for the DET scheme.

In the next section, we present a speculative scheduling framework, SPEC, that can mask these delays by *carefully pipelining packets* to APs, in anticipation of current conflicts disappearing at a known time in the future. The scheduler maintains statistics (using Link statistics manager) on prior delays on each path to predict the time when prior frames are likely to have been successfully transmitted, so that the next set of frames can be released to their respective APs.

3.3 Speculative centralized scheduling (SPEC)

Ideally, a central scheduler should know exactly when each frame is successfully transmitted by each AP, so that it can schedule a set of frames that are guaranteed not to collide with each other. A speculative scheduler, instead, predicts the frame completion times based on past history, and schedules a set of frames before receiving acknowledgments of previously scheduled frames from the APs. On receiving a frame, APs immediately transmit it on the wireless interface, modulo

deferrals due to a busy wireless medium as specified in the 802.11 standards. Note that by its very nature, speculation errors are possible in SPEC. Thus, we also permit APs to perform backoffs to resolve collisions caused by speculation errors. However, we limit the degree of backoff by setting the backoff counter to the minimum value permitted by the 802.11 standard.

3.3.1 Working of SPEC

We now discuss how a speculative scheduler deals with uncertainties in wireless frame transmissions. We decouple this uncertainty into two components:

- Uncertainty in the wireless transmission time of a frame, including delays due to potential re-transmissions (*Wireless-RTT-ReTx*). APs measure the *Wireless-RTT-ReTx* corresponding to a particular frame size and PHY layer data transmission rate, and piggyback these measurements on the acknowledgments sent to the controller over the wired link.

- Uncertainty in the one-way delay between the controller and the APs (*Wired-delay*). These measurements are obtained from the round trip time between the AP and the controller through appropriate timestamps piggybacked on wired acknowledgments.

The controller maintains a history of these values and uses this to compute an exponentially weighted moving average of the mean transmission time. In SPEC, a frame that should finish its wireless transmission at time t is released by the scheduler at time $t - \delta$, where,

$$\begin{aligned} \delta &= \mu(\text{Wireless-RTT-ReTx}) + \beta \cdot \sigma(\text{Wireless-RTT-ReTx}) \\ &\quad + \mu(\text{Wired-delay}) + \beta \cdot \sigma(\text{Wired-delay}) \end{aligned} \tag{3.2}$$

where μ and σ refer to EWMA and mean deviation of the error, and β is a constant that determines the degree of aggressiveness in releasing frames. Experimental tuning led us to choose $\beta = 1$ because speculation is useful only if applied in a relatively aggressive manner. Note that the controller is always assured of an eventual successful wireless transmission, because of the underlying use of re-transmission mechanisms that are part of the 802.11 standards. Hence, being aggressive does not come at the expense of correctness.

3.3.2 Evaluating SPEC

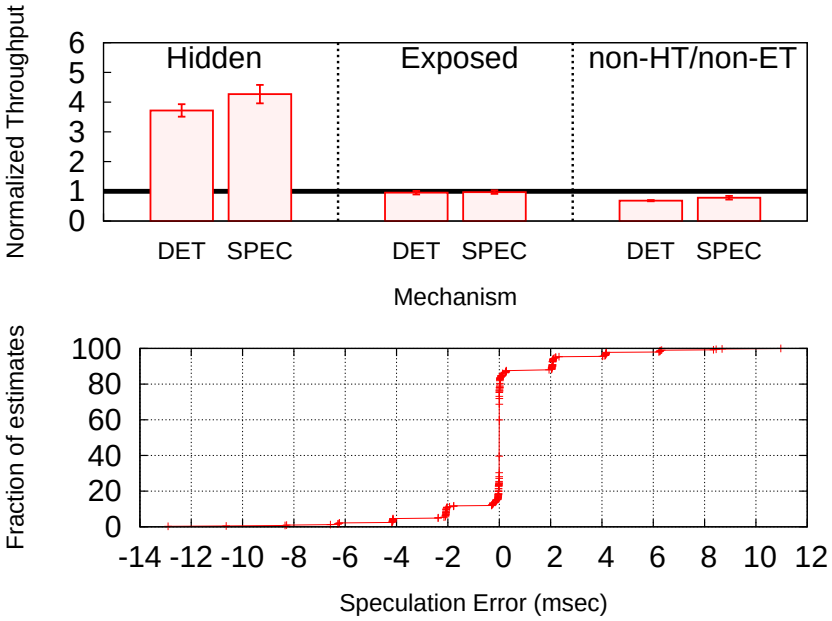


Figure 3.5: Throughput achieved using SPEC and DET (normalized to DCF throughputs) on a two-link topology for three different scenarios of HT, ET and non-HT/non-ET. SPEC outperforms DET for HT scenarios, but is unable to provide any gains for ET and non-HT/non-ET scenarios.

Figure 3.5 shows the throughput gains of SPEC and DET (normalized to DCF) for the three canonical topologies described in Section 3.2.2. We present results for the high load scenario, where the impact of scheduling overheads is most prominent. As shown in the Figure, for the HT case, SPEC outperforms DET as it is able to mask the wired acknowledgment delays through pipelining. However, SPEC's performance is similar to DET for both ET and normal links, providing no gains in these scenarios. The inability of SPEC to improve upon DCF in this scenario brings up multiple issues. Although SPEC is an improvement over DET, it is not an optimal centralized scheduling algorithm. The bottom plot of Figure 3.5 shows the speculation errors, defined as the time difference between the predicted duration of a wireless frame transmission and the actual duration of a wireless frame transmission on a link using a frame size of 1400 bytes and data rate of 6 Mbps. As shown in the Figure, SPEC mis-speculates about 20% of the time, usually in multiples

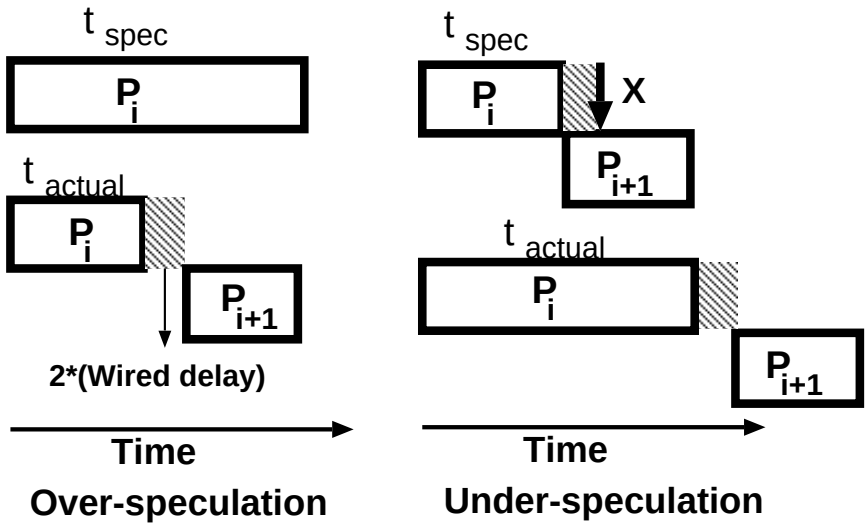


Figure 3.6: Penalty for over and under speculation in SPEC. In case of over-speculation, penalty is bounded by $min(2 \times Wired_delay, t_spec - t_actual)$. While in under-speculation, it is bounded by $max(t_{transmission}(P_i), t_{transmission}(P_{i+1})) + 2 \times Wired_{delay}$, where $t_{transmission}(P_i)$ refers to the transmission duration for packet P_1 excluding retransmissions.

of $2ms$. This is the same as the average Wireless-RTT on the link under study and points to mostly under-counting or over-counting the number of re-transmissions while estimating the transmission time for the next frame. Depending whether the scheduler under-estimates or over-estimates the transmission time of a scheduled packet, it can impact the performance of SPEC differently as described below:

3.3.2.1 Over-speculation

As shown in Figure 3.6, when the scheduler overestimates the total transmission time of the scheduled packet, then the next set of conflicting packets will only be released by a wired ack sent by the AP. In this scenario, the scheduler will not be able to pipeline the packets and incur the penalty associated with the wired acknowledgment mechanism. The total overhead in this scenario is $min(2 \times Wired_{delay}, t_{spec} - t_{actual})$. This delay is equivalent to the per packet overhead in DET framework.

3.3.2.2 Under-speculation

In case of under-speculation, the packet transmission takes longer than expected and hence the scheduler can incorrectly schedule the next conflicting packet (P_{i+1}) before the transmission of the previously scheduled packet (P_i) is over. However, the absence of a wired acknowledgment from the AP transmitting packet (P_i) will eventually inform the scheduler that it under estimated the total transmission time for packet P_i. In this case, the scheduler will squelch the scheduled conflicting packet (P_{i+1}) by sending a wired message to the AP on which the conflicting packet is scheduled. Then the scheduler waits for a wired acknowledgment from the AP transmitting packet (P_i) to restart speculative scheduling. Under-estimation can potentially lead to collision in one time slot in which the conflicting packets were scheduled incorrectly by SPEC. Hence the total penalty of such over-speculation could be approximated as $max(t_{transmission}(P_i), t_{transmission}(P_{i+1})) + 2 \times Wired_{delay}$, where $t_{transmission}(P_i)$ is the expected air time of packet P_i for one transmission attempt.

It is clear that SPEC can incur significant performance penalty in dynamic wireless environments, where average transmission time for different clients is difficult to predict. Motivated by our experience with DET and SPEC we present the design and implementation of CENTAUR that overcomes the deficiencies of both DET and SPEC by effectively using a combination of techniques — *epoch-based scheduling, fixed backoffs, packet staggering,* and *a hybrid data path*. Through a combination of all these techniques, CENTAUR achieves throughput gains for exposed as well as hidden terminals scenarios, without sacrificing performance in more common cases.

3.4 CENTAUR Design

CENTAUR incorporates the basic scheduling approach of DET and augments it to mitigate some of its main limitations. We describe this by defining the three main objectives of CENTAUR beyond what DET already provides. They are to: (i) exploit exposed terminals without disabling carrier sensing, (ii) amortize overheads in the scheduling process, and (iii) allow co-existence of

uplink as well as non-enterprise traffic by combining our centralization approach with DCF. We describe how CENTAUR meets each objective, in turn.

3.4.1 Exploiting exposed terminals without disabling carrier sensing

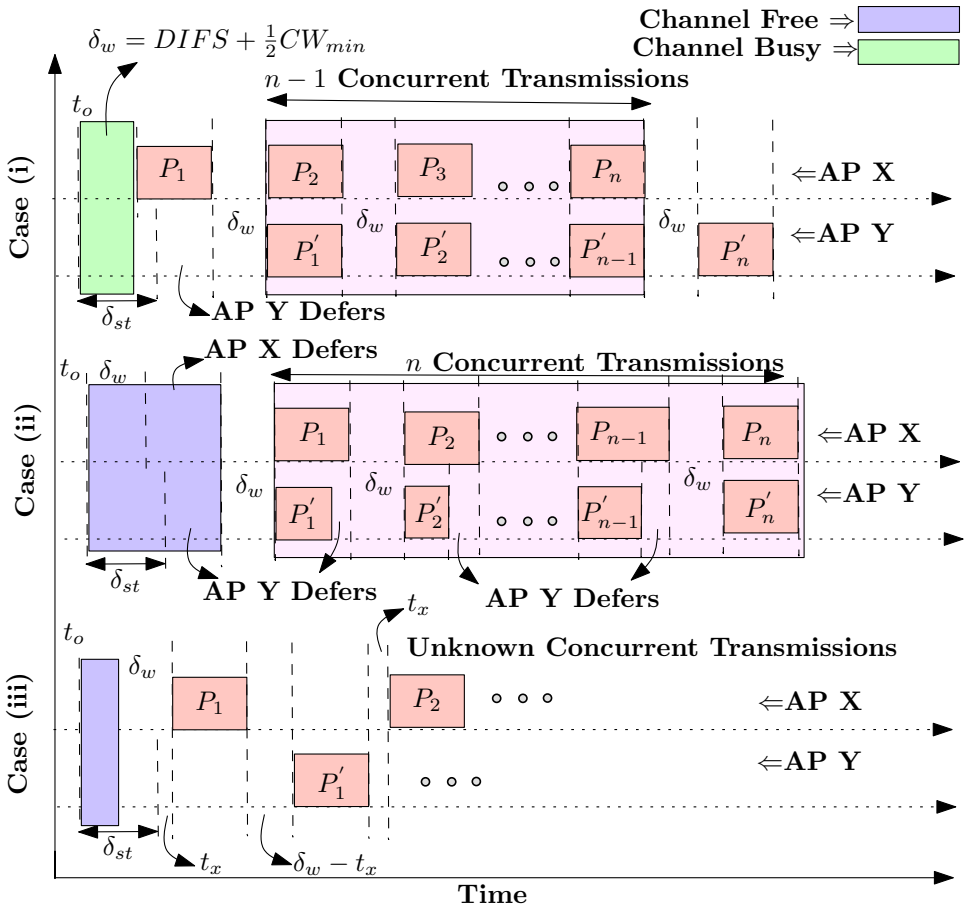


Figure 3.7: Staggering packets by a time δ_{st} increases transmission concurrency. Cases (i) and (ii) illustrate the scenarios where the channel state remains the same for the back-off duration δ_w therefore synchronizing the transmissions. Case (iii) depicts the scenario where the gains can be unpredictable.

A typical way to allow simultaneous communication over exposed terminal links is to disable carrier sensing. However, disabling carrier sensing for all nodes is particularly dangerous, as it might increase the possibilities of interference. A more intelligent approach is to implement *selective carrier sensing* wherein a transmitter would carrier sense (and therefore back-off) for

non-ET links but continue with the transmission for ET links. CMAP [161] is an example of such an approach. However, as the authors discuss in [161], the design of such a mechanism either requires software level modifications for both APs and clients, or it requires a change in the existing 802.11 protocol standard. In keeping with our design goal of requiring no changes at clients or in the underlying 802.11 standard, we achieve simultaneous communication over exposed terminals using an alternate approach as follows: (i) maintain carrier sensing, (ii) use fixed back-offs, and (iii) stagger packets destined to exposed APs. We describe the use of (ii) and (iii) in detail, next.

Fixing back-off intervals

Consider a scenario where n packets are enqueued at each of the APs X ($P_1 \dots P_n$) and Y ($P_1^{'} \dots P_n^{'}$) which are exposed terminals. For simplicity, assume that each packet transmission takes time t_p. In case of DCF, the total transmission time required would be $2nt_p$, excluding backoff and idle times. Now consider a case where back-off at both APs is fixed to some value bo. Each of the APs will now only defer for a fixed amount of time, $\delta_w = DIFS + bo$ before transmitting a packet. If we assume a simplistic scenario where both the APs start contending for the medium at the same time, and are successful in transmitting their first packet at the exact same time, then the transmission concurrency on these ET links is doubled. After the first packet transmission, both the APs will sense the carrier to be free for a period of δ_w, and then transmit their second packet at the same time. Thus, all packet transmissions after the first packet are synchronized achieving the effect of disabling the carrier sense[4]. In reality, however, the first packet transmissions are highly unlikely to be synchronized due to wired jitter. In this case, the two APs will get out of sync, and due to carrier sensing, will not be able to transmit simultaneously in the same slot. Therefore, we use packet staggering, which requires delaying the first packet of the two APs relative to each other such that the following packets of both the APs are perfectly synchronized. Next we explain this process in detail.

[4]The nodes will indeed carrier sense each other but they won't defer since they will be perfectly synchronized in their transmissions.

Packet staggering

Staggering packets P_1 and $P_1^{'}$ by $\delta_{st} > \delta_w$ results in one the three cases shown in Fig. 3.7: (i) at t_o, AP X starts contending for transmitting P_1 and the channel remains free during the duration δ_w. In this case, AP X transmits the first packet while AP Y defers its transmission due to carrier sense (AP Y had to wait longer to receive its packet due to the fact that $\delta_{st} > \delta_w$). After the first packet transmission, both APs will sense the carrier to be free for a period of δ_w, and then transmit the packets at the same time. Thus, all packet transmissions after the first packet are synchronized. In this case, the total time for transmission is $(n+1)t_p$ ($n-1$ packets are transmitted concurrently and two out of sync), resulting in a throughput gain of $\frac{2n}{n+1}$ (Fig. 3.7(i)) (ii) the channel remains busy during the duration δ_w, in which case all the n packets are transmitted concurrently, resulting in a gain of 2 (Fig. 3.7(ii)) (iii) the channel is busy only during some part of the duration δ_w, which results in unpredictable gains as the transmissions of AP X and AP Y may not be synchronized during the entire epoch (Fig. 3.7(iii)). In CENTAUR P_1 and $P_1^{'}$ are staggered by an amount $\delta_{st} = \delta_w + \gamma \cdot (wired\_jitter)$. We found that the value of $\gamma = 1$ gave the best performance in our testbed. Note that the transmissions in cases (i) and (ii) will be synchronized even when the packet sizes differ for the same link or across links. The effectiveness of packet staggering will also depend on the amount of unscheduled traffic in the network and its interaction with the exposed links. In practice, we show that it leads to remarkable gains over generic traffic mixes (Section 3.6).

Fairness

In order to contend fairly with other DCF traffic, APs in CENTAUR use a fixed back-off value of $bo = \frac{1}{2}CW_{min}$ which is the average amount of time other transmitters using DCF would spend in deferral. Indeed, experimental results confirm such a property in CENTAUR. The further lack of exponential back-off is not a concern since conflicting links are by design scheduled in different epochs, and are not going to be active simultaneously.

Algorithm 3 CENTAUR : Downlink processing

INPUTS: epoch time (t_{ep}), conflict graph $G = (L, E)$

$max\_ep \leftarrow 0$, $curr\_ep \leftarrow 0$ ***//Initialize***

Procedure ProcessDownlinkPacket(P_i):

for each epoch $ep[j]$ in $ep[curr\_ep \ldots max\_ep]$

if *canFit* $(P_i, ep[j])$ **then**

addPacket$(ep[j], P_i)$; return;

max_ep++; ***addPacket***$(ep[max\_ep], P_i)$

Procedure addPacket($ep[j], P_i$):

$ep[j].links = ep[j].links \cup \lambda(P_i)$

if $j \neq curr\_ep$

$ep[j][\lambda(P_i)].txfill += \tau(P_i)$

else

$ep[j][\lambda(P_i)].txfill = max(ep[j][\lambda(P_i)].txfill,$

$curr\_time - ep[j].start\_time) + \tau(P_i)$

$ep[j][\lambda(P_i)].lastack = P_i$; $ep[j][\lambda(P_i)].enqueue(P_i)$

Procedure canFit($P_i, ep[j]$):

if $\lambda(P_i) \in ep[j].links$ **or** $((l, \lambda(P_i)) \notin E \; \forall \; l \in ep[j].links)$ **then**

if $j \neq curr\_ep$ **then**

if $ep[j][\lambda(P_i)].txfill + \tau(P_i) \leq t_{ep}$ **then**

return true

else

if $\tau(P_i) + max(ep[j][\lambda(P_i)].txfill,$

$curr\_time - ep[j].start\_time) \leq t_{ep}$ **then**

return true

return false

3.4.2 Amortizing overhead using epochs

Per-packet scheduling in DET proved to be sub-optimal in generic topologies (without a large number of hidden terminals) due to the delay overhead between the controller and the APs. In essence, DET releases a packet to its intended AP at the time it can get transmitted into the air. The variability in the amount of time it takes for that packet to actually arrive at the AP and finally to the client is what leads to inefficiencies - thus disturbing the inherent timing of the derived schedule.

Epoch-based scheduling: Inefficiencies, described above, can be reduced if the schedule operates on *epochs*, periods of time when packets are transmitted in batches. As long as the batch transmission duration, i.e. epoch, is sufficiently greater than the wired delay variability between the APs and the controller, slight synchronization errors are unlikely to have as significant an effect.

CENTAUR, however, does not only use epochs to amortize the scheduling cost, but also to take advantage of exposed links [5]. Epoch-based scheduling has an important parameter — the time duration of an epoch. This parameter captures an inherent tradeoff between scheduling efficiency and increase in latency experienced by scheduled packets. In particular, the larger the epoch duration, the greater is the scheduling efficiency, but the higher is the path latency experienced by individual packets. After significant parameter sensitivity testing (some results in Section 3.5), we realized that an epoch duration in excess of 5 ms was sufficient to achieve good scheduling efficiency without adding a high amount of packet latency. To be conservative, we used a default epoch duration of 10 ms in our implementation.

3.4.3 Handling downlink non-HT/non-ET, uplink, and non-enterprise traffic

As our experiments will show, the scheduling approach is particularly beneficial to hidden and exposed terminal traffic in the downlink path, while scheduling traffic to non-hidden and non-exposed terminals in the downlink does not provide much gain.

Hybrid data path: To relieve the load on the scheduling system, we partition all downlink traffic into two parts — traffic to hidden and exposed terminals, which gets *scheduled,* and all other traffic, which is *unscheduled.* As Fig. 3.8 shows, when downlink packets arrive at the controller for hidden

[5] Packet staggering is effective if packets are transmitted in batches, which is possible under epoch based scheduling

or exposed terminals, they get forwarded to the scheduler. All remaining packets are forwarded directly to the APs to be transmitted using the standard DCF mechanisms with carrier sensing and backoffs. Further, all uplink and non-enterprise traffic is, also, unscheduled and contends for the channel using DCF. Since our scheduled traffic continues to use the carrier sensing mechanism, our scheduled traffic can co-exist with all unscheduled traffic. We illustrate this further in Sections 3.5 and 3.6.

3.4.4 Putting it all together

Summarizing, CENTAUR differs from DET in multiple important ways. In particular, CENTAUR includes packet staggering, fixed backoffs, epoch scheduling, as well as the hybrid data path. When a downlink packet arrives, CENTAUR decides first whether to schedule the packet or not. In our implementation we use a generic epoch-based scheduler, whose logic is presented in the pseudo code shown in Algorithms 1 and 2. Whenever a downlink packet is forwarded to the scheduler, it enqueues the packet into one of the epochs, based on the inputs from the conflict graph (G(L,E)), epoch time (t_{ep}) and ETT of the link ($\tau(P_i)$). An epoch therefore consists of multiple packets for each link which are forwarded to the respective AP at the beginning of the epoch. Note that the packets belonging to HT links are packed in separate epochs, thereby ensuring robust conflict resolution. When dealing with ET links, CENTAUR uses packet staggering to increase the possibility of concurrent transmissions. The controller schedules the packets of the next epoch, after receiving the wired acknowledgments of the last packet scheduled on each of the links in the current epoch (Algorithm 2). Measurements on the conflict graph are taken periodically using the micro-probing technique [23] which has minimal overhead. Our evaluation shows performance gains of CENTAUR in spite of such overheads.

3.5 CENTAUR Microbenchmarks

To evaluate whether our design of CENTAUR meets our goals, we first present a few micro-benchmarks on targeted scenarios.

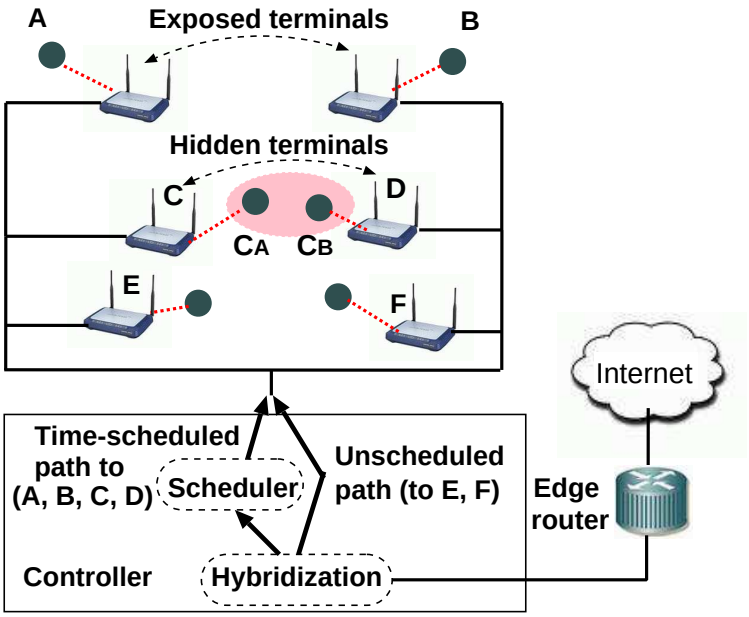


Figure 3.8: Overview of the **CENTAUR** hybrid data path.

Algorithm 4 CENTAUR : Feedback processing

Procedure StartNextEpoch():

For each link in $ep[curr\_ep].links$ **do**

if link is ET, use *staggering* to forward packets

else forward packets to AP

Procedure ProcessWiredAck(ack):

Update the ETT for link $\lambda(ack.id)$

if got lastacks for all $ep[curr\_ep].links$ **then**

$curr\_ep++$; $ep[curr\_ep].start\_time = curr\_time$

StartNextEpoch();

3.5.1 CENTAUR and hidden and exposed terminals

To test the ability of CENTAUR to mitigate hidden and exposed terminal interference, we created topologies with all hidden and all exposed terminal links — 21 and 30 respectively. We describe the setup and metrics for comparing different schemes in the microbenchmarks.

Setup: We imposed a high downlink traffic load across all these links to keep them saturated and observed how various versions of CENTAUR compared to DCF, both with and without RTS-CTS. For precise comparison, we fixed the PHY rate at 6 Mbps, packet size to 1440 bytes, and ran each scenario 10 times for 3 minutes each.

Metrics: We compare the total throughput acheived by each mechanism under different scenarios. Further, in order to understand the ability of each mechnaism to provide fairness, we also compute fairness index for the throughput distribution achieved with different mechanisms. We use Jain's fairness Index [76] to evaluate the fairness provided by individual schemes. The Jain's Fairness Index for a thtoughput vector $\vec{T} = (t_1, t_2, ..., t_n)$ is given by

$$\frac{(\sum_{i=1}^{n} t_i)^2}{n \cdot \sum_{i=1}^{n} l_i^2}$$

Intuitively, Jain's Fairness Index of a throughput vector is 1 if it is perfectly fair (i.e., all links achieve equal throughput), and is $\frac{1}{n}$ if it is completely unfair (i.e., only one link gets full throughput and rest of the links starve).

ET-only topology

Figure 3.9(left) shows the distribution of throughput across different exposed terminal links found in the testbed. CENTAUR with a epoch duration in excess of 5 ms is far superior to DCF (median throughput increases from 2.4 Mbps to 4.6 Mbps). In fact, the throughput of all links in the topology improve with CENTAUR. Only CENTAUR with a 2 ms epoch is unable to leverage the gains, because of scheduling inaccuracies at the small epoch size. Disabling carrier sensing completely performs slightly better than CENTAUR. However, a full and robust implementation of such an approach will require client-side changes (as in CMAP [161]) and does not meet our goals.

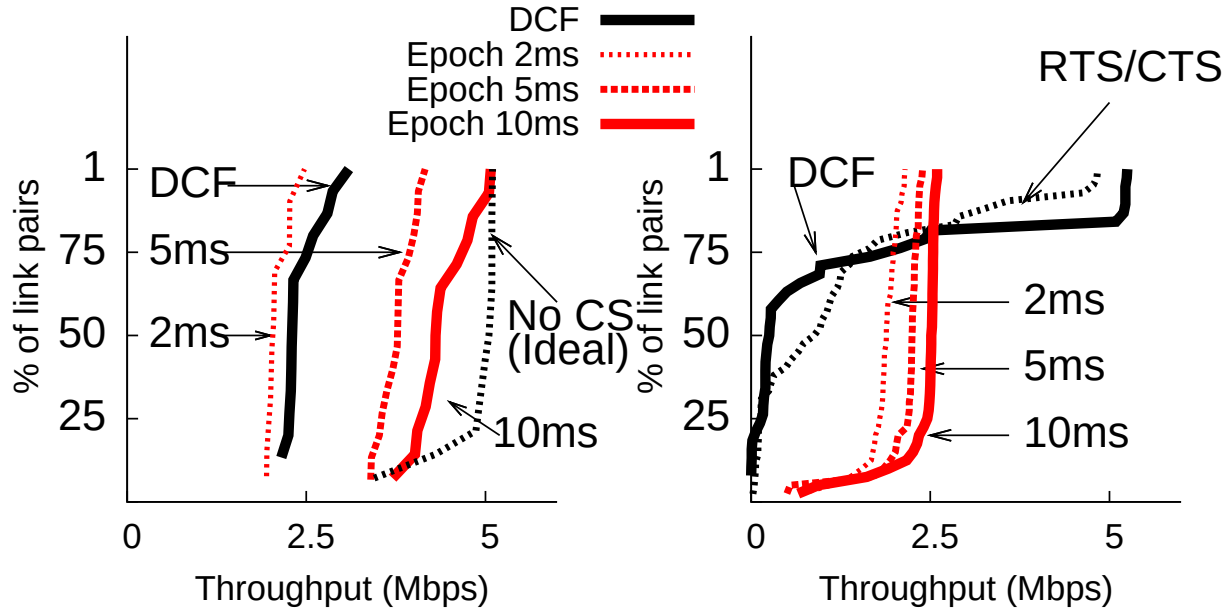


Figure 3.9: Distribution of throughputs achieved by exposed (left) and hidden (right) link pairs under different access mechanisms. An epoch period of 2ms is equivalent to per packet scheduling. (Testbed 1)

HT-only topology

Figure 3.9(right) shows that all variations of CENTAUR (with different epoch times) help mitigate the hidden terminal problems. While DCF has a large number of underperforming links (median throughput of 0.2 Mbps without RTS-CTS and 0.8 Mbps with RTS-CTS), CENTAUR with 10 ms epoch has a median throughput of 2.5 Mbps (a factor of 10 and 3 over the two DCF scenarios). The increase in throughput for the hidden terminal links, naturally reduces the throughput of the remaining links. In fact, CENTAUR results in a value of 0.94 for Jain's fairness index, while DCF and RTS/CTS achieve 0.33 and 0.51 respectively.

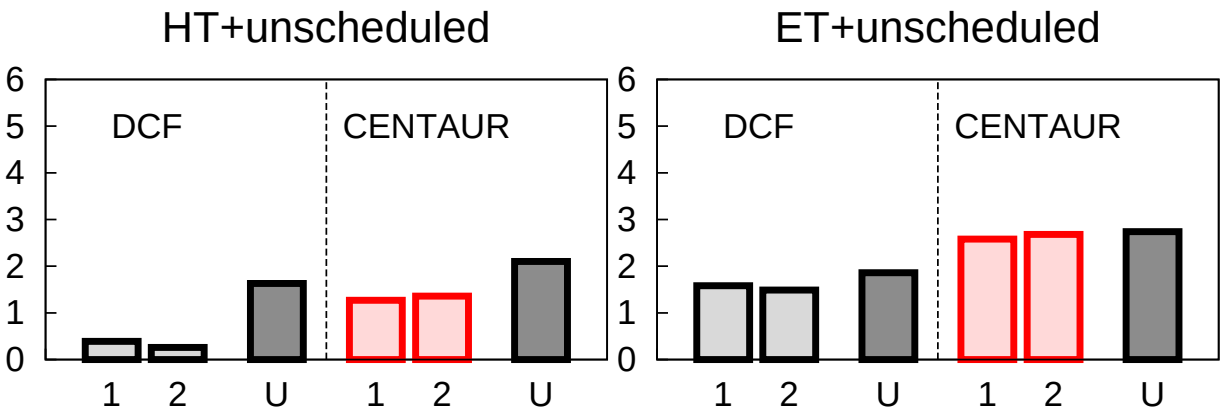


Figure 3.10: CENTAUR throughput in the presence of unscheduled traffic(Mbps, Testbed 1). Both scheduled and unscheduled link performance improves.

3.5.2 Co-existence with unscheduled/uplink traffic

Success of CENTAUR will require efficient co-existence of the downlink scheduled traffic with all unscheduled traffic, including uplink traffic. Therefore, in initial targeted experiments, we created two-link hidden and exposed terminal scenarios (clients 1 and 2), and augmented it with a third client which was responsible for sending continuous uplink traffic (U). There are multiple possible configurations of the client U depending on its exact interference relationships with the clients 1 and 2. We evaluated all possible variations of these scenarios, and summarize our observations in Figure 3.10. The left plot shows the performance of CENTAUR compared to DCF for the downlink traffic as well as the uplink traffic in the HT scenario. The right plot shows the same for the ET scenario. The results indicate good co-existence properties — in fact, the reduction in interference and contention levels in the downlink, helps the uplink to gain in throughput as well. This is a useful aspect of CENTAUR and helps improve the performance of the entire wireless environment as a whole.

3.6 CENTAUR Evaluation

We evaluated the performance of CENTAUR in detailed evaluation over two testbeds emulating the WLAN topologies of $W1$ and $W2$. We have compared the performance of CENTAUR to basic DCF as well as DCF with RTS/CTS. While DCF with RTS/CTS performed slightly better than DCF in HT-only scenarios, in mixed topologies (that include some non-HT/non-ET nodes) it performs worse due to increased overhead (13% and 24% througput reduction for UDP and TCP traffic respectively). All overheads of CENTAUR, e.g., micro-probing [23] are included in our experiments. All results reported are an average of 10 runs, where each run lasted 3 minutes.

Topologies

In all our experiments we emulate the structure of in-building WLANs by placing one testbed AP node near each production AP in the environment. We first present a comprehensive set of results for a *representative mixed* scenario that randomly distributes client nodes into offices with no particular bias. The topology has 7 APs and 12 clients with a mix of hidden (7%), exposed (16%), non-HT/non-ET (44%), and non-interfered scenarios (23%). All experiments are conducted in the 802.11a band to avoid interference with the existing infrastructure WLAN. Although the conflict graph for the same topology might change for different frequency bands, it will not affect $CENTAUR$.

Traffic and metrics

We used different types of traffic for various experiments, traversing both directions of the AP-client links. We have experimented with various PHY rates for 802.11 schemes, including the popular auto-rate fallback (ARF) mechanism that dynamically adapts the data rate. Our performance gains are persistent across all scenarios. In order to better interpret our results, most of the data presented in this section illustrate the performance for a PHY rate of 6 Mbps. Results on multiple fixed PHY rates, as well as ARF, are presented at the end of this Section.

Controlled traffic: We used UDP, TCP, as well as VoIP-like traffic (small payloads and frequencies drawn from VoIP traces). The relative volume of uplink and downlink traffic is varied across experiments. We report results on the UDP and TCP throughputs, path delays, and VoIP Mean Opinion Scores (MOS) calculated using [32].

Playback of real wireless traces: From the public SIGCOMM 2004 conference traces [139], we extract the HTTP traffic and partition it into sessions. Each session consists of a set of timestamped operations starting with a connect, followed by a series of sends and receives (transactions), and finally a close. These sessions are replayed on our testbed, by clients, emulating the mechanism described in [52]. Timing gaps between transactions are preserved. We evaluate the delays in completing each of these transactions under different schemes.

3.6.1 Performance under controlled workloads (representative topology)

We start by examining the throughput, delay, and performance of VoIP-like traffic in our representative scenario. The results are shown in Figure 3.11.

UDP throughput

Figure 3.11 (top) shows the UDP throughput of different schemes when the downlink traffic load is upto 6 Mbps per client and the uplink load is upto 1.2 Mbps per client (20% of downlink). CENTAUR with 2 ms epochs provides significant throughput gains for all underperforming links in DCF (especially links 1 and 5) by almost 5×. The aggregate throughput increases from 17.9 Mbps to 18.6 Mbps. However, CENTAUR with 10 ms epochs can take advantage of some exposed terminals and increase their throughput even further (e.g., link 8) by 1.8×. On the whole, CENTAUR with 10 ms epochs improves aggregate throughput across all links 46% over DCF.

TCP throughput

TCP traffic is bi-directional in design due to the return flow of ACKs. In this experiment we have both downlink and uplink TCP traffic with a 80:20 split as before. The overall gains are

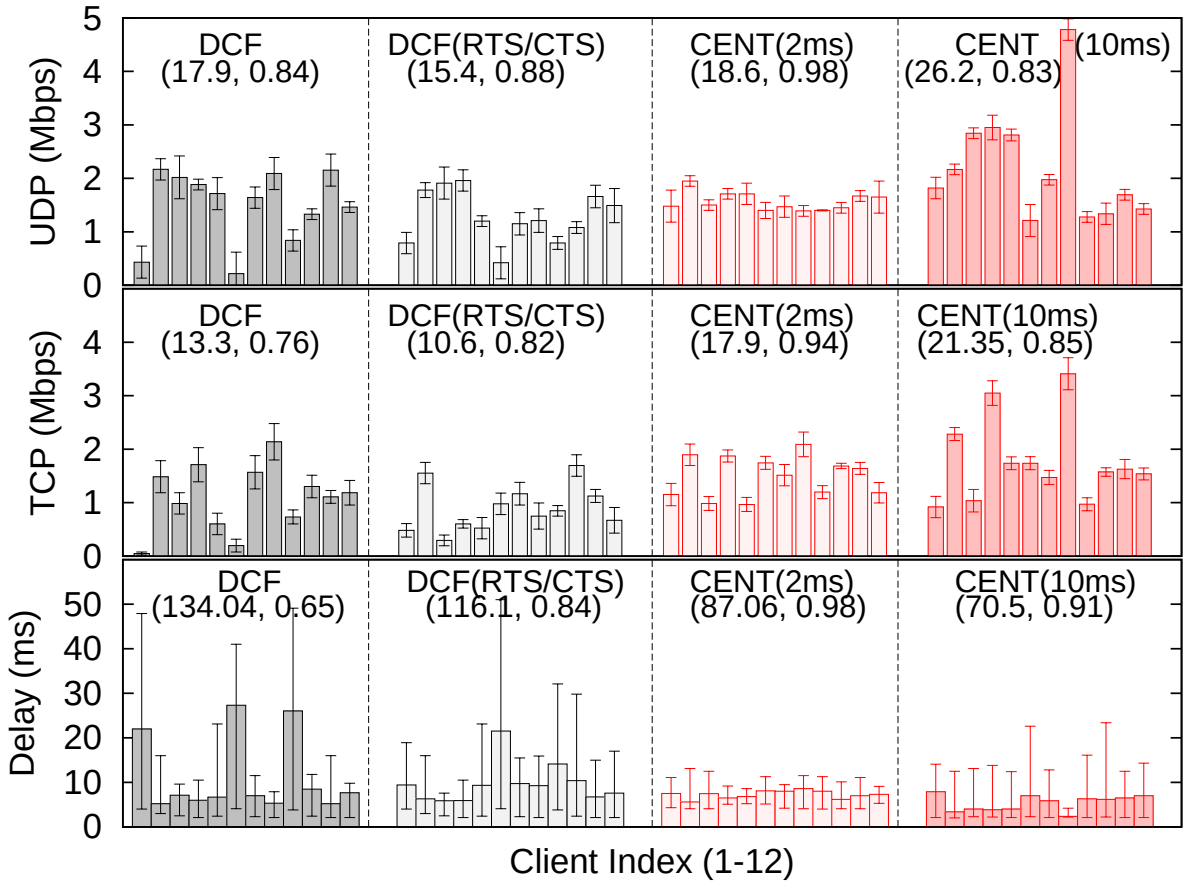


Figure 3.11: (Testbed 1) Throughput achieved under different mechanisms for a 19 node (7 AP,12 Client) topology. Plot shows the UDP throughput (top), TCP throughput (middle) and UDP delay (bottom). Experiments were run with the uplink data load being 20% of downlink load. 10th and 90th percentile values shown by error bars.

even higher than UDP. CENTAUR's ability to reduce losses and mitigate interference has an even greater impact on TCP's performance, reflected in the overall throughput gain of 61.5% over DCF.

UDP delay

We next examine the performance of UDP delay (Figure 3.11, bottom). CENTAUR with 10 ms epochs reduces the delay across all links by 47.4% when compared to DCF. The impact is particularly impressive on HT links, since their delay reduces from 49 ms in DCF to 23 ms in CENTAUR. The average delay of CENTAUR with 2 ms epoch is slightly worse than that of CENTAUR with

10 ms epochs since a 10 ms epoch is able to exploit exposed terminals efficiently. In addition, as expected, CENTAUR with 10 ms epochs leads to a higher *variability* in delay as can be observed by the 10th and the 90th percentile values also marked in the plots with error bars. We show next that this does not negatively impact delay sensitive applications.

VoIP traffic

In our VoIP-like traffic experiment, we compute the MOS values of different VoIP streams that were transmitted both in the uplink and downlink directions. Most VoIP implementations use a de-jitter buffer which limits the impact of higher latency on voice quality. However, variability in latency and packet loss are dominant contributors to VoIP MOS. The MOS value can range from 1-5, where above 4 is considered good and below 3 is considered bad. While DCF achieves a MOS of 3.35, CENTAUR with 10 ms epochs achieves a MOS of 3.75. Further, CENTAUR with 2 ms epochs, owing to its lower latency variability achieves a MOS of 4.02. We also observe that HT links get poor call quality (mean MOS was 1.83) due to increased loss rates under DCF, while the mean MOS for these links under CENTAUR was 4.05(2ms) and 3.95(10ms) respectively. Further, the impact on latency can be controlled by limiting the epoch period for scheduling. Variable epoch sizes for different class of applications, will further reduce the impact on latency. We defer such exploration of variable application specific epoch times for future work.

Impact of uplink

In order to show the impact of uplink traffic on the performance of CENTAUR, we repeat our experiments with different uplink / downlink profiles. Table 3.1 shows consistent throughput gains of CENTAUR with increase in uplink traffic volume (the values in the table are CENTAUR 10 ms epoch throughput gains normalized to DCF). We can infer that the savings in downlink hidden and exposed terminal interference result in more efficient medium utilization improving overall network performance.

Downlink load (Mbps)	Uplink load (Mbps)	Downlink Throughput 10%	50%	90%	Uplink Throughput median
6	1.2	6.78×	1.48×	1.78×	1.15×
6	2.4	3.17×	1.37×	1.75×	1.04×
6	6	2.24×	1.21×	1.53×	1.01×
2.4	1.2	1.05×	1×	1×	1×
2.4	2.4	1.32×	1.11×	1.27×	1.06×
2.4	6	1.68×	1.21×	1.49×	1.18×

Table 3.1: Normalized throughput gains of CENTAUR over DCF for different combinations of uplink/downlink UDP traffic mix. Each link is operating at 6 Mbps.

Impact of PHY rate and auto-rate fallback (ARF)

In order to understand the impact of higher rates as well as dynamic rate adaptation we repeated our experiments with different fixed rates and with ARF. We use the mechanism presented in [25] to estimate conflicts for multi rate scenarios. Note that multiple data rates can be seamlessly handled by CENTAUR through its dynamic ETT estimation, which packs a variable number of packets in an epoch depending on the data rate being used. Table 3.2 shows the mean, 10th and 90th percentile throughput gains of CENTAUR over DCF in three cases (fixed 6 Mbps, fixed 12 Mbps, and ARF). We observe that the 10th percentile of the throughput distribution is significantly improved with CENTAUR. This is because the performance gain from mitigating hidden terminals will increase if those links can transmit at higher transmission rates. With ARF, links under HT interference fall back to lower rates while CENTAUR continues to operate at a higher rate, providing persistent gains. Note that the improvement in gain slightly decreases for the 90th percentile and for higher transmission rates. This is because the use of faster transmission rates may "hide" some exposed links from each other. So, CENTAUR will have less exposed links to improve upon.

Rate	10th percentile	mean gain	90th percentile
6Mbps	6.78×	1.48×	1.78×
12Mbps	8.12×	1.54×	1.67×
Auto	7.43×	1.25×	1.32×

Table 3.2: Normalized throughput gains of CENTAUR over DCF with different PHY rates and ARF.

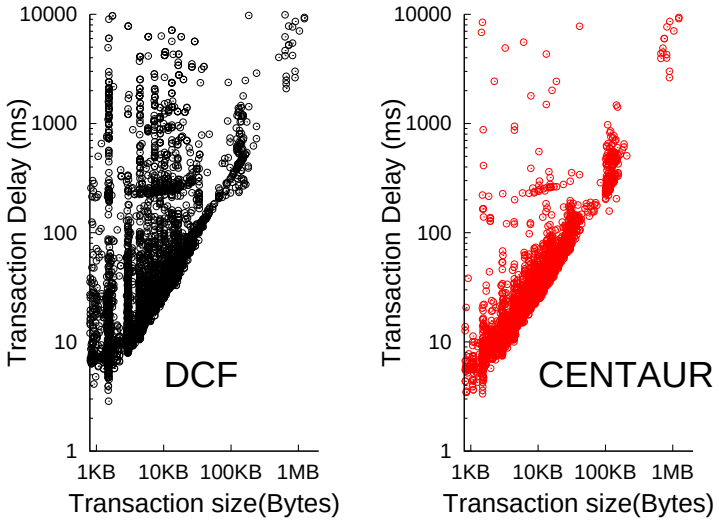


Figure 3.12: Scatter plot of delay required to complete a transaction during heavy traffic periods under DCF and CENTAUR (Testbed 1). Average transaction delay: 13.8ms (CENTAUR), 29ms (DCF).

3.6.2 Performance with real traffic traces (representative topology)

Finally, we extract the HTTP traffic out of the traffic traces captured at the SIGCOMM 2004 conference and replay it to understand how CENTAUR performs under realistic loads. We partitioned the original trace into a heavy and a light period, based on the total volume of traffic. We evaluated the performance of CENTAUR and DCF separately under the heavy and light conditions. In our experiments, each client emulated the behavior of one real client from the trace, faithfully imitating its HTTP transactions.

Table 3.3 shows the load and the corresponding reduction in transaction delay for the different HTTP transactions during the heavy and light periods used for replay. The average transaction

Name	Load(MB) (MB)	Session Count	Transaction Count	Ratio of delay (CENTAUR/DCF)		
				10%	50%	90%
Heavy	392	1655	23660	0.53	0.81	0.95
Light	68.2	744	6671	0.62	0.92	0.98

Table 3.3: Traffic periods replayed and the corresponding ratio of HTTP transaction delay (CENTAUR/DCF).

	Hidden-heavy (Testbed 2)	Exposed-heavy (Testbed 1)	Mixed (Testbed 1)
% HT	14%	0%	6.7%
% ET	0%	22%	10.2%
Overall Gains	34.7%	47.2%	44%
(HT/ET gains)	(HT: 6×)	(ET: 1.7×)	(HT: 3.2×, ET: 1.4×)

Table 3.4: Normalized throughput gains of CENTAUR over DCF for different representative topologies.

delay is reduced to 81% of its DCF counterpart during the heavy period and to 92% during the light period. Clearly, the advantages of the scheduling system are greater under higher loads. Interestingly, the 10th percentile of the delay distribution is significantly improved to 53% and 62% of its DCF value. We further examine the overall improvement in the transaction delay distribution for the heavy period in Fig. 3.12. Transaction delay is plotted against the transaction size for CENTAUR and DCF. We observe that the transaction delay with CENTAUR is close to expected, while DCF's delay can be highly variable even for smaller transaction sizes, thus revealing the effect of severe hidden terminal interference.

3.6.3 Impact of topology

Last, we examine the performance of the different schemes in three different topologies where the fraction of hidden and exposed terminals is varied. Table 3.4 lists the overall performance results obtained on three types of topologies we constructed — hidden-heavy, exposed-heavy, and mixed. The percentage of hidden and exposed terminals in these topologies are also shown in the table. All these topologies were created by changing the client positions. Uplink traffic load was 20% of the downlink load.

- *Hidden heavy topology (Testbed 2, 10 AP-client pairs):* As expected CENTAUR leads to a significant improvement in performance for all hidden terminals, improving the overall throughput by 35%. The overall fairness (computed by Jain's fairness index) improves by 89.6% as a result.
- *Exposed heavy topology (Testbed 1, 6 AP-client pairs):* In this topology, CENTAUR again outperforms DCF by 47.2% in system throughput by primarily improving the throughput of exposed terminals.
- *Mixed topology (Testbed 1, 19 nodes):* CENTAUR provides an aggregate throughput improvement of 44%. More results on this topology were presented in Section 3.6.1.

3.6.4 Summary of results

A super-set of the results presented until now is shown in Table 3.5. Our results show that (i) CENTAUR resolves HT conflicts efficiently (ii) CENTAUR when used with an epoch of 10ms also successfully exploits ET links. (ii) performance gains of CENTAUR over DCF (w/ and w/o RTS/CTS options) is higher for TCP flows over UDP flows, (iii) CENTAUR provides higher gains at increased downlink loads (iv) performance gains depend on the amount of unscheduled traffic , (v) gains of CENTAUR also depend on the fraction of HT and ET links in a topology. (vi) CENTAUR improves the overall VoIP quality, with lower epochs performing better as they introduce smaller delay.

Section	Experimental setup	Evaluation scenario	CENTAUR Gains
§ 3.2.2	2-link HT/ET/non-HT/non-ET	DET vs. DCF	HT:4×, ET:1×,non-HT/non-ET:0.82
§ 3.5	HT/ET links (Testbed 1)	DCF, DCF(w/ RTS/CTS), DET, CENTAUR	10× for HT, 1.89× for ET
§ 3.5	2-link HT/ET	CENTAUR vs. DCF with unscheduled traffic	1.4×, Uplink: up to 1.6×
§ 3.6	20-node HT-heavy (Testbed 2)	CENTAUR vs. DCF (UDP, 20% uplink)	1.34×, HT: up to 6×
§ 3.6	12-node ET-heavy (Testbed 1)	CENTAUR vs. DCF (UDP, 20% uplink)	1.47×, ET: up to 1.7×
§ 3.6.1	19-node Mixed (Testbed 1)	CENTAUR vs. DCF (UDP, variable uplink/downlink)	up to 1.48×, HT: up to 6.78×, ET: up to 1.78×
§ 3.6.1	19-node Mixed (Testbed 1)	CENTAUR vs. DCF (TCP, 20% uplink)	1.61×, HT: up to 7.4×, ET: up to 1.64×
§ 3.6.1	19-node Mixed (Testbed 1)	Impact on delay	47% (reduction in delay)
§ 3.6.1	19-node Mixed (Testbed 1)	Effect on VoIP traffic	1.4× (MOS for HT links)
§ 3.6.1	19-node Mixed (Testbed 1)	Effect of PHY rate and ARF	1.54× (12 Mbps), 1.25× (ARF)
§ 3.6.2	19-node Mixed (Testbed 1)	CENTAUR vs. DCF (Replay of real traces)	up to 0.53× (transaction delay)

Table 3.5: Summary of evaluation results. Gain is reported for throughput unless otherwise noted

3.7 Discussion and lessons learnt

Through our efforts in implementing a centralized data plane for enterprise WLANs, we learned a few valuable lessons that we summarize in this section. We also outline a few limitations of our current system.

3.7.1 Advantages of simplicity

At the beginning of this effort, we felt that centralization was an intuitive and "clever" idea, and through engineering, e.g., careful synchronization between APs, etc., we should be able to reap all benefits. However, our sustained implementation and evaluation effort proved us wrong. The biggest revelation was that DCF is a surprisingly robust protocol, which makes the gains of pure centralization small in many (non-hidden, non-exposed terminal) scenarios. While it may be possible to design better centralized scheduling algorithms that systematically outperform DCF in these scenarios, such a design is likely to be fairly complex, and may also require sophisticated hardware-level customizations. Therefore, we chose a path of simplicity, where we applied centralization to resolve hidden and exposed terminal problems. It re-iterated the belief that a simple solution is the one which can be most effective.

3.7.2 Evaluation on two wireless testbeds

To demonstrate and understand the performance of our system, we used two different large-scale testbeds, each with significantly different configurations and RF environments. We found this approach to experimentation useful, as it allowed us to eliminate testbed artifacts from inherent features of our data plane design.

3.7.3 Limitations of CENTAUR

Our current implementation of CENTAUR has two limitations. First, CENTAUR schedules packets in the order of their arrival. Better scheduling algorithms can be implemented that buffer incoming packets and schedule them in an order which maximizes the overall system throughput. While such an optimization can improve scheduling gains, it will likely increase the scheduling complexity and processing time for each packet, which may be undesirable for the scalability of the system. We defer the investigation of such tradeoffs for future work. Second, the entire CENTAUR implementation is in software. While this can facilitate immediate deployability for existing wireless networks, we were handicapped with various inaccuracies due to the fact that our off-the-shelf hardware was not designed to be scheduling friendly. We believe that better performance is achievable through customized hardware. For instance, there are potential gains in performance, if the data path between the Ethernet and the 802.11 chipsets in the AP could be made faster and jitter-free. For best results, the controller should also be equipped with multiple high speed wired interfaces, each of which control the data path of a subset of different APs in the system.

In this chapter, we try to answer the question of whether there is a useful role for a centralized data path in enterprise WLANs. We show that while centralization does not offer gains in all cases, it has a very significant role to play in mitigating downlink hidden terminals and exploiting downlink exposed terminals. Motivated by these observations, we propose CENTAUR, a hybrid architecture that centrally schedules hidden and exposed terminals, while employing DCF for uplink and legacy downlink traffic. It is based on the novel use of epoch-based scheduling, fixed backoff, packet staggering and the use of a hybrid data path. We showed that CENTAUR is able

to deliver significant performance gains for scheduled traffic, but also improves the performance of the network as a whole due to the improved utilization of the wireless medium. Importantly, CENTAUR can be implemented by any individual WLAN vendor without any changes required for clients. In the next chapter, we present the design and implementation of a practical transmit power control mechanism (Model-TPC), that is a part of the centralized control plane in enterprise WLANs.

Chapter 4

Model-TPC: Practical transmit power control

CENTAUR (Chapter 3) leverages the centralized data plane in enterprise WLAN to mitigate interference. CENTAUR modifies the data path of the wireless transmissions by adding delays to ensure that conflicting transmissions do not proceed simultaneously. It is primarily implemented by the centralized controller and does not explicitly modify the key operating parameters of the wireless APs like channel of operation and transmit power. In this chapter, we explore the possibility of utilizing the centralized controller to explicitly configure the wireless APs with suitable transmit power levels that minimize contention in the system. We present the design and implementation of Model-TPC, a practical Transmit Power Control (TPC) mechanism for enterprise WLANs, where the centralized controller collects feedback from the wireless APs to determine efficient power levels to be used for different APs in the system. We classify this mechanism as a control path mechanism as it configures the control parameter (transmit power) for the wireless APs in the system.

Power control mechanisms in wireless networks have been used to meet two different objectives — to reduce energy consumption in mobile devices, so as to conserve battery life, and to reduce interference in the shared medium, thereby allowing greater re-use and concurrency of communication. In this chapter, our focus is to study power control as applicable to the interference reduction objective. As an example, we consider the impact of power control for WLAN clients interacting with servers on the Internet. Recent theoretical work has shown that ideal medium access protocol using optimal power control can improve channel utilization by up to a factor of $\sqrt{\rho}$, where ρ is the density of nodes in the region (using fluid model approximations) [64]. Power control mechanisms [75, 168, 144] typically try to optimize the floor space acquired by wireless

transmissions by limiting the transmit power of control and data packets, thereby providing opportunity for multiple flows to coexist.

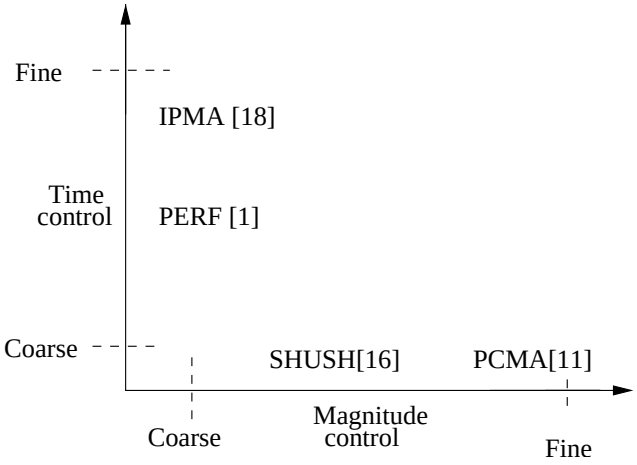


Figure 4.1: Two dimensions of transmit power control taken by prior approaches. PCMA, SHUSH rely on changing transmit power by small values (1dBm) and lie on the magnitude dimension. IPMA, Subbarao et al. rely on changing the transmit power on a per packet basis and hence lie on the time dimension

A number of research efforts have studied power control based on the theoretical abstraction of wireless signal propagation in free space and consider transmit power as a continuous variable (i.e., a fine grained parameter), that can be set per packet to yield optimal performance. Conventional power control mechanisms have exercised fine grained control in the two dimensions as shown in Figure 4.1 : 1) Time granularity at which power level is changed, 2) Magnitude granularity by which the power level is changed. We analyze both the dimensions of fine grained power control and provide guidelines for power control granularity in typical indoor environments.

Prior work [54] has pointed out that lack of vendor support for fine-grained power control mechanisms in the wireless cards inhibit deployment of these mechanisms. In this work we ask the following questions:

Is fine-grained power control really useful and would it lead to a better design of power-control algorithms? If not, what is the minimum granularity of power control that is useful in different

wireless environments, including Internet oriented wireless communication ?

We answer the first question in the negative. As we discuss in detail in this chapter, in practical indoor WLAN deployments, multipath and fading effects, coupled with various sources of interference in the unlicensed bands, imply that power control algorithms cannot derive significant benefits from very fine-grained mechanisms. We demonstrate this through detailed experimentation in different indoor wireless network environments. We estimate the distributions of Received Signal Strength Indicator (RSSI)[1] for various transmit power levels at the transmitter and show that although more power at the transmitter on average translates to more power at the receiver, there is significant overlap between the RSSI distributions for nearby power levels, making them practically indistinguishable at the receiver. This can be attributed to dominant multipath and fading effects, that lead to significant signal strength variations in indoor environments.

Our answer to the second question is that a power control algorithm can make practical use of only a *few (2-3)* discrete power levels. The exact number and choice of power levels is a characteristic of the multipath and fading of a particular wireless environment and the presence of other interfering sources.

Our observations are true for both ad-hoc networks and Internet oriented wireless communications (WLANs), and in this chapter we present our results from the latter setting. In particular, through this work we build an empirical model that allows us to characterize the specific set of power levels that is useful for a given environment and could be used to perform per packet power control.

The remainder of the chapter is organized as follows. Section 4.1 motivates the infeasibility of fine grained power control in indoor WLANs and discusses various transmit power mechanism proposed in the literature, with their respective evaluation in the context of our practical models

[1] Variations in RSSI typically correspond to variations in Signal to Noise Ratio (SNR) as shown by Reis et. al in their measurement based study of delivery and interference models for static wireless networks [53]. Moreover commodity wireless cards only report the RSSI values for each packet and hence we base our observations on the measurement for RSSI values. We further discuss this in detail in Section 4.2.

for transmit power control. In Section 4.2, we analyze the RSSI distributions under varying indoor scenarios and propose an online mechanism (Online-RSSI) to characterize the distribution in real time. We use the online mechanism to derive an empirical model for transmit power control (Model-TPC) described in Section 4.3. Section 4.4 highlights the impact of using our empirical model on end user experience through experiments on mobile enterprise clients. Finally, in Section 4.5, we discuss the key differences between cellular networks and WLANs that impact the design and functioning of transmit power control in the two settings.

4.1 Motivation : Power Control Approaches and Limitations

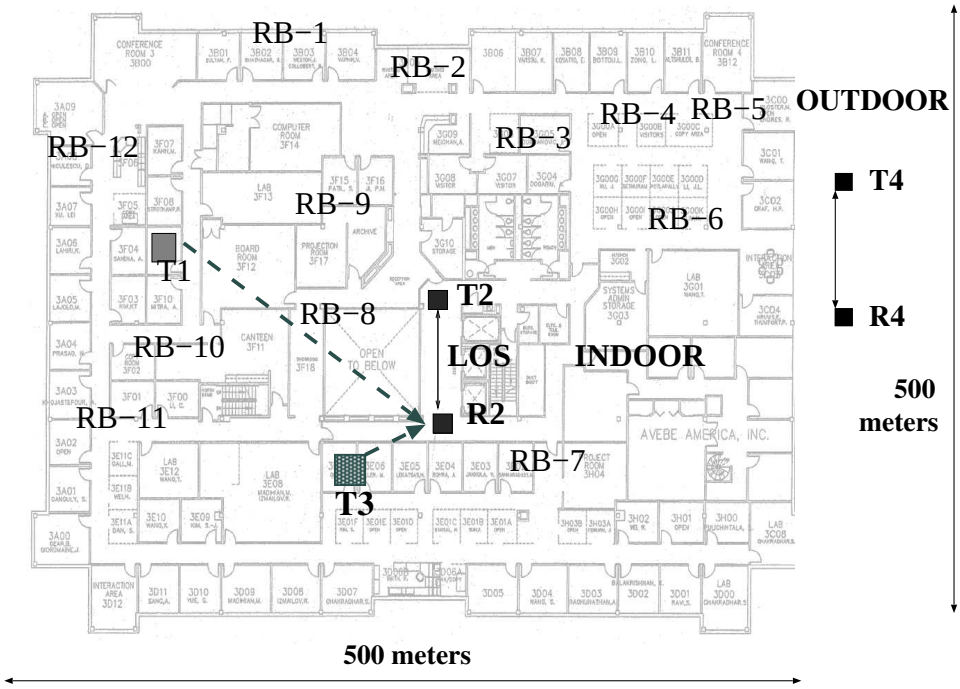


Figure 4.2: The wireless testbed, consisting of seven 802.11 a/b/g nodes (transmitters marked by T1, T2 and receivers marked by RB-1 - RB 12]). The dotted arrows indicate the transmitter-receiver pair T1-R2 and T3-R2 for our Internet oriented experiments.

Implementation of fine grained power control mechanisms has been limited by the hardware support in current 802.11 wireless cards which only have a limited number of discrete power levels.

As described in [54] , most of the wireless cards support only 4 to 5 power levels at the hardware, which is in stark contrast to the fine grained control preferred by most power control schemes like PCMA [75], SHUSH[144] and IPMA [168]. This being a limitation of current state of the art hardware, can be resolved in future wireless cards that may support fine grained power levels. However, we argue that *there are fundamental constraints to power control in indoor wireless environments, which limits the number of feasible power levels that is useful in such mechanisms.* We substantiate our claim through indoor WLAN and outdoor experiments in the following section, where we show that RSSI variations are present in both outdoor and indoor environments, but are especially dominant in indoor scenarios.

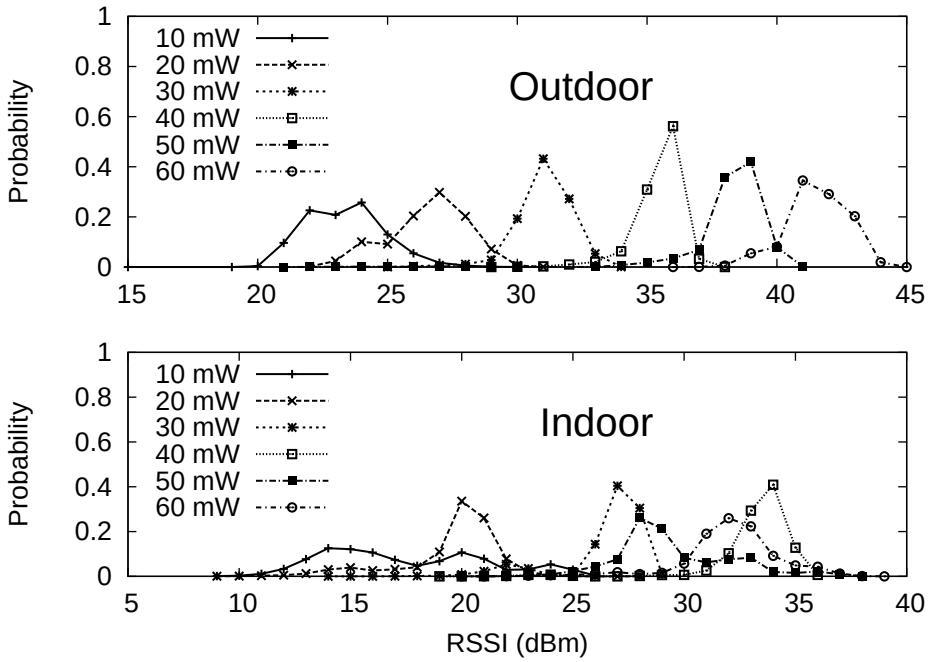


Figure 4.3: Probability Distribution of RSSI for varying power levels at the transmitter is shown in the figure. The top figure corresponds to the outdoor scenario with 6 distinguishable power levels while the bottom figure shows the effect of increased multipath and interference in the indoor WLAN scenario with the number of distinct power levels reduced from 6 to 3. Band:802.11g Data Packet Size:1Kbytes

4.1.1 Infeasibility of Fine Grained Power Control

First, we present preliminary results to illustrate the fundamental constraints of fine-grained power control.

Outdoor Scenario

This experiment consists of a set of outdoor transmitter-receiver pair (T4-R4) shown in Figure 4.2 operating using the 802.11a standard. At R4 we capture the packets transmitted by T4 for different power levels available at T4's Atheros based wireless chipset. Since low RSSI is more likely to cause a packet error, we have enabled the Madwifi driver to receive packets in error and in order to prevent the bias towards high RSSI values, we include the RSSI of erroneous packets in our calculations for RSSI distributions. Figure 4.3 shows the probability density function of RSSI distribution for various power levels at the transmitter. The power levels are increased from 10mW to 60mW (max. transmit power), in steps of 10mW. For the sake of clarity, these power levels are chosen so that there is minimal overlap between their respective RSSI distributions. For example at a power level of 60 mW, the RSSI values vary from 35dBm to 45dBm, with 40 percent of the packets being received at 41dBm. The average variation in RSSI value over all power levels is approximately 7.5 dBm.

This variation can be attributed to the multipath and fading effects, due to which the packets transmitted at the same power level, may be received with varied signal strength at the receiver. A difference of an order of wavelength in the paths taken by the wireless signals from the transmitter, can lead to the two signals being out of phase [112], resulting in variations in the signal strength at the receiver. Even though more power at the sender translates to more power at the receiver, the distributions of the received signal power overlaps significantly, thereby making them hardly distinguishable. As we show next, this effect is even more pronounced in indoor environments than in outdoor environments where there are only a few strong paths that impact the signal.

Indoor Scenario

We repeat the aforementioned experiments for an indoor transmitter-receiver pair T2-R2 as shown in Figure 4.2. The resulting distribution of RSSI values is shown in Figure 4.3. As expected the RSSI variations increase, thereby increasing the overlap between RSSI of neighboring power levels. This observation indicates that in indoor settings, the number of power levels having non-overlapping RSSI distributions is further reduced, thereby making fine-grained transmit power control much less effective. These experiments show that fine grained transmit power control mechanism are much more difficult to realize in indoor deployments.

It is evident from Figure 4.3 that in a collective fashion, the distribution of all the six power levels cover a wide range of RSSI values (20 - 45 dBm). Also note that for any single power level, its RSSI distribution overlaps significantly with that of neighboring power levels. The introduction of fine grained power levels at the hardware will imply significant overlap between the distribution of existing power levels (0,10,14,15,17,18)dBm and the new power levels. *A significant overlap between the RSSI distributions of two (successive) power levels correspondingly diminishes the practical effect of having the respective distinct power levels – they become practically indistinguishable at the receiver.* This can be considered analogous to the concept of channels in 802.11 band, where there are 11 channels available but only 3 channels are non overlapping and hence useful. Similarly, fine grained power levels cannot be distinguished easily at the receiver due to RSSI variations and hence may not be useful simultaneously.

We performed the same set of experiments at two different location, at the NEC Research Labs at Princeton and at the Computer Sciences Department at University of Wisconsin-Madison. We observed that, although the exact shape of the RSSI distribution may depend on the exact indoor environment and other interference effects, the general nature remains similar to Figure 4.3. In this work, we report our measurements from the NEC Research Labs, which we believe are representative of a typical indoor WLAN scenario.

Next we summarize prior approaches proposed in the literature that rely on fine grained power control. We show why such approaches might face difficulty in a practical implementation. We

also discuss how our proposed empirical model could act as an oracle to guide such algorithms to change transmit power effectively in practice.

4.1.2 Implications on Existing Power Control Approaches

We categorize some of the prior power control methods applicable to WLANs into two categories : 1) fine-grained in magnitude of transmit power and 2) fine-grained in magnitude of time (per-packet). Existing power control approaches can be categorized in the two aforementioned categories as shown in Figure 4.1. We explain the implications of our observations on both categories of protocols:

Magnitude Dimension of Fine Grained Power Control

Monks et al. proposed a power controlled multiple access wireless MAC protocol (PCMA [75]), within the collision avoidance framework. PCMA generalizes the transmit-or-defer "on/off" collision avoidance models to a more flexible "variable bounded power" collision suppression model. Using PCMA, the transmitter-receiver pairs can be more tightly packed into the network by adjusting the power level of the transmitter to the minimum required for a successful transmission, thereby allowing a greater number of simultaneous transmissions (spectral reuse). In order to ensure successful packet delivery, each receiver in PCMA first calculates the extra noise that it can tolerate, such that the SNR for its own packets is above the threshold for correct reception. It then advertises this noise tolerance by sending a busy tone on the auxiliary channel, and all the transmitters in the vicinity measure the received signal strength of the tone to determine the maximum power with which they can initiate their own transmissions. This mechanism requires exact calculations of received power, which may not be predictable under multipath and fading effects. Moreover, the authors treat transmit power as a continuous parameter, which may not be feasible in indoor environments due to significant RSSI variations.

Seth et al. propose a reactive transmit power control mechanism, called SHUSH [144], where nodes operate on the optimum(minimum) power required for communication. On detecting interference, SHUSH calculates the exact power required to send a RTS to the interferer and hence

optimizes the "floor space" acquired by any flow. Unlike PCMA, however SHUSH transmits at a higher power only when a flow is interrupted by external interference. Again SHUSH assumes fine grained control on power levels and ignores RSSI variations which can make it difficult to infer the exact interference at the receiver, thereby complicating the calculation of target transmit power required to SHUSH the interferer. Our experimental observations suggest that such observations are too deviant from realistic scenarios. Using our empirically derived power control model (Section 4.3), the above mechanisms could dynamically determine an exact set of feasible power values to be used in an environment.

Time Dimension of Fine Grained Power Control

Many researchers in the past have proposed schemes which require change in the power level on a per packet basis.

Akella et al. [26] discuss some power control mechanisms in their work on wireless hotspots. They propose that APs should use the minimum transmit power required to support the highest transmission rate. In their scheme, the receiver sends the value of observed RSSI, averaged over some small number of packets, as a feedback to the transmitter. The transmitter on receiving the average RSSI value on the receiver side, decides the optimal power level suitable for use in the current channel conditions. However they do not provide exact values for power level granularity that should be used. As discussed earlier, a simple average of RSSI values at the receiver may not give a correct estimate of the actual SNR.

Subbarao [155] has proposed a dynamic power-conscious routing mechanism that incorporates link layer and physical layer properties in routing metrics. It routes the packet on a path that requires the least amount of total power expended and each node transmits with the optimum (minimum) power to ensure reliable communication. This scheme requires per packet power control and also needs feedback from the destination regarding RSSI on a per packet basis.

Similar to PCMA approach, Yeh et al. [168] proposed an interference/power aware access control. They augment the normal RTS/CTS mechanism of IEEE 802.11 with provision for multi level RTS, where the transmit power of the RTS mechanism is set on the basis of the intended

receiver. Such a dynamic per packet approach becomes difficult in the face of significant RSSI variations, making it unattractive for practical deployments.

We analyze the stationarity (coherence time) of signal strength for various scenarios and propose a simple algorithm Online-RSSI, that can be used to determine the distribution of signal strength for a given transmit power level in any scenario. Once the set of feasible power levels (having non overlapping signal strength distribution) is derived, the receiver can use this model to determine the transmit power of the transmitter for a packet received at any given signal strength and hence provide correct feedback to the transmitter on a per packet basis (or similar time scales).

4.2 Characterizing Signal Strength Distribution

Our experiments serve three main purposes: (i) to gain an understanding of the characteristics of RSSI variations under varying practical scenarios (in terms of user movements, shadowing, multipath and external interference) (ii) as a learning data-set to build our empirical model for identifying the set of feasible power levels (iii) as an input to validate this model.

In this section, we characterize the distribution of RSSI under varying magnitudes of multipath, shadowing and other 802.11 and non 802.11 interference for a real WLAN deployment shown in Figure 4.2. By studying the RSSI distribution across different power levels and different channel conditions, we formulate mechanisms to dynamically predict and construct such distributions in real-time. Such mechanisms shall be used in the next section where we build a model to predict the useful power-levels in a given environment. We briefly describe various components of our experimental setup.

4.2.1 RSSI measurements

The performance of most wireless applications depends on the packet delivery probability. The SNR is widely used in the literature to model packet delivery probabilities: packets are successfully received if S/(I+N) is above a certain threshold, and otherwise are not. Commodity wireless cards do not report the information required to compute SNR. For instance, our cards report only their version of RSSI, the minimum feedback allowed by the 802.11 standard. Some other cards

also report an estimate of I by measuring energy in the air when no packets are being sent, but this estimate may be inaccurate during packet delivery. It has been shown in a prior measurement based study by Reis et. al [53] that RSSI is generally predictive of delivery probability in static wireless networks and while wireless networks exhibit substantial variability, measurements of average behavior over even relatively short time periods tend to be stable. This phenomenon was also observed in our joint power and data rate adaptation experiments (described as an application of our model in Section 4.4), where the power levels with significant overlap in their corresponding RSSI distribution, perform similarly in terms of rate adaptation. Since rate adaptation again depends on packet delivery rate, we can infer that RSSI is a reasonable estimate for SNR and two power levels with significant RSSI overlap at the receiver will perform similarly for packet delivery probabilities. Hence we base our measurements and models on RSSI values that are readily available from the commodity wireless cards.

RSSI estimates signal energy at the receiver during packet reception, measured during PLCP headers of arriving packets and reported on proprietary (and different) scales. Atheros cards, for example report RSSI as $10log_{10}(\frac{S+I}{n})$, where S is the signal strength of the incoming signal, I is the interfering energy in the same band, and n is a constant $(-95dBm)$ that represents the "noise floor" inside the radio. Atheros RSSI is thus dB relative to the noise floor. To give results that are independent of card vendors, we transform RSSI values to *received signal strength* (RSS) values, that give absolute energy levels. That is, RSSI is defined to be S+I. Note that these RSSI measurements are performed at the receiver and then provided as a feedback to the transmitter for constructing the empirical model for feasible power levels.

4.2.2 Validating Available Hardware Power Levels

To ascertain the available power levels in 802.11 WLAN cards, we measure the voltage across the wireless card of the transmitter by the setup shown in figure 4.4. The setup constitutes of a 0.1 ohm sense resistor, R, connected in series to the circuit of the wireless device (pcmcia card) that exposes the voltage supplied to the device. For the pcmcia based 802.11 card, we used the Sycard 140A cardbus adapter, to expose the voltage supplied to the card. A Data Acquisition Card (DAQ),

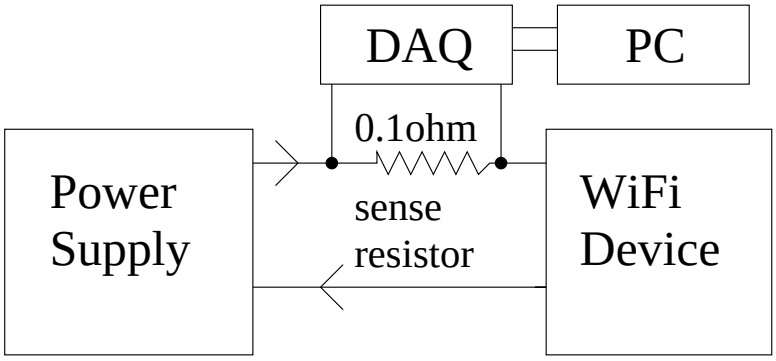


Figure 4.4: Figure shows the setup used to determine power drawn by wireless cards. The DAQ samples voltage across the WiFi device and sends it to a PC via USB. Performed at the transmitter to validate the power levels available at the hardware.

DS1M12 Stingray Oscilloscope, samples the voltage through R at a rate of 1 million samples per second, thereby giving us voltage measurements on a per packet basis. The instantaneous power consumption, P_i can therefore be written as $P_i = V_d \times V_R/R$ where V_d is the voltage provided to the WiFi device and V_R is the voltage drop across R at a given moment. These measurements are performed at the transmitter and show that indeed the right power levels are implemented at the hardware circuitry of the transmitter's wireless interface. On the basis of power consumed by the wireless interface, we validated that Cisco Aironet cards provide 6 different power levels for 802.11g and 5 different power levels for 802.11a respectively.

4.2.3 WLAN Trace Collection

In order to understand the behavior of RSSI under varying interference and multipath effects, we conduct detailed experiments to collect RSSI traces in an office building under varied indoor settings. In all our experiments, we use a fixed data rate of 1Mbps and fixed packet size of 1KB, so that the time intervals are directly translated into number of packets (modulo 802.11 DCF), which is the X axis for most of our plots. This facilitates easier packet based analysis of RSSI traces and their implications to power control mechanisms, which generally base their decisions on a per packet basis. For our experiments, 1 sec of receiver time window ≈ 1000 packets of 1KB

each (unless otherwise specified). *We repeated the same experiments with other wireless cards and found the results were consistent with the ones reported here.* We discuss the exact set up for each of these scenarios.

Line of Sight - light interference(LOS-light)

These experiments represent a scenario where the transmitter-receiver pair are in direct line-of-sight and were performed at night to minimize external interference. Figure 4.2 shows the placement of transmitter-receiver pair T2 and R2 respectively for LOS-light experiment. The experiment used 2 IBM Thinkpad laptops running Linux kernel 2.6. Each of the laptops housed an Atheros chipset based 802.11a/g Linksys wireless card and used Madwifi drivers. We used *Netperf 2.2* to generate UDP flows between the two laptops and collected MAC-level traces for the packets received at the receiver using the *pcap* standard library. We vary the power of the transmitter to understand their corresponding effects on RSSI.

Non Line of Sight - light interference(NLOS-light)

The experiment comprises of a single transmitter T1 and 5 receivers (RB-1, RB-8, RB-10, RB-11 and RB-12) as shown in Figure 4.2 placed at various locations in the building and used *netperf* and *pcap* library to generate flows and collect traces respectively. None of the receivers were in direct line-of-sight of T1 and these experiments were performed at night to minimize external interference.

Line of Sight - heavy interference(LOS-heavy)

We investigate the effect of controlled interference on RSSI. We use our experimental testbed shown in figure 4.2 for line of sight experiments to evaluate the effect of heavy interference (like bulk data transfers) on RSSI variations. Nodes RB-12, RB-11 and RB-2 act as separate APs and perform bulk data transfers with their respective clients (3 IBM laptops). Nodes T2 and R2 form a transmitter-receiver pair.

Non Line of Sight - heavy interference(NLOS-heavy)

We use our experimental testbed shown in figure 4.2 for non-line of sight experiments to evaluate the effect of heavy interference (like bulk data transfers) on RSSI variations. Nodes RB-12, RB-11 and RB-2 act as separate APs and perform bulk data transfers with their respective clients (3 IBM laptops). Nodes T1 and RB-8 form a transmitter-receiver pair.

4.2.4 Analyzing WLAN Traces

Figure 4.5 shows the smoothed moving average of RSSI per packet for the four categories of traces described in the previous section. Although we collect many traces from each category (namely LOS-light, NLOS-light,LOS-Heavy and NLOS-heavy), we present only one representative trace from each category. The representative trace is chosen such that it manifests the basic characteristic of traces from that particular category. All these traces are collected at 1Mbps of data rate with packet size of 1000 bytes.

As clear from Figure 4.5, the variations in RSSI are minimum for LOS-light trace and is maximum for the NLOS-heavy trace. This behavior is expected because the factors contributing to RSSI variations increase in both number and magnitude from the topmost plot to the bottom. Figure 4.6 shows the probability distribution of RSSI values at the receiver for the four scenarios. Clearly, the distribution of RSSI becomes flatter (larger variation) with the increase in interference and multipath effects, with the distribution of LOS-light and NLOS-light resembling a Gaussian distribution. Next we analyze these traces in detail to understand temporal variations in RSSI and propose an algorithm to dynamically characterize the distribution of RSSI in any environment.

Stationarity

Figure 4.5 shows the variation of RSSI on a per packet basis, but it would also be useful to observe the amount of fluctuation over a set of packets (or a burst). Such an analysis would reveal any characteristic burst intervals where RSSI values vary largely over different bursts but deviate minimally within a burst. Also note that since our experiments are conducted with the traffic sent at uniform rates packet intervals directly correspond to time intervals (modulo 802.11 DCF

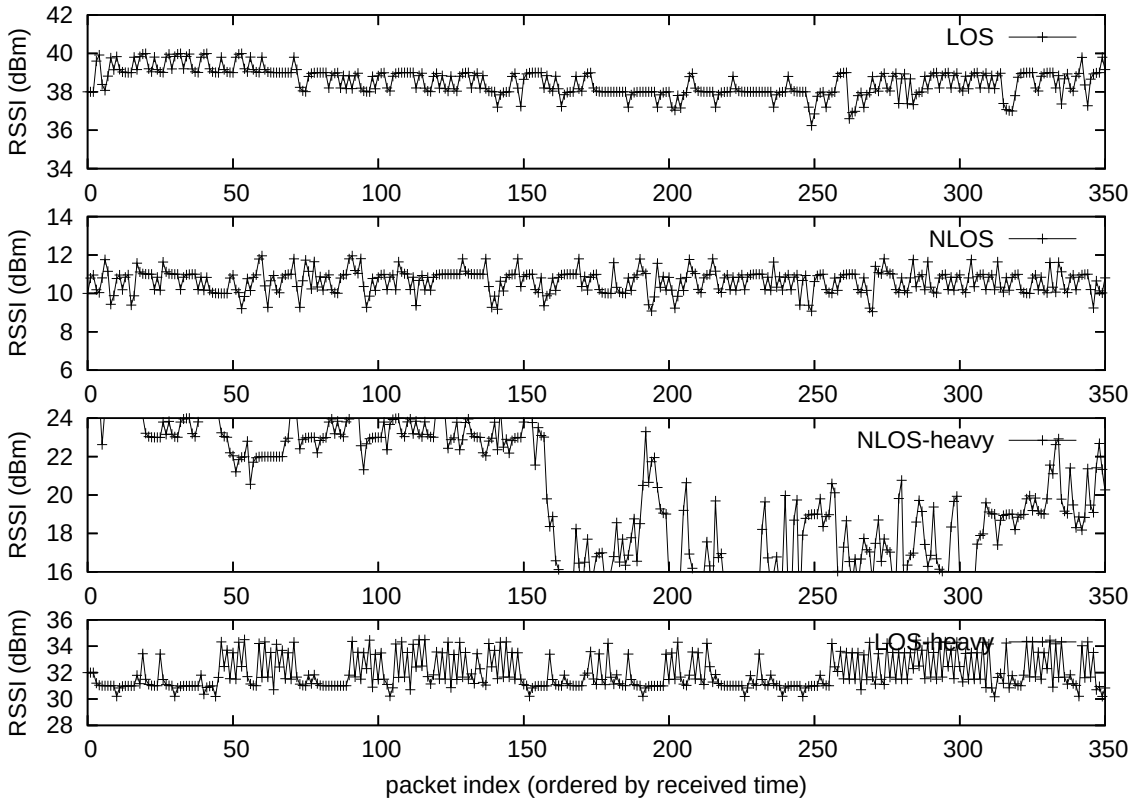


Figure 4.5: Exponentially weighted moving average of RSSI over time for four traces collected under various practical scenarios, with varying degree of external interference, multipath, shadowing and fading effects. The packets are sorted in order of received time. The traces from topmost plot to the bottom belong to LOS-light, NLOS-light, NLOS-heavy and LOS-heavy. Note that the scale of Y axis is adjusted for each trace for clarity. The high variation of RSSI for NLOS-heavy can be observed in the figure.

effects). One way to summarize changes at different time scale is to plot the Allan deviation [27] at each packet interval. Allan deviation is the square root of the two sample variance formed by the average of the squared differences between successive values of a regularly measured quantity taken from sampling periods of the measurement interval. Allan deviation differs from standard deviation in that it uses differences between successive samples, rather than the difference between each sample and long term mean. In this case, the samples are the fraction of packets delivered in successive intervals of a particular length. The Allan deviation is appropriate for data sets where

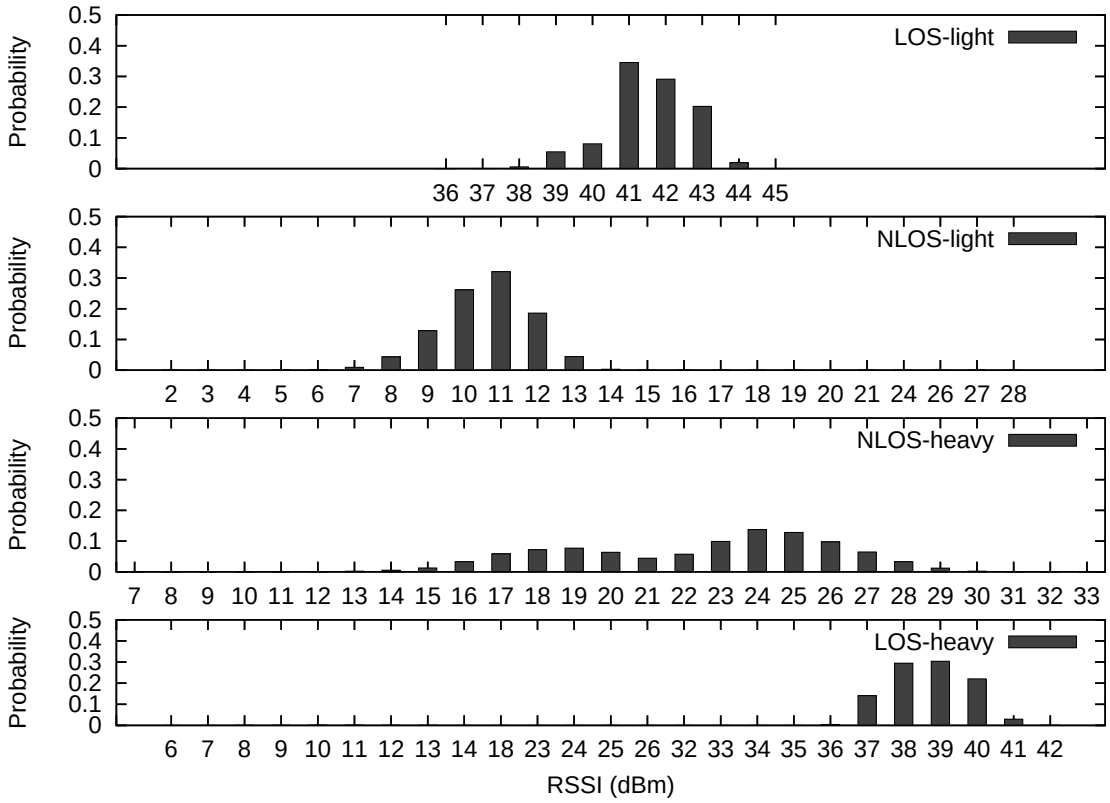


Figure 4.6: Probability distribution of RSSI for the four traces shown in Figure 4.5. The spread in RSSI distribution is noticeable in all the traces, with the NLOS-heavy trace having the maximum deviation. In the NLOS-heavy scenario, the RSSI values show persistent fluctuations about two different RSSI values (bimodal distribution).

data has persistent fluctuations away from the mean. The formula for the Allan deviation for N measurements of T_i and sampling period τ_0 is:

$$\sigma_y(\tau_0) = \sqrt{\frac{\sum_{i=1}^{N-1}(T_{i+1} - T_i)^2}{2(N-1)}} \tag{4.1}$$

The sampling period is varied by averaging n adjacent values of T_i so that $\tau = n\tau_0$. For simplicity of expression, we define: $X_i = \sum_{k=1}^{i} T_k$. The Allan deviation for different values of n can be given by:

$$\sigma_y(\tau) = \sqrt{\frac{\sum_{i=1}^{N-2n+1}(X_{i+2n} - 2X_{i+n} + X_i)^2}{2(N-2n+1)}} \tag{4.2}$$

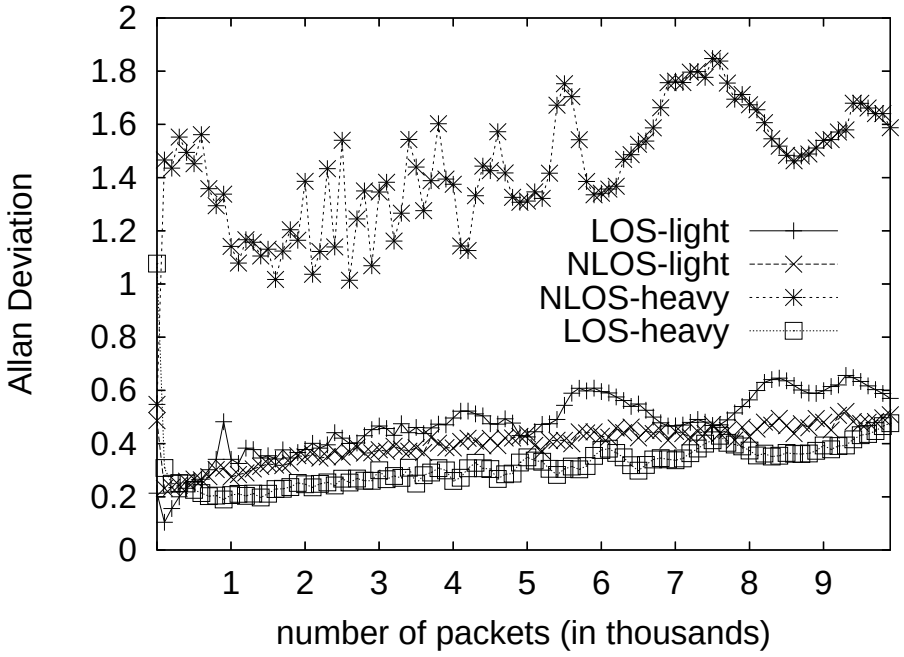


Figure 4.7: Allan deviation for the four representative traces shown in figure 4.5. The y axis shows the Allan deviation ($\sigma(\tau)$), while the value of n (sampling period in Equation 4.2) is varied on the x axis. It shows that there are no clear peaks for the RSSI bursts for any scenario, however it is clear that Allan Deviation becomes quite stable (between 0.2 and 0.5) for LOS-light, NLOS-light and LOS-heavy scenarios. The NLOS-heavy has relatively higher deviation and shows significant fluctuations but remains in the range of (1.6-1.8).

The Allan deviation inherently provides a measure of the behavior of the variability of a quantity as it is averaged over different measurement time periods, which allows it to directly quantify and distinguish between different types of RSSI variations. The Allan deviation will be high for interval lengths near the characteristic burst length. At smaller intervals, adjacent recent samples will change slowly, and the Allan deviation will be low. At longer intervals, each sample will tend towards the long term average, and the Allan deviation will again be small.

Figure 4.7 shows the Allan deviation of RSSI over large scale packet intervals (thousands of packets). We can observe that although there are no prominent peaks for the RSSI bursts for any scenario, Allan Deviation becomes quite stable (between 0.2 and 0.5) for LOS-light, NLOS-light

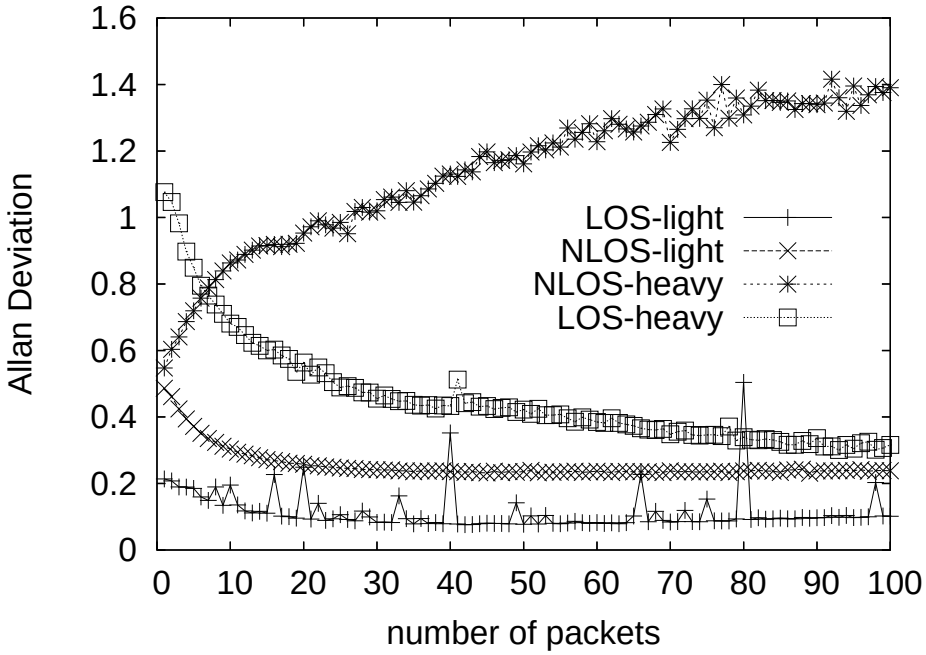


Figure 4.8: Zoomed version of Allan deviation for short interval of time ($\approx$ 100 packets). Allan deviation decreases sharply for LOS-light, NLOS-light and LOS-heavy traces, indicating independent packet losses. But Allan deviation for NLOS-heavy increases, indicating very small bursts and highly variable wireless channel. This is a strong indication that fine grained power control becomes even more difficult when multipath effects are coupled with 802.11 interference.

and LOS-heavy scenarios. The NLOS-heavy has relatively higher deviation and shows significant fluctuations in the range of (1.6-1.8). In Figure 4.8, we show the zoomed version for Allan deviation for intervals less than 100 packets. This figure shows the short term characteristic of RSSI variations. As clear from the figure, Allan deviation for LOS-light, NLOS-light and LOS-heavy is maximum at 1 packet, then decreases sharply because averaging over longer intervals rapidly smoothes out fluctuations. This means that the RSSI variations for the aforementioned three categories are independent for intervals less than 100 packets. On the other hand, NLOS-heavy shows sharp increase in Allan Deviation from 0.6 to 1.4. This indicates that in NLOS-heavy trace, the RSSI averaged over small sample sizes (τ in Equation 4.2), changes quickly leading to a sharp increase in Allan Deviation at such small scales. On further analysis, we found that deviation for

NLOS-heavy reaches 1.7 for about 400-500 packets and as shown in Figure 4.7, fluctuates around that value for larger packet intervals as well. We agree that there is no clear decrease in the Allan deviation for any scenario, so we approximate the value of burst size at the point when the deviation becomes quite stable (or the rate of increase in deviation becomes very low). Hence we choose ≈ 400 packets for NLOS-heavy and on the order of thousand packets for LOS-heavy, LOS-light and NLOS-light.

We report these burst size for various LOS and NLOS scenarios in Table 4.1. The burst size information is used by our algorithm Online-RSSI (explained in Section 4.2.5), that samples the packets in multiples of these burst sizes for determining the signal strength distribution for a given transmit power level. As RSSI varies significantly across bursts, the online mechanism needs to consider at least an increment of burst size in its sampling process to determine if the online distribution being computed has stabilized. This allows us to quickly converge to an accurate RSSI distribution as explained in Section 4.2.5.

Summary: RSSI variations are bursty for intervals on the order of ≈ 1000 packets for LOS-light, NLOS-light and LOS-heavy scenarios. But for NLOS-heavy traces, the Allan deviation increases even in the small interval of 100 packets, depicting bursts even in short packet intervals. This can be explained because the interference coupled with multipath effects make the wireless channel highly variable and leads to bursts even in very short time intervals. This behavior was observed in all our NLOS-heavy traces (for various receivers) and indicates high variability in the wireless environment. Allan deviation provides an estimate of burst length of a trace and could be interpreted as an effect of temporal variations in wireless channel. So if Allan deviation shows that a trace has very small burst periods (as in the case of NLOS-heavy), it can be used as an indication that per-packet power control will be highly unpredictable. Finally we observe that all the scenarios show substantial non-stationarity in RSSI variations, which will further impede fine grained mechanisms for power control.

Entropy

Through the empirical analysis presented in Section 4.1.1, we observed that due to multipath, fading and other propagation effects, the RSSI values at the receiver show significant variation (also corroborated by Figure 4.6). Depending on the exact environment, RSSI distributions for close transmit power levels can have substantial overlap, making them practically indistinguishable at the receiver. For a power control scheme to be effective, it needs to determine the set of useful power levels i.e. power levels with minimum overlap. In order to estimate the number of power levels in any setting, we need to estimate the corresponding RSSI distribution for various power levels. Ideally, we can sample the RSSI values for a very long period of time ($\approx 10mins$) to obtain the true behavior of the RSSI distribution. But, as we show next, sampling a very large number of packets may not be necessary (or practical, due to computation and storage limitation on the clients) in most settings. This observation leads us to the following question: *What is the minimum number of packets we should sample to get a "good" approximation of the RSSI distribution ?*

We first describe an offline mechanism to determine the number of samples that are required to generate a distribution close to the one computed over a large number of packets, as shown in Figure 4.6. On the basis of insights obtained from the offline analysis, we then present a simple online mechanism to dynamically determine the number of packets sufficient to characterize the RSSI distribution in any environment.

Let us define the actual probability distribution function for RSSI (over a large number of packets $\approx 100,000$) as $p(x)$. The approximate distribution obtained by our mechanism is denoted by $q(x)$. We now describe the statistical measure that we use to quantify the performance of the model.

Let $p(x)$ and $q(x)$ be two probability distribution functions defined over a common set χ. We describe a commonly used statistical measure *Kullback-Leibler Divergence* (KLD) that quantifies the 'distance' or the relative entropy between two probability distributions. This comprises a general measure and allows us to compare the statistics of all the orders for the two distributions. The *Kullback-Leibler Divergence* (KLD) [45] is defined as

$$D(p(x)||q(x)) = \sum_{x \in \chi} p(x) \left| \log \frac{p(x)}{q(x)} \right| \tag{4.3}$$

The KLD is zero when the two distributions are identical and increases as the distance between the distributions increase. The KLD is a measure used in information theory to calculate the 'distance' between two distributions $p(x)$ and $q(x)$. The definition of the KLD carries a bias for random variables with higher entropy. Hence to evaluate the relative distance accurately for our purposes, it is important to weigh in the entropy of the original distribution which can be large. The entropy $H(p(x))$ of the random variable x with distribution $p(x)$ is the average length of the shortest description of the random variable given by:

$$H(p(x)) = \sum_{x \in \chi} p(x) \log \frac{1}{p(x)} \tag{4.4}$$

Hence we use the normalized Kullback-Leibler divergence NKLD [98] defined below as a measure of distance between two distributions

$$\text{NKLD}(p(x)||q(x)) = \frac{D(p(x)||q(x))}{H(p(x))} \tag{4.5}$$

However the above metric is asymmetric and we make it symmetric by taking an average of NKLD(p(x)——q(x)) and NKLD(q(x)——p(x)). The symmetric average distance between two distributions is given by

$$\text{NKLD}(p(x), q(x)) = \frac{1}{2}\Big(\frac{D(p(x)||q(x))}{H(p(x))} + \frac{D(q(x)||p(x))}{H(q(x))}\Big) \tag{4.6}$$

Like KLD, NKLD is also zero for identical distributions and increase with the increase in distance between the two distributions, p(x) and q(x). Ideally we could have characterized the distance between two probability distributions by calculating the area of their intersection on some data set X. However this will require calculating their points of intersections and some numerical integration techniques, which may be cumbersome depending on the exact shape of the distribution. Hence we use NKLD as it compares the statistics of all orders for two distributions and is very simple to compute in real time. Further NKLD works efficiently for our experimental scenarios.

We consider the long term probability distribution as $p(x)$ and those derived from our offline mechanism as $q(x)$. Let n be the length of the packet sequence that is used for computing the distribution $q(x)$. The value of n is varied and we measure the corresponding NKLD for each $q(x)$ (with $p(x)$ as the reference long term distribution).

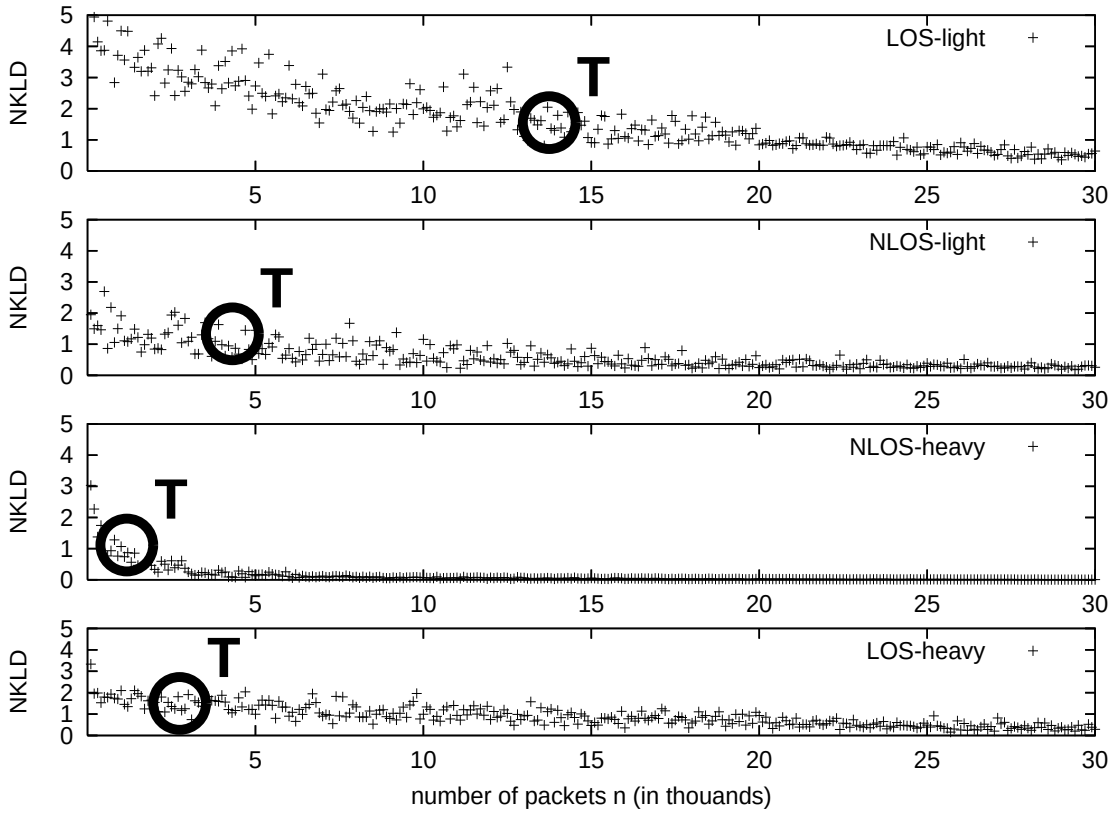


Figure 4.9: Normalized Kullback-Leibler Divergence (NKLD) for the four representative traces. Clearly for NLOS-heavy trace, NKLD decreases sharply with the increase in number of packets, reaching a value of 1 for a sample size on the order of 5000 packets. For LOS-light however, this value is around 30,000 packets.

Figure 4.9 shows the NKLD curve obtained for the representative traces from the four categories discussed before. NKLD is a decreasing function of n, although the exact shape of the curve varies as per the environment. We assume without the loss of generality, the tolerable error or relative distance between actual distribution and distribution obtained by sampling n packets

be 10%. Figure 4.9 can be used to calculate the length of packet sequence required to achieve the error bound under varying scenarios. While LOS-light and NLOS-light require about 20,000 packets each, LOS-heavy and NLOS-heavy scenarios require less than 10,000 packets as shown in Table 4.1.

Summary: The number of packets required to determine a close approximation for RSSI distribution is especially high for the LOS-light scenario while for a NLOS-heavy scenario the number is relatively lower. The accuracy of an RSSI distribution varies directly with the number of bursts captured. Since, the NLOS-trace has short burst sizes we can obtain large number of bursts using a smaller trace to accurately model the RSSI distribution while the trace required for LOS-light scenario is larger owing to longer burst sizes. This analysis shows that sampling very large number of packets (≈ 100000) to obtain the RSSI distribution is not required in the majority of traces, with the notable exception of LOS-light scenario.

4.2.5 Algorithm Online-RSSI

Based on the above analysis, we describe an online algorithm to compute the RSSI distribution in an online fashion by predicting the number of packets needed in order to accurately characterize the distribution in any environment. As shown in Figure 4.9, initially NKLD (or error) decreases rapidly with the increase in n, but stabilizes after a threshold T, slowly tending to zero. It implies, that beyond a certain length of packet sequence, the decrease in NKLD(or error) is minimal and hence there is not much gain in sampling longer packet sequences. The online algorithm is shown in Figure 4.10. The enabling observation for the above algorithm is that after the NKLD curve stabilizes, increasing the length of packet sequence does not change the distribution substantially. So we compute the RSSI distribution for n and $n + burst\_size$ for varying values of n and return the value for which both the distributions have relative distance less than the tolerance level. We use burst_size as an increment, as RSSI varies significantly across bursts and we need to consider at least a gap of more than burst_size to conclude that the RSSI distribution has stabilized. For our experiments we find that typically an increment of one burst_size is sufficient to yield correct results using the online mechanism.

Online-RSSI(burst_size,tolerance)
initialize n to 1
sample(n) = Sample_Random_Sequence(n)
q(x) = Compute_RSSI_Distribution(sample(n))
do
n' = n + k * burst_size
sample(n') = sample(n) + Sample_Random_Sequence(k * burst_size)
q'(x) = Compute_RSSI_Distribution(sample(n'))
if Compute_NKLD($q'(x)||q(x)$) $\leq$ tolerance
return q(x)
update n = n', q(x) = q'(x) and continue

Figure 4.10: Algorithm to find length sequence n for which the RSSI distribution stabilizes

Table 4.1 shows the values of n obtained for the four representative traces shown in figure 4.5. The value of n obtained using an online mechanism is close to the value obtained using offline analysis of the traces. In order to evaluate the efficacy of our online mechanism (to determine n) we compare the distribution obtained using a packet sequence of length n with the distribution obtained using large traces ($\approx 100,000$). Figure 4.11 shows that the distribution obtained using n as determined by the online mechanism closely approximates the true distribution for all the traces.

Validating Efficiency of Online-RSSI : We validate the efficiency of Online-RSSI by using the traces collected in our indoor WLAN deployment as described in Section 4.2.1. Using those traces, we first build an accurate estimate of the signal strength distribution for each scenario for different power levels. These distributions are computed over large traces (comprising of $\approx 100,000$ packets) and act as a baseline against which we compare the distribution generated by Online-RSSI. Figure 4.11 shows the accuracy of Online-RSSI for a given power level in different scenarios. The

Trace	Burst Size	Offline		Online	
	# of pkts	# of pkts	NKLD	# of pkts	NKLD
LOS-light	≈ 1000	30,000	0.5	22,000	0.8
NLOS-light	≈ 2500	20,000	0.5	20,000	0.8
LOS-heavy	≈ 3000	16,000	0.5	9000	0.8
NLOS-heavy	≈ 400	3000	0.5	5000	0.05

Table 4.1: Minimum packet length sequence for capturing the distribution of RSSI, as calculated by offline and online mechanisms. Corresponding NKLD distance with the long term ”true” distribution is also given. NKLD of 0.5 is chosen as the threshold for determining the packet length sequence in the offline mechanism. Burst sizes corresponding to first noticeable peak in Allan deviation is shown.

results for different power levels are similar in nature to the ones presented here. The base line distributions for different scenarios are shown in dotted lines and the real time distribution generated by Online-RSSI is shown in solid lines. As shown in the figure, Online-RSSI is able to accurately estimate signal strength distribution and the errors (NKLD distance between baseline and estimated) are found to be within **5**% for LOS-light, NLOS-light and NLOS-light, while for NLOS heavy it was found to be with **20** %. This indicates that the algorithm has reasonable accuracy in estimating the RSSI distribution in an online fashion for different scenarios.

4.3 Empirical Model for Power Control

As discussed in Section 4.1.1, RSSI values of neighboring power levels tend to overlap significantly in indoor scenarios, with some indoor settings more prone to multipath effects (like cubicles) than others (like large conference halls). Similarly the interference and other factors that determine the extent of RSSI variations will be different for different indoor environments. Hence, it is possible that some indoor environments may allow more power levels to be distinguishable (where RSSI variations are low) as compared to others (where RSSI variation is high). Based on our online mechanism to dynamically determine the number of packets required to characterize

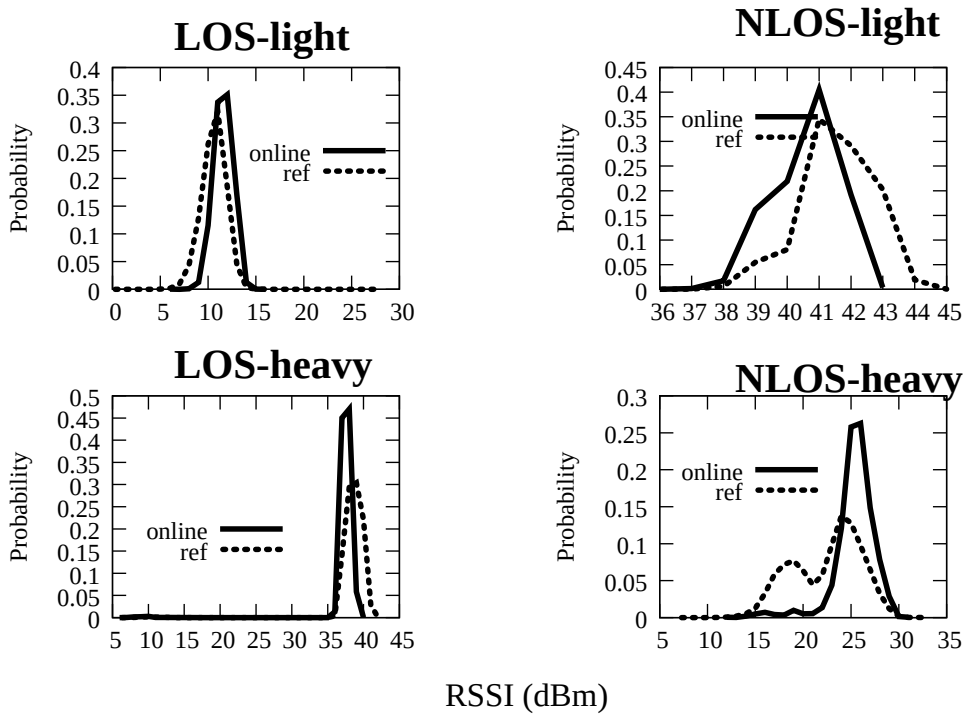


Figure 4.11: Comparison between distributions obtained from n packets (as determined by the online algorithm) and the true distributions obtained from long term traces. We use the highest power level of 60mW for this experiment. Similarity between the two distributions indicate the efficacy of our online mechanism

RSSI distribution in any environment, we present an empirical model for transmit power control, *Model-TPC*, that outputs the set of feasible (non-overlapping distribution) power levels for a given indoor setting.

4.3.1 Model-TPC

Construction of our model proceeds through the following important steps, also shown in Figure 4.13. Assume we are operating in the context of a wireless node X.

1. **Estimating RSSI distribution:** The RSSI distribution for any given power level is estimated using the Online-RSSI algorithm described in Section 4.10. Note that the RSSI distribution is captured at the receiver and communicated back to the sender as a feedback, as shown

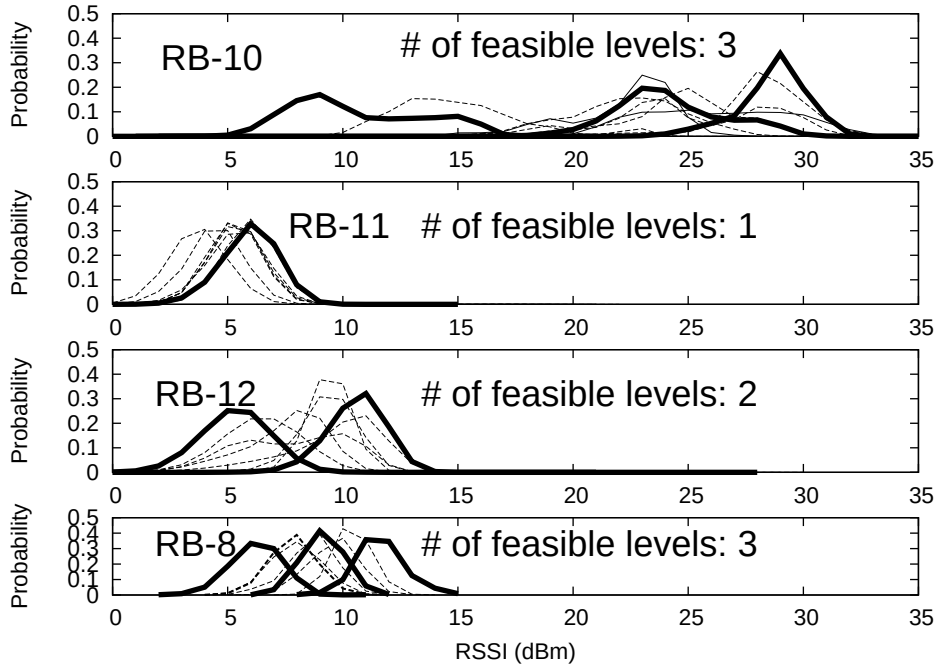


Figure 4.12: Probability distribution function for RSSI values received at varying power levels at the transmitter. The plots represent the distributions at receiver RB-10, RB-11, RB-12 and RB-8, in order from top to bottom. The exact positions of these receivers with respect to the transmitter can be seen in figure 4.2. The amount of overlap varies with the location and only 2-3 power levels are distinguishable at most of the receivers.

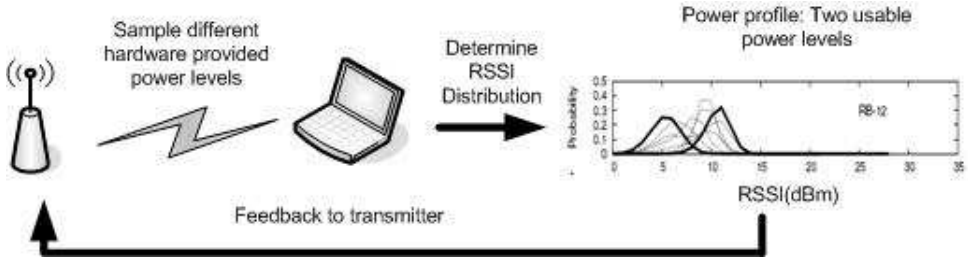


Figure 4.13: Steps involved in construction of Model-TPC. The receiver estimates the RSSI distribution using our Online-RSSI and computes the set of feasible power levels as applicable to itself. This information is then sent to the transmitter to be used in power control

in Figure 4.13. Many proposed approaches (such as [26]) already incorporate protocol-level constructs to implement such functionality. Ongoing data communication between the participating nodes can be leveraged to gather this information. This process is repeated for different power levels available in the hardware. Note that for our experiments, this procedure is repeated for different hardware available power levels (6 for Cisco Aironet). In future, if the wireless hardware supports a large number of power levels, the cost for this step can be limited through a combination of sampling and simple approximation techniques to determine the RSSI distribution of power levels. We leave such extensions as directions for future work.

2. **Deciding the feasible power levels:** At completion of Step 1, the wireless node X would have built an empirically tuned model for the different power levels, much like Figure 4.12. At this point, if the NKLD of distributions of any two power levels is greater then a threshold $NKLD_{thresh}$, then the two power levels are considered to be distinct and can be expected to yield desired distinguishable performance changes at the wireless client. In theory, dynamic programming can be used to determine the largest set of feasible power levels satisfying the above condition. For simplicity, we scan from the maximum power level to the lowest power level, picking all the power levels that satisfy the $NKLD_{thresh}$ criteria.

Figure 4.12 shows the distribution of RSSI for various receivers in our indoor WLAN deployment (Figure 4.2), when T1 is used as a transmitter and the power level is varied at the granularity of 10mW. The power levels are not shown in the graph for the sake of clarity. The top most plot is for receiver RB-10, followed by RB-11, RB-12 and RB-8. We use the steps outlined above to determine the feasible power levels for the aforementioned receivers. The distributions corresponding to these feasible power levels are marked in black in Figure 4.12. As can be seen, the selected power levels overlap minimally (NKLD ≥ 4). We also computed the error (captured by the NKLD function) between the accurate distributions and the distributions estimated by Online-RSSI. For each of these power levels, we found the error to be within **10** % of the desired maximum error. Clearly the amount of overlap (and hence the number of distinguishable power levels) depends

on the location of the receiver, with RB-10 observing less overlap as compared to RB-11, which practically observes only a single power level. These results clearly indicate that the set of feasible power levels is highly correlated with the location of the receiver and motivates the case for location-based power control, where the transmitter uses a different set of power levels for each client depending on the client's location.

4.3.2 Summary

For a given a wireless environment, our proposed model and its associated algorithms were able to accurately determine a good and useful set of power levels. The set of useful power levels as computed by Model-TPC are valid till traffic characteristics (other interference source) and wireless environments (physical obstacles etc) remain similar. Using our Online-RSSI algorithm, we already sample sufficient packets to reflect small scale changes in the wireless environments in our model. However the set of power levels must be recomputed against large scale changes in the wireless environment like transmitter mobility, introduction of a new physical obstacle or a new interference source. We are investigating various triggering mechanisms to refresh the Model-TPC, although a simple strategy to refresh the model every 10 minutes seems to work fine for our indoor experiments.

4.4 Experimental Evaluation of Model-TPC

To validate our model, we pick an existing algorithm [101] that uses transmit power control for improving client throughput and spatial re-use. The algorithm proposed increases transmit power in steps and measures signal quality to ascertain the optimal power setting for a given client.

At a high level, the algorithm operates as follows. It starts with the lowest power level and performs normal data rate adaptation using Onoe [17](a standard data rate adaptation mechanism). Once the data rate stabilizes around a value, the power level is increased and the rate adaptation process is continued. This process is repeated until the transmitter reaches the maximum rate available or reaches the highest power level.

To demonstrate the benefits of our proposed model, we create a set of useful power levels through Model-TPC and restrict the above algorithm to use only this set of power levels in its adaptations. We then compare the adaptation performance of the algorithm under two different scenarios – (i) which uses all possible power levels as available from the wireless interface, and does not use our model-TPC, and (ii) which uses the power levels provided by Model-TPC.

4.4.1 Setup

For the experiment described, the setup is identical to NLOS scenario, with the transmitter using an Atheros card having five power levels as validated by our power level validation setup in Figure 4.4. The mobile client continuously transmits data from itself to a departmental server located at the position of receiver R2, shown in Figure 4.2. The client roams from locations T1 to T2 to T3, which are annotated in the Figure 4.2 of our indoor WLAN deployment. Initially the client is at T1, which has 3 feasible power levels of 10mW, 20mW and 40mW, as per Model-TPC. After 12 seconds, the client goes to location T2, which is very close (LOS) to the server R2 and hence the client decreases its power level and is able to use the lowest power level of 10mW to achieve a data rate of 54Mbps. After 2 seconds, the client again moves to location T3, which has four feasible power levels as per our empirical model. We show the data rate and power adaptation process at T1 and T3 (The adaptation at T2 is obvious, with the client simply reducing power levels as it is very close to the server).

4.4.2 Results

There are two benefits of Model-TPC: First, it allows for significantly faster convergence for the transmitters to the best suited power level in their operating environments. Second, by eliminating the need to explore many redundant power levels with corresponding poor throughput performance, the transmitters achieve higher throughput over the entire adaptation duration. This is particularly important for clients that are mobile in nature and hence, need to adapt their transmission parameters, including power levels, quite frequently. We illustrate these gains through our reference implementation of the algorithm in [101], both with and without Model-TPC.

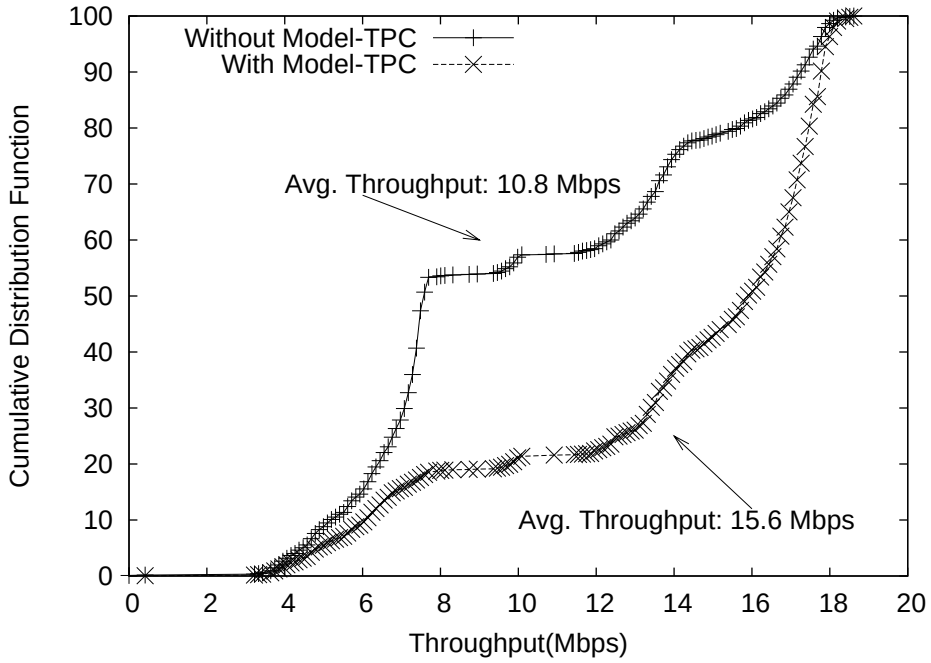
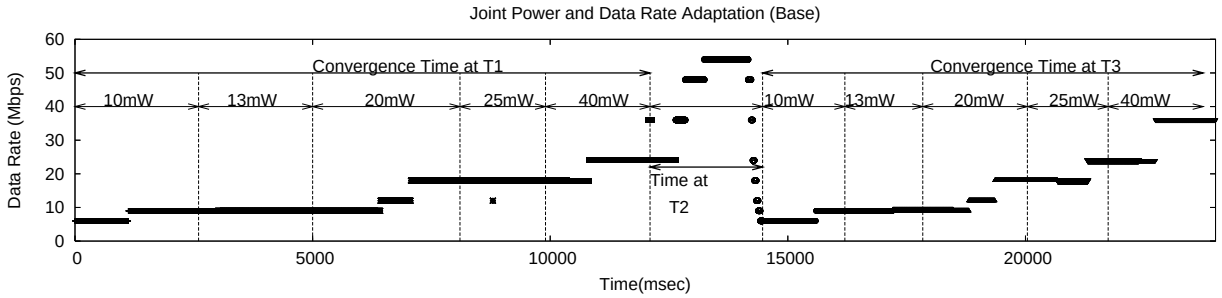


Figure 4.14: Cumulative distribution of throughput achieved by the wireless clients with/without the empirical model for adaptation at location T1. The average throughput for the adaptation process is also shown in the figure

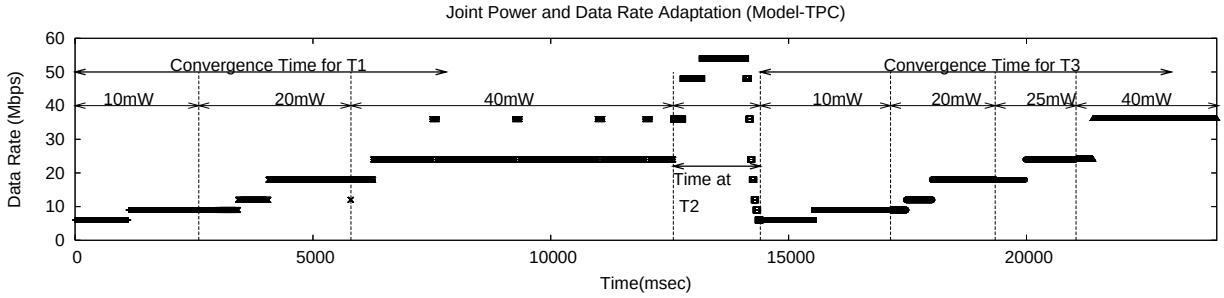
We first present the cumulative distribution function of the instantaneous throughput (measured every 100 ms) of the two variants of the transmit power control algorithm in Figure 4.14. The figure shows that using Model-TPC to restrict power levels leads to higher instantaneous throughput for a significant part of the experiment as shown in Figure 4.16. We explain this difference by examining the adaptation mechanisms in the two cases in Figure 4.15.

Figure 4.15(a) shows the adaptation behavior when all five power levels are used by the algorithm. We can see that over time, the algorithm attempts to identify signal quality at each different data rate and power level, spending a significant amount of time testing parameter values which are redundant for a given environment, thus impacting performance. In contrast, Figure 4.15(b) shows the adaptation with our Model-TPC. Clearly adaptation is much faster with our model, with more pronounced gains at T1 (as difference between hardware and feasible power levels is higher) than T3.

Note that here we only show the throughput gains arising from quicker convergence from a small power level to the right (greater) power level for locations T1 and T3. Model-TPC also provides much better convergence when adapting from a high power level to lower (right) power level as for T2, by skipping all the redundant high power levels in between. A faster convergence reduces the energy consumed in scanning high power levels and leads to energy savings, which is an important consideration for mobile clients.



(a) Without Empirical Model



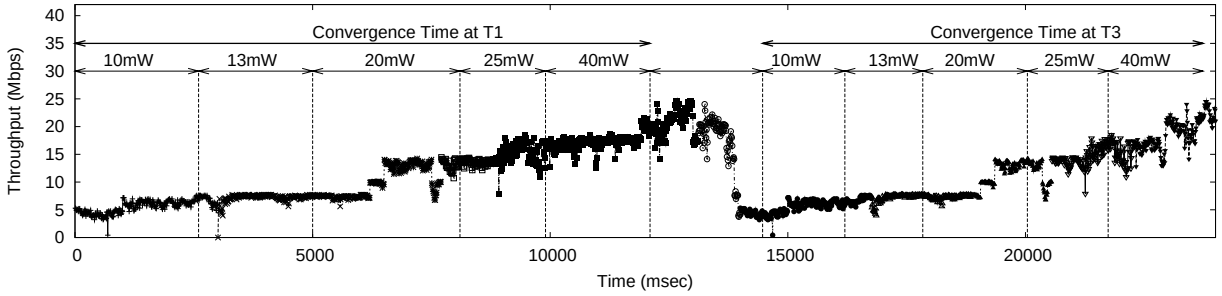
(b) With Empirical Model

Figure 4.15: Joint power and data rate adaptation mechanism with/without the empirical model. Convergence is much faster with the empirical model.

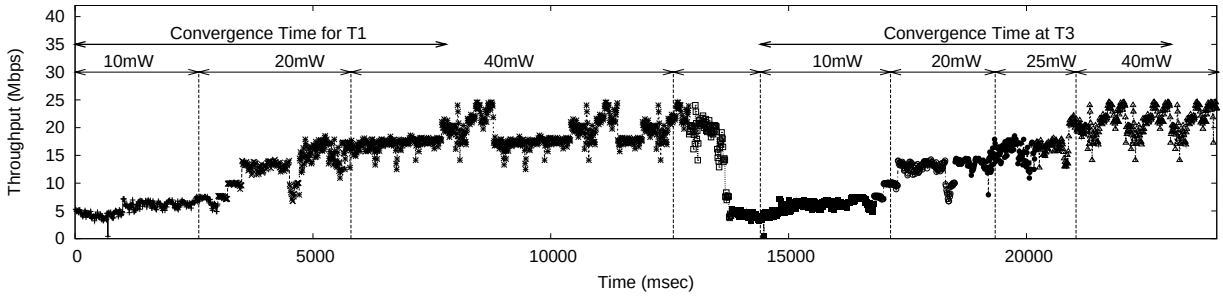
4.4.3 Summary

Our gains in the above wireless experiments stem from faster adaptation achievable when using the Model-TPC as an input to power control. Note that in our experiments, we compared benefits when only five power levels are available from the wireless interface. The performance gains of

Model-TPC will only be greater if the wireless interface makes more power levels available to the system software, that will clearly increase the number of redundant levels that the transmitter will scan in a typical power control algorithm, while our model will facilitate much faster convergence and performance.



(a) Without Empirical Model



(b) With Empirical Model

Figure 4.16: Goodput of the end wireless clients for joint power and data rate adaptation mechanism with/without the empirical model.

4.5 Discussion

While our work is targeted towards indoor WLANs, we discuss the feasibility of fine-grained power control in the context of cellular networks, where power control is again an important design parameter. Power control in cellular networks is used for reducing co-channel interference, managing voice quality, dealing with fast fading and near-far problem [69, 140]. The reason that

fine grained power control works in cellular networks is because of the following reasons: i) Cellular hardware is much better equipped to measure energy and distinguish between signal and interference. On the other hand, state-of-the-art wireless cards report only cumulative energy measurements from the frame header of the packet they receive and cannot distinguish between signal and interference. ii) Cellular networks are equipped with a fast control channel to perform reliable SINR measurements, which is not available to standard WiFi devices. iii) Cellular networks do not perform rate adaptation in the inner loop (real time or per packet basis) of power control, whereas data rate adaptation is an integral component of 802.11 based WLAN systems. Thus the SNR threshold for cellular networks is varied slowly in the outer loop of power control, whereas in WLANs, data rate adaptation is performed on very small time scales, thereby making RSSI variations even more critical for system performance.

The focus of this chapter has been in understanding what power control mechanisms are useful to design efficient power control algorithms. More specifically, we show that fine-grained power control cannot be effectively used by such algorithms in a systematic manner. In fact, our work suggests that a *few 3-5* discrete power level choices is sufficient to implement any robust power control mechanism in typical indoor WLAN environments. Through our work, we also build an empirical model that guides these appropriate number and choices of power values that is adequate. Our model can be used as a plug-in to previously proposed power control mechanisms, to make them implementable in real settings. We believe our work provides an important framework that can be used by researchers to develop robust and practical power control mechanisms. In the next chapter, we present the design, implementation and evaluation of an interference detection tool that can detect interference in real time and hence can facilitate the realization of interference mitigation mechanisms like transmit power control discussed in this chapter and centralized scheduling (Chapter 3).

Chapter 5

PIE : Passive Interference Estimation

In prior chapters, we describe data plane (CENTAUR - Chapter 3) and control plane (Model-TPC -Chapter 4) mechanisms to mitigate interference and ensure robust and predictable client performance in enterprise WLANs. The effectiveness of such interference mitigation mechanisms is contingent on their ability to estimate interference accurately in realistic settings. Although significant progress has been made in planning, deploying, and managing enterprise WLANs, administrators today have very limited tools that can help them understand how much interference exists in their network, and how such interference patterns are evolving over time. Such a tool for interference estimation can enable WLAN managers to improve the network performance by dynamically adjusting operating parameters like channel of operation and transmit power of access points, but also diagnose and potentially proactively fix problems. Prior work on interference estimation employs active probing techniques and suffers from three main problems: a) it incurs moderate to significant measurement overhead and cannot be employed to continuously obtain interference information as they evolve over time, b) it offers limited visibility into the root cause of interference, c) it often requires specific client support.

Building an on-line, real-time tool for enterprise-wide WLAN interference is particularly challenging because interference is changing all the time. Each time a new client arrives, departs, moves, or changes its own traffic pattern, the number of other nodes in the network it interferes with (and the degree to which it interferes) changes. Further, wireless channel conditions are never static but continuously evolve with changes in the environment, e.g., even with the opening or closing of a door, people walking, etc. The goal of this work is to answer the following question

for this highly dynamic environments:

Given an enterprise WLAN consisting of a number of APs and a variable number of mobile clients, with highly variable and uncontrolled active periods, describe its conflict graph, i.e., identify the precise set of nodes that interfere with each other and the degree to which they do so, as their interference patterns continuously evolve over time.

As the final contribution of this thesis, we present the design, implementation and detailed evaluation of a Passive Interference Estimator (PIE) that can dynamically generate fine grained interference estimates across an entire WLAN. The most attractive feature of PIE is that it imposes no measurement traffic, and yet provides an accurate estimate of WLAN interference as it changes with client mobility, dynamic traffic loads and varying channel conditions. Our experiments conducted on on two different testbeds show that PIE is able to not only provide high accuracy but also operate beyond the limitations of prior tools, providing a true solution to performance diagnosis and real time WLAN optimization, as manifested through its use in multiple WLAN optimization applications, namely channel assignment, transmit power control, and data scheduling.

The rest of the chapter is organized as follows. Section 5.1 discusses the motivation for developing a real time interference estimation tool. The fundamental principles behind PIE are described in Section 5.2. We describe the design and operation of PIE in Section 5.3. Finally we present our evaluation of PIE in Section 5.4 and show its usefulness for end clients by efficiently integrating it with different interference mitigation mechanisms 5.5.

5.1 Motivation

The problem of interference estimation is fundamental to understanding the behavior of any wireless network. When a new network is planned and deployed, a possible goodness metric of the deployment is the degree of interference observed in it over time. Since interference might be quite transient based on the distribution of users and their traffic patterns, a one-time or infrequent measurement of interference may not be effective. If a tool can capture *all* actual interference in

the network in a real-time and on-line fashion, then the administrators can validate the goodness of their deployment. In addition, if the interference of the network is perceived to be too high, such information can help them identify how to adapt their deployment to improve performance in response to usage patterns, e.g., by placing new APs or re-configuring existing APs.

In addition, interference estimates and the conflict graph serve as an important input to many WLAN configuration problems, e.g., channel assignment for each AP, transmit power selection for these APs, and even emerging strategies of data scheduling across the enterprise WLAN [149].

Given the challenging nature of this problem, a number of recent research efforts have made significant progress towards this tool building goal. Some of the recent approaches (e.g., Interference maps [123] and Micro-probing [24]) inject active traffic into the enterprise to infer occurrences of interference. While such approaches may be fairly accurate in determining interference, the overheads of making continuous measurements across the entire WLAN might inhibit their adoption.

In this chapter, we explore an alternate design for a practical online interference estimation mechanism, one that does not introduce any active measurement traffic on the WLAN, i.e., is completely passive in nature, and creates all its interference estimates by simply observing ongoing traffic at the different APs alone. More specifically, we present the design, implementation, and detailed evaluation of the Passive Interference Estimator system (or PIE).

Our work in online interference estimation is inspired by two key passive WLAN monitoring approaches proposed earlier, namely Jigsaw [44, 43] and WIT [106]. Such prior work provides us with two useful building blocks: (i) a platform for capturing wireless traffic and merging traces collected from different vantage points and (ii) some specific tools to infer some interesting properties about the 802.11 network from such merged traffic traces. However, both these research efforts stop short of addressing our goal of designing a real-time interference estimation tool as it evolves in time across the entire enterprise WLAN. PIE is such a tool and its unique features include:

1. Captures dynamic interference information quickly and robustly: PIE captures wireless interference information across the entire WLAN quickly even with client mobility and changing channel conditions, i.e., within a few hundred milliseconds, and robustly, i.e., can effectively distinguish between true and false interferers when multiple overlapping transmitters are present.

2. Based on real traffic patterns: Unlike the active measurement techniques, PIE is a completely passive scheme, and its interference estimates are based purely on actual traffic patterns in the network, and capture all effects of PHY transmission rate adaptation, packet sizes, traffic interarrival times, etc.

3. Low overhead and no network downtime: Being passive, PIE incurs no overhead on the wireless spectrum and does not take away wireless bandwidth resources from its users.

4. Does not require additional infrastructure or client support: The PIE mechanism is implemented purely using APs and a central controller (placed within the enterprise wired network). No additional infrastructure component is required. In addition, there is no requirement of any support in clients.

We believe our work bridges the gap between passive monitoring and its practical usability, by designing a real-time interference estimation mechanism based on passive monitoring.

5.2 Interference estimation in PIE

Interference in an enterprise WLAN can be broadly classified into two categories: (a) sender-side interference caused due to carrier sensing between two transmitters, and (b) receiver-side interference caused due to collision of simultaneous transmissions at the receiver. While carrier sensing determines how the transmitters share the wireless medium, collision induced interference determines whether transmissions are successfully decoded at the intended receiver. The goal of PIE is to identify both of these interference properties in a non-intrusive manner. We now explain the intuition behind PIE with the help of a simple example.

Intuition behind PIE: Consider a scenario from an enterprise WLAN (shown in Figure 5.1) where APs A and B are far enough apart such that they cannot carrier sense (CS) each other. Assume that two clients C_A and C_B are associated to APs A and B respectively. Suppose some downlink

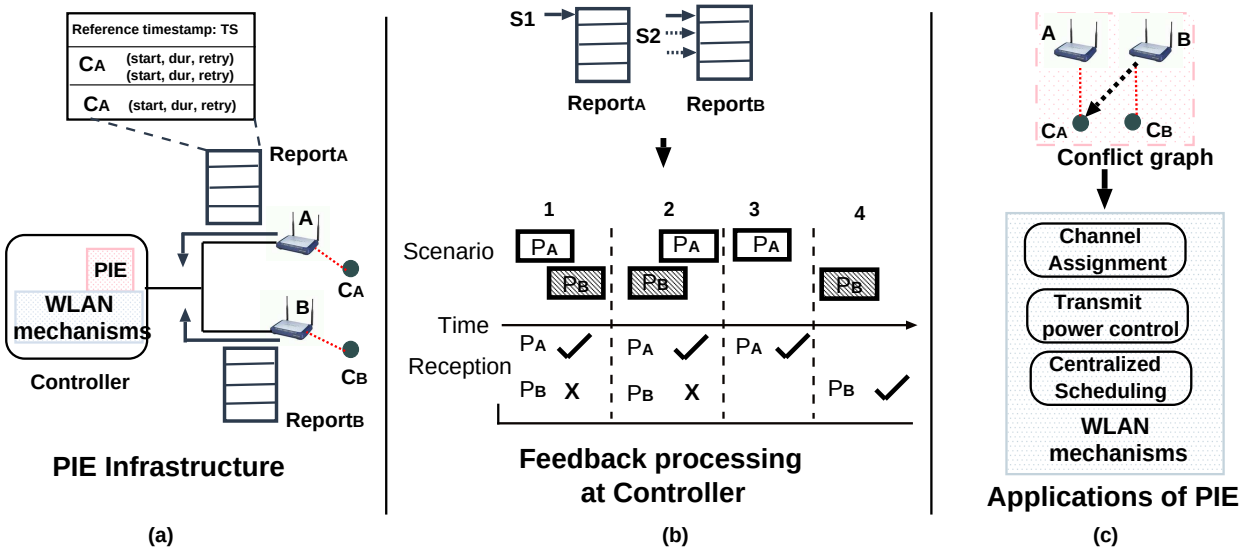


Figure 5.1: Overview of PIE, showing the overall infrastructure, the feedback processing performed at the Controller and the integration of PIE with channel assignment and scheduling. The detection of conflict between AP B and client C2A i) places the two APs in separate channels when channel assignment is performed, or ii) serializes the transmissions between AP A and B.

packets being forwarded to the APs A and B, for transmission to their respective clients, C_A and C_B. The APs follow the regular 802.11 carrier sensing mechanism, and transmit to their clients whenever possible.

In PIE, APs A and B would periodically send their frame transmission timestamps to the controller. Further, the frames are tagged with their reception status indicating whether this frame transmission was successful or not (i.e., whether the AP has received an ACK for this or not). The controller parses these timestamps and identifies the four scenarios shown in Figure 5.1(b). Looking at scenarios 1 and 2, the controller observes that frame transmissions from A and B (denoted by P_A and P_B) overlap in both directions, indicating that A and B do not defer their transmissions for each other, and hence are not in the carrier sense range of each other. Additionally, the controller can also infer that whenever a transmission for client C_B overlaps with a transmission by AP A, then C_B is not able to decode the transmission (i.e., P_B is lost). On the other hand,

transmissions for C_A are not lost despite overlapping transmissions by AP B. Hence the controller concludes that AP A interferes with link (B, C_B) but B does not interfere with (A, C_A). The controller can then leverage this information to efficiently mitigate interference for C_B. For example, it can perform downlink data scheduling ([149]) and allocate different time slots to (A, C_A) and (B, C_B) transmissions. Alternatively, the controller can also assign different channels to APs A and B, thereby allowing both transmissions to proceed simultaneously without any interference. As this example demonstrates, having accurate interference estimates could enable the controller to improve client performance in an enterprise WLAN by employing interference mitigation mechanisms effectively. We now give a detailed explanation of how PIE identifies these interference properties in a non-intrusive manner.

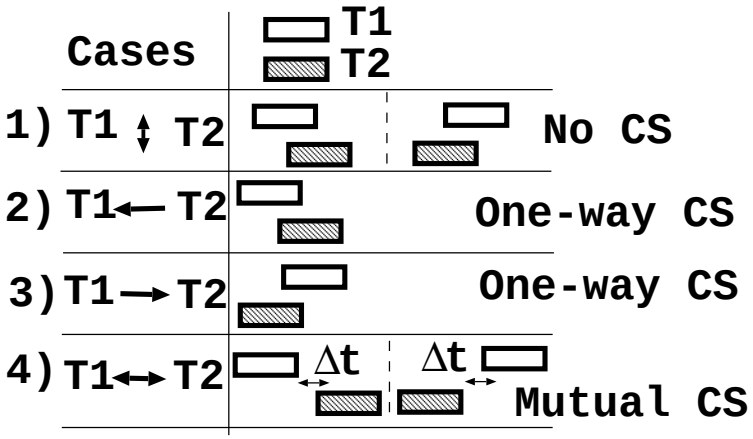


Figure 5.2: Detecting the carrier sense relationship between two links on the basis of timestamps of transmissions by the two transmitters A and B. Timestamps refer to the MAC timestamp of wireless frames as reported by the wireless card.

5.2.1 Estimating carrier sense (CS) interference

PIE identifies the carrier sense relationships based on the order in which competing transmitters access the wireless channel. Figure 5.2 shows the possible order of channel access for different carrier sensing relationships. As shown in the Figure, there can be four cases of channel access:

(a) Overlapping frame transmissions (Cases 1, 2 and 3): Case 1) When two competing transmitters are not in carrier sensing range, they can access the channel in any order and hence the controller would observe that their frames overlap in both directions. Case 2,3) In case of one-way carrier sensing, the frames will only overlap in one direction. For example, if $T_1 \rightarrow T_2$ (i.e., T_1 carrier senses T_2) then T_1 will defer for T_2's transmissions. However, T_2 will not defer for T_1's transmissions, and would transmit even if T_1's frame is still in the air. Hence the controller should only observe overlaps where T_1's transmission is already in the air and is overlapped by a later T_2 transmission.

(b) Non overlapping transmissions (Case 4): If both the transmitters can mutually carrier sense each other, the controller should not see any overlaps as carrier sensing will serialize their frame transmissions. However, we note that non overlapping transmissions may also be observed in scenarios where the two transmitters do not simultaneously contend for the channel, and transmit their frames one after another due to their specific traffic patterns. In such a scenario, it is difficult to make any inference regarding the carrier sense relationship of the two transmitters. In order to distinguish the cases where transmitters are actually contending for the medium, we use the mechanism outlined in [106]. The controller labels a pair of frames as being transmitted by "contending" transmitters if their starting timestamps are within a δ_t time interval, where δ_t is the total time that can be spent by competing transmitters performing back-off. Prior work in [106] shows that using the duration of average 802.11 contention window as δ_t correctly classifies the contending transmitters in the system. Accordingly, we use a value of $\delta_t = 50 + 320\mu sec$ (DIFS + Max back-off period for 802.11g) to identify transmissions that are considered to be competing for channel access. The pseudo-code for estimating carrier sense properties in PIE is shown in Algorithm 5 (Procedure ComputeCS).

5.2.2 Estimating collision induced interference

PIE identifies collision induced interference at the receiver by computing the probability of a frame loss at the receiver when it overlaps with a simultaneous transmission from a competing transmitter. Intuitively, the extent of interference is directly proportional to the probability of losing

Algorithm 5 PIE : CS and INT computation

Procedure ComputeCS:

Inputs: number of frames in contention n_c, number of case (3) overlaps n_f, and number of case (2) overlaps n_r, cs threshold δ_t

n_o = $n_f + n_r$

n_n = $n_c - n_o$

if ($(\frac{n_n}{n_c} > \delta_t)$ || $(\frac{n_o}{n_c} > \delta_t)$) **then**

/* cases 1, 4 */

return $(\frac{n_n}{n_c}, \frac{n_n}{n_c})$

if ($(\frac{n_f}{n_c} > \delta_t)$ || $(\frac{n_r}{n_c} > \delta_t)$) **then**

/* cases 2, 3 */

return $(\frac{n_f}{n_c}, \frac{n_r}{n_c})$

else

/* inconclusive (wait for more samples) */

return $(-, -)$

Procedure ComputeINT:

Inputs: total number of frames n_p, number of frames lost n_l, number of overlapping frames n_o, number of overlapping frames lost n_{ol}

l_{iso} = $(n_l - n_{ol})/(n_p - n_o)$ /*loss in isolation*/

l_{int} = n_{ol}/n_o /*loss under interference */

LIR = $(1 - l_{int})/(1 - l_{iso})$

return LIR

overlapping frames. Note that this allows PIE to maintain a continuous interference model, where the extent of interference can be any value between 0 and 1. Such a model is better suited for realistic environments where the binary model of interference may not suffice. On the basis of this observation, in PIE, we use Link Interference Ratio (LIR) described below, as the metric to quantify interference for a link.

Link Interference Ratio (LIR): For a pair of interfering links, LIR captures the loss in performance observed when the two links are interfering as opposed to operating in isolation. Consider a link (A, B) and its interferer C. We measure D_{AB}, the delivery probability of the link (A, B) in isolation (A is active, C is inactive). We then measure D^C_{AB}, the delivery probability of the link

when interferer C is also active with A. The LIR is given by:

$$LIR = D^{C}_{AB}/D_{AB} \tag{5.1}$$

LIR takes values between 0 and 1. LIR of 0 means that link (A, B) cannot deliver frames in the presence of C, while LIR of 1 means that C does not impact link (A, B). LIR values between 0 and 1 indicate the extent of interference on link (A, B) by interferer C. When A and C are in carrier sense range, LIR will be equal to 1, since the interferer C is able to share the channel with the transmitter A without causing any decrease in the delivery ratio of link (A, B) [1]. The pseudo code for estimating interference is shown in Algorithm 5 (Procedure ComputeINT).

5.3 PIE Design and Operation

In this section, we formally describe the design and operation of PIE which adheres to the key requirements for effective and practical interference estimation, as outlined in Section 5.1. Our description focuses on PIE in the context of an enterprise WLAN and a schematic overview of the overall design can be seen in Figure 5.1. Next, we discuss the design and functioning of three key components of PIE .

Sniffing at the APs: In our current implementation of PIE sniffing of the wireless medium is limited to the APs in the enterprise WLAN. This allows us to avoid the additional overhead associated with the deployment and management of extra sniffers in the enterprise building. However, sniffing solely at the APs might result in reduced coverage of uplink client traffic, as compared to a dense sniffer deployment (e.g., as in Jigsaw [44]). In order to overcome this limitation, we employ the finite state mechanisms outlined in [106] (based on 802.11 states) to infer some of the missing client transmissions.

PIE requires accurate timestamp information for accurate interference estimation. However, due to limitations of the existing Atheros driver and firmware, it is difficult to extract the exact

[1] Note that this measure of LIR differs slightly from the interference metric proposed in [125], that relies on effective throughput and not delivery probability. However, throughput based LIR is ambiguous for carrier sensing scenarios, where a LIR value of 0.5 could mean 50% loss or carrier sensing. Hence we use delivery probability as it provides greater clarity into the LIR values in all scenarios.

time at which a packet is transmitted in the medium[2]. In order to overcome this problem, in our implementation of PIE, APs are equipped with two radios: one radio is used for normal packet transmissions and receptions, while the other radio is used for capturing packets in the wireless medium. The Atheros driver timestamps every frame that is received over the interface using an on-board 64-bit microsecond resolution timer. Thus a second radio that captures packets can record the exact timestamp of the packet transmission. Moreover, the proximity of the two radios ensures that the second radio receives every frame transmitted by the AP due to capture effect.

Synchronization of clocks at the APs: PIE needs the APs to synchronize their clocks so that the controller can compare their packet transmission reports and determine the extent of overlap between any two transmissions reported by the APs. Further, time synchronization should be tight to allow accurate 802.11 analysis (of the order of 20-30 μsec [44]). Prior mechanisms for 802.11 analysis [44, 43, 106, 169] synchronized the APs by finding common beacon packets in their transmission reports. However, performing such offline synchronization at the controller can be time consuming, and non-realistic for a real time interference estimation mechanism. To synchronize the clocks across the APs, we use the time synchronization protocol implemented by Atheros driver [14]. As part of the protocol, the AP embeds a 64-bit micro second granularity time stamp in every beacon frame, and the nodes that listen to the AP adjust their local clock based on this broadcasted timestamp [74]. In order to make this synchronization seamless, we set up a virtual ad-hoc interface on the second radio of each AP. Now all the APs that join the ad-hoc network, synchronize themselves in real time using the beacons of the reference AP for the network. This approach has two key benefits:

- It is an online mechanism, meaning the nodes synchronize their clocks every time the beacons are received from neighboring nodes. Syncronization accuracy can be tuned by varying the beacon period.
- It is transitive in nature, and works as long as the network is not partitioned.

[2]This is because once the driver passes the packet to the firmware, a variable delay is introduced based on the length of the firmware transmit queue and the amount of time the radio performs carrier sensing/back-off. Further, retry and other 802.11 packets (like beacons) are handled solely by the firmware, making timestamp estimation more challenging.

In PIE, we take advantage of the Ethernet backplane to synchronize APs across network partitions. A similar approach has been proposed in [23] to achieve network-wide synchronization. Note that it suffices to synchronize the clocks of the reference APs (one in each network partition). In order to do so, the controller periodically broadcasts a sync packet over the wired backplane to these reference APs. The reception of sync packets serves as a synchronization event and on receiving the sync packet, the APs report the value of their local clock to the controller. The controller then computes the drifts for the set of reference APs according to a global reference AP. These drifts are communicated back to the APs which then adjust their local clocks accordingly. We note that the synchronization accuracy depends on whether the reference APs receive the sync packet at the same time instant. Our measurements revealed that the controller-to-AP path can introduce some variable delays, on the order of 10-12 μsec. We benchmark the overall synchronization error in PIE by periodically probing the APs for their local clock values and computing the maximum difference in clock values for any given probe. Figure 5.3 shows the distribution of synchronization error for 20 APs in Testbed 1. We observe that for 90% of the probes, synchronization error is within ± 23 μsec, which is sufficient for the purposes of PIE .

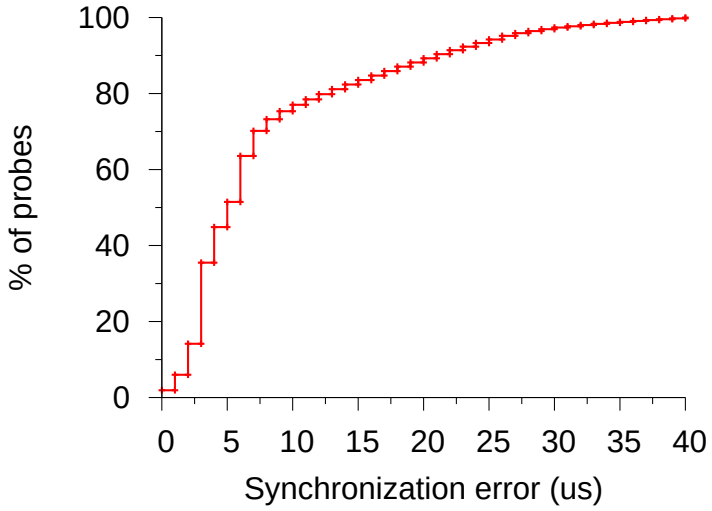


Figure 5.3: Distribution of maximum clock error across 20 APs in Testbed 1.

Collecting feedback from the APs: In PIE the Controller periodically polls the APs for their transmission reports. The granularity of polling is a tunable parameter, which can be determined empirically. Lower polling periods will enable PIE to update interference estimates faster. On the other hand, increasing the polling period allows APs to sample more packets per transmission report, increasing the accuracy of interference estimates. We evaluate this tradeoff in Section 5.4 and show that a polling period of ∼100ms achieves a good balance between accuracy and responsiveness.

Feedback processing at the Controller: As discussed in Section 5.2, PIE uses packet overlaps between any two links to estimate the carrier sensing and collision induced interference. Algorithms 6 and 7 shows the pseudo-code for processing sniffer reports and updating interference estimates in PIE . Overlap between packets is computed by the Controller, which merges the transmission reports sent by the APs and iterates over all the packets in the combined sorted list, maintaining a carrier sense estimate for every pair of transmitters and a collision interference estimate for every link-interferer pair. As shown in Algorithm 6, feedback processing takes O(m^2n) time, where m is the number of APs and n is the number of packets per AP[3].

Putting it all together: Figure 5.4 shows the overall working of PIE . As shown in the Figure, APs periodically send their sniffer reports back to the centralized controller, that computes the carrier sense and interference relationships by merging these sniffer reports. The controller computes the overlap between different AP-client pairs and determines the impact of such overlap on the performance of the clients. The controller then updates the interference estimates for different (AP-client, Interferer) pairs in the system (Algorithm 7). As shown in the Algorithm, PIE uses the instantaneous interference estimates (computed in each polling period) to update the overall interference estimate for a AP-client pair using EWMA. Specifically, the interference estimate for a given AP-client pair with respect to different interferers in the system is updated as follows:

$$\begin{aligned} LIR(AP-client, Interferer) &= (1-\alpha) \times LIR_{old}(AP-client, Interferer) \\ &\quad +\alpha \times LIR_{new}(AP-client, Interferer) \end{aligned} \tag{5.2}$$

[3]Since the transmission report by each AP is already sorted, the overhead of merging at the controller is minimized

Algorithm 6 PIE : Feedback Processing at Controller

Procedure ProcessReports():

Input: Set of transmission reports M, one from each sniffing AP. Each report $m_i = (start, end, src, dst, rate, loss)$. Set of transmitters T and set of links L.

Initialize: $n_p[l_i] \leftarrow 0$, $n_l[l_i] \leftarrow 0 \forall l_i \in L$, $n_o[l_i, t_j] \leftarrow 0$, $n_{ol}[l_i, t_j] \leftarrow 0 \forall l_i \in L, t_j \in T$

for each report $m_i \in M$

$S = \{S_t | S_t \leftarrow m_j.start \forall j \neq i\}$

while $S_i \neq m_i.end$

do

$l_i \leftarrow (S_i.src, S_i.dst)$, $n_p[l_i]$++

if $S_i.loss$ **then** $n_l[l_i]$++

for each $S_j \in S$

do

while $S_j.start\_tx > S_i.end\_tx$

do

$overlap_{ij}$ = ComputeOverlap(S_i, S_j)

if $overlap_{ij} > 0$

Overlap[l_i, $S_j.src$]++

if $S_i.loss$ **then** $n_{ol}[l_i, S_j.src]$++

$n_c[S_i.src, S_j.src]$++

if $S_i.start > S_j.start$

$n_f[S_i.src, S_j.src]$++

else

$n_r[S_i.src, S_j.src]$++

else if $|overlap_{ij}| < \delta(320\mu sec)$

$n_c[S_i.src, S_j.src]$++

S_j++

done

S_i++

done

done

UpdateEstimates(n_p, n_l, n_o, n_{ol});

Algorithm 7 PIE : Updating Interference Estimates at Controller

Procedure UpdateEstimates():

Input: $\forall l_i \in L$ total number of packets $n_p[l_i]$ and total losses $n_l[l_i]$.

$\forall l_i \in L, t_j \in T$ total overlapping packets $n_o[l_i, t_j]$ and total overlapping packets lost $n_{ol}[l_i, t_j]$

for each link $l_i \in L$

  for each transmitter $t_k \in T$

    Interference[l_i, t_k] = (1 - α) $\times$ Interference[l_i, t_k] + $\alpha \times$ ComputeINT($n_p[l_i]$, $n_l[l_i]$, $n_o[l_i, t_k]$, $n_{ol}[l_i, t_k]$)

for each transmitter $t_i \in T$

  for each transmitter $t_j \in T \; \forall j \neq i$

    CarrierSense[t_i, t_j] = ComputeCS($n_c[t_i, t_j]$, $n_f[t_i, t_j]$, $n_r[t_i, t_j]$)

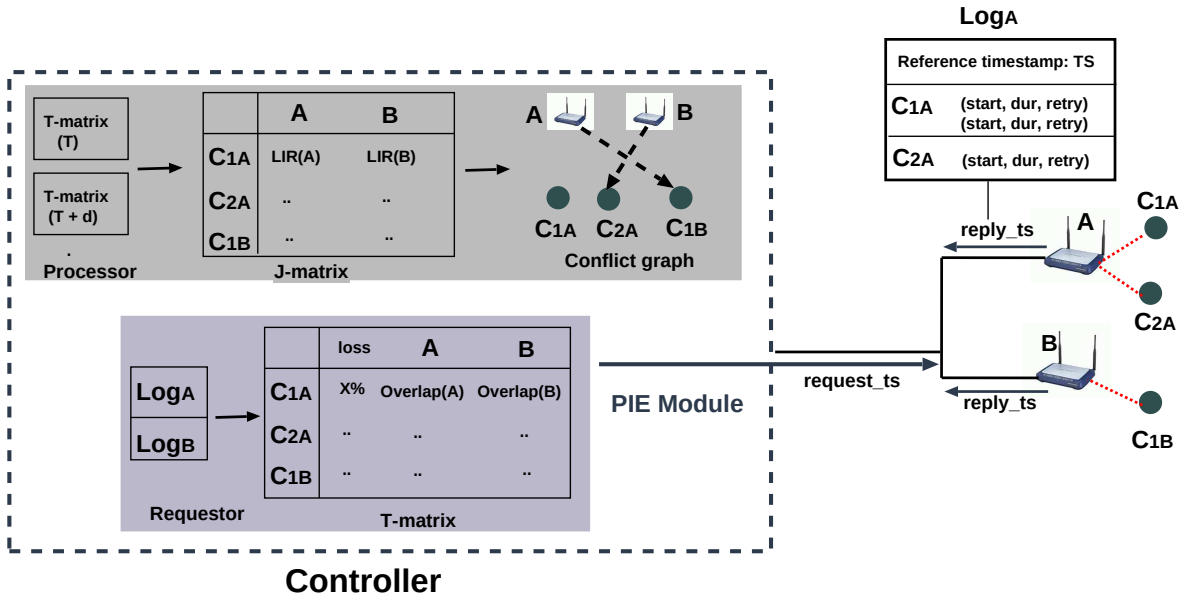


Figure 5.4: Overview of PIE, showing the overall infrastructure and the feedback processing performed at the controller. As shown in the figure, the controller updates the interference estimates with every new set of reports received during a polling period

where LIR_{old} is the interference estimate that controller currently has for the (AP-client, Interferer) pair from earlier reports and LIR_{new} is the latest estimate that the controller measures from feedback received in the most recent polling period. We use a value of $\alpha = 0.75$ so that the conflict estimate is heavily weighted towards the most recent interference pattern observed in the

network. Such a high forgetting factor allows the wireless controller to react aggressively to the changing interference patterns and is more suited for dynamic wireless environments, where such interference estimates need to updated very rapidly to reflect a correct view of the interference in the system.

5.4 Evaluation of PIE

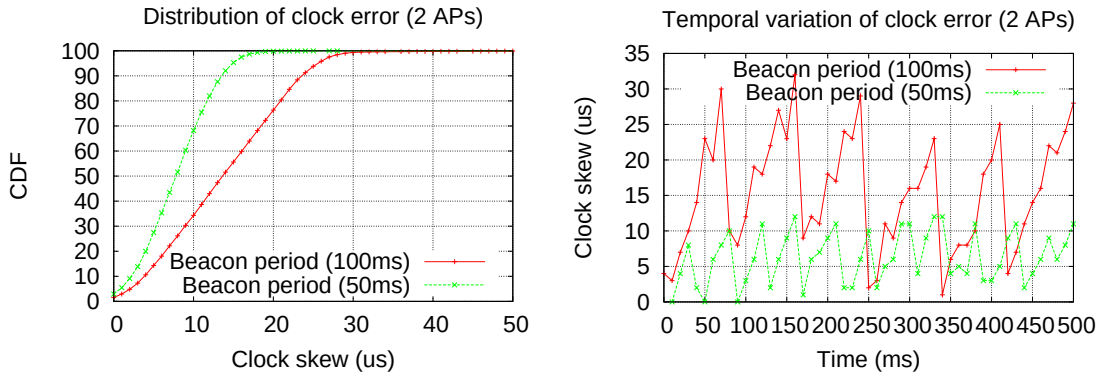
In the previous sections we motivated the need for online interference estimation and described the design of PIE. To further validate such a design we break the evaluation section into three distinct sub-sections. First, we are going to demonstrate that PIE accurately captures interference in real time. Such an evaluation is going to be done in comparison with today's state of the art approach, that of bandwidth tests. However, PIE goes beyond the capabilities of bandwidth tests enabling unprecedented visibility into wireless interference in real time, while capturing the effect of transmission rate, packet size, and traffic workload. As a result, our second sub-section will focus on demonstrating the added value provided by PIE compared to prior approaches, that rely on broadcast probing traffic to capture interference. Lastly, our third sub-section will integrate PIE with a number of real time WLAN optimization mechanisms to offer evidence on the usefulness of PIE in real time diagnosis and operation of a WLAN.

Setup: The entire section is based on PIE's evaluation on two different testbeds. We run our central controller on a standard Linux PC (3.33 GHz dual core Pentium IV, 2 GB DRAM) (in about 3,000 lines of C code and a few hundred lines of Perl script), and Soekris (Testbed 1) as well as VIA-based (Testbed 2) wireless APs, modified slightly to improve path latencies. Each node in the two testbeds is equipped with two Atheros AR5212 chipset wireless NICs. Unless otherwise specified, we perform these experiments in 802.11a to prevent interference from departmental WLAN.

5.4.1 Accuracy of time synchronization in PIE

Accuracy of interference estimation also depends on the level of time synchronization that can be achieved in PIE . In order to understand the extent of time synchronization that can be achieved in PIE , we perform targeted experiments in a testbed comprising of 19 wireless APs, each equipped

with two radios. As discussed in Section 5.3, we put one radio on each node in monitoring mode and the other radio in ad-hoc mode, so that the nodes are synchronized using the beacon messages from the reference AP in the ad-hoc network. In order to assess the level of synchronization, each AP transmits a probe periodically and all other APs that hear the probe, timestamp it with their local clock and send a log with their local timestamps of the probes to the WLAN controller. Now the controller computes the difference in timestamps as reported by different APs for the same probe. We define $Dispersion$, as the maximum difference in the timestamps for a given probe packet [4].



(a) Distribution of clock skew

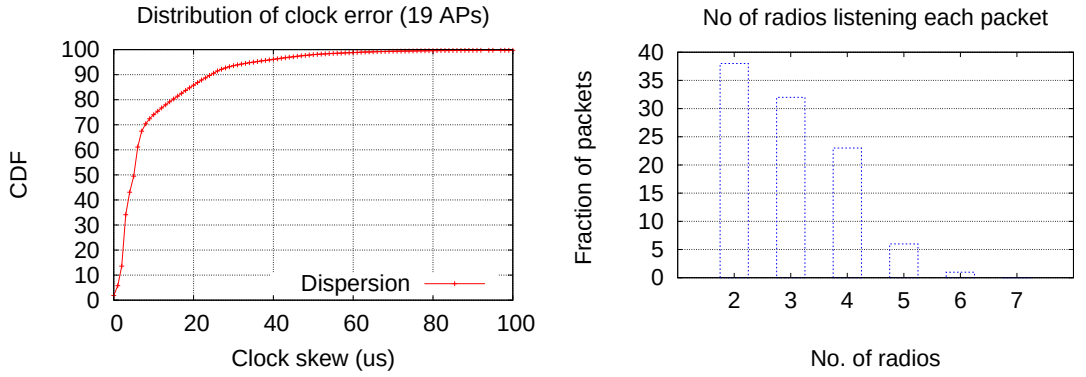
(b) Temporal variation of clock skew

Figure 5.5: Analyzing clock skew using beacon based synchronization. (a) shows the distribution of clock skew for measured every 10ms for a pair of APs. Notice that clock skew is much smaller for a 50ms beacon period as compared to 100ms beacon period, as 50ms beaconing allows the APs to synchronize twice as frequently. (b) shows the temporal variation of clock skew for the AP pair. Again notice that the clock skew shows a periodic behavior. It keeps on increasing withing a beacon period and minimizes at each beacon interval when APs synchronize using a beacon. Periodicity of clock skew is very close to the beacon period used in that experiment.

Impact of beacon period on synchronization: Figure 5.5(a) shows the distribution of dispersion over two APs for a duration of 3 minutes, where each AP transmits $18,000$ packets (once every

[4]This is similar to the notion of dispersion used in [44]

10ms) during that duration. We observe that the 90th percentile of dispersion is less than 24μsec and 14μsec for beacon periods of 100 and 50 ms respectively. As shown in the figure, the median dispersion for most APs is much less then 20μsec, required for accurate passive analysis of 802.11 packets. We conclude that using a beacon period of 100ms, provides us with sufficient accuracy to perform the passive analysis. Further, Figure 5.5(b) shows the temporal variation of clock skew for the AP pairs. Notice that the clock skew keeps on increasing within a beacon period and minimizes near the beginning of every beacon interval [5]. Beacon period could be further reduced to improve the accuracy of synchronization, although that will increase the overheads of beacon transmission and might not be desirable for dense WLANs.



(a) Distribution of clock skew (19 APs)

(b) Number of radios hearing each transmission

Figure 5.6: Analyzing clock skew using beacon based synchronization for larger deployments. (a) shows the distribution of clock skew for measured every 200ms for 19 APs using the experimental setup described before. (b) shows the number of radios that hear any given transmission during the experiment. Notice that large fraction of transmissions are heard by only 2 or 3 radios, which is expected if we only perform monitoring at the APs.

Synchronization across larger topology: Figure 5.6(a) shows the maximum clock skew (dispersion) for each probe packet transmitted by the 19 APs during the experiment. Figure 5.6(b) shows

[5] Beacon messages at the beginning of each beacon interval synchronizes the clocks of different APs, minimizing the clock skew.

the number of radios that hear any given transmission during the experiment. Notice that large fraction of transmissions are heard by only 2 or 3 radios, which is expected if we only perform monitoring at the APs. Finally, figure 5.7 shows the dispersion categorized by the probes sent by different APs. As shown in the figure, the median dispersion for most APs is much less then 20μsec, required for accurate passive analysis of 802.11 packets. We conclude that using a beacon period of 100ms, provides us with sufficient accuracy to perform the passive analysis. Beacon period could be further reduced to improve the accuracy of synchronization, although that will increase the overheads of beacon transmission and might not be desirable for congested WLANs.

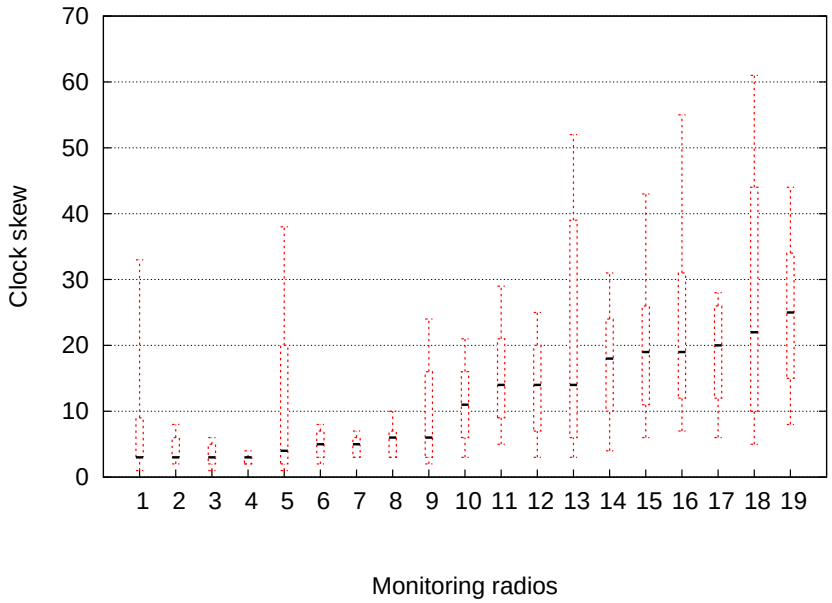


Figure 5.7: Dispersion error observed for synchronization probe packets transmitted by different APs in the system using a beacon interval of 100 ms. 90 and 10 percentiles are shown with the error-bars, while the 75 and 25 percentile is shown by the box.

5.4.2 Accuracy of interference estimation in PIE

Accuracy is a key requirement for an interference estimation mechanism. We evaluate PIE's accuracy using two different methods. First, we construct all possible conflict scenarios using a canonical two link topology. This experiment serves as our controlled experiment that does

not only allow us to assess accuracy but also focus on the underlying phenomena causing any discrepancies between PIE and bandwidth tests. Second, we generalize our findings across a large scale testbed, quantifying PIE's overall accuracy. Overall accuracy is further evaluated across a number of dimensions that take into account diverse transmission rates, packets sizes, interference scenarios, and density.

Metrics for comparison: Both experiments are evaluated according to the Link Interference Ratio (LIR) described in Section 5.2.2. LIR is the ratio of the frame delivery probability of a link (A, B) under interference from C and in isolation (D^C_{AB}/D_{AB}).

Compared schemes: We compare three approaches that measure LIR with differing levels of overhead.

1) Unicast bandwidth tests (Ground truth): The traditional approach, followed by the literature, is to use unicast bandwidth tests (UBT) to determine the impact of an interferer on a link [125]. In unicast bandwidth tests, A transmits unicast packets to B in isolation and under interference from C. We then report LIR as the ratio of frame delivery probabilities under the two scenarios. This is the most accurate test to determine LIR as it uses unicast traffic, which takes into account the impact of C on the receiver (data packet collisions) and the sender (ack collisions). Henceforth, we use the LIR value reported by unicast bandwidth tests as the "ground truth" in our experiments. Note that UBT incurs significant overhead – it takes O(n^4) measurements to compute a conflict graph for n link topology using unicast bandwidth tests, and hence is not practical to use under dynamic wireless environments.

2) Broadcast bandwidth tests : In broadcast bandwidth tests (BBT), broadcast traffic from A to B is used to compute the frame delivery ratios, both in isolation and under interference from C. This method was proposed as a relatively fast way to measure interference relationships among a large number of links [125]. Broadcast tests can compute the conflict graph for a n link topology in O(n^2) measurements (as opposed to O(n^4) for UBT). However, broadcast tests do not take data-ack collisions into account and hence may be inaccurate in some scenarios.

3) PIE : PIE computes the LIR value in a passive fashion by determining the conditional loss probability of packets on link (A, B) that are interfered by interferer C. A packet P_i on link (A, B)

is considered to be interfered if it overlaps with a transmission from interferer C that leads to packet loss. The LIR in this case is computed by passively observing the events in the wireless medium as recorded at the controller. Psuedocode for PIE is shown in Figure 5 (function ComputeINT).

In what follows, all experiments are performed in 802.11a to prevent interference from the co-located department WLAN that operates on 802.11g. Furthermore, the PIE measurements are collected passively through the observation of the probe traffic generated by the bandwidth tests.

5.4.2.1 Static interference settings

We start by comparing the LIR generated by the three mechanisms for different canonical scenarios, as shown in Table 5.1. In order to have a fair comparison, we first evaluate the accuracy of PIE under static data rate (6Mbps) and packet size (1400 bytes) settings, as the overhead for computing LIR for dynamic (client mobility, variable rates) can be significant for bandwidth tests. We then relax these constraints and evaluate the performance of PIE under dynamic interference scenarios triggered by client mobility, the use of variable transmission rates and packet sizes.

Controlled experiments: Using a canonical two link topology we benchmark different carrier sensing and interference scenarios. We selectively disable the carrier sensing of transmitters to create the complete set of scenarios. Table 5.1 presents all possible cases. The first column captures the interference relationship between the two links assuming that C_1 is associated with AP A, and C_2 is associated with AP B. Four cases of interference are captured: (i) A interferes with C_2 and B interferes with C_1, (ii) A interferes with C_2 but not B with C_1, (iii) B interferes with C_1 but not A with C_2, and (iv) A, and B do not interfere with each others client. The second column captures the carrier sensing relationship between the two transmitters: (i) A, and B carrier sense each other, (ii) B carrier senses A, (iii) A carrier senses B, and (iv) A and B are not in carrier sensing range. Then we report the carrier sensing estimate of PIE and the associated LIR (computed as described in Section 5.3). Finally, in the fifth and sixth column we report the LIR of bandwidth tests and the measured throughput.

As shown in the table, the LIR estimates of PIE are very close to the values reported by the unicast bandwidth tests for all scenarios. PIE is also able to detect the carrier sensing relationships

Topology		PIE stats				BW stats			
INT	CS	CS		LIR		LIR		Tput(Mbps)	
		$A \rightarrow B$	$A \leftarrow B$	(A, C_1)	(B, C_2)	(A, C_1)	(B, C_2)	(A, C_1)	(B, C_2)
$A \rightarrow C_2 \wedge B \rightarrow C_1$	$A \leftrightarrow B$	0.96	0.95	1.00	1.00	0.99	0.99	2.88	2.84
	$A \rightarrow B$	0.97	0.03	0.12	0.99	0.08	0.97	0.40	5.03
	$A \leftarrow B$	0.05	0.95	0.99	0.13	0.96	0.07	4.99	0.41
	$A \updownarrow B$	0.00	0.02	0.62	0.47	0.51	0.48	2.6	2.5
$A \rightarrow C_2 \wedge B \updownarrow C_1$	$A \leftrightarrow B$	0.95	0.95	0.99	1.00	1.00	1.00	2.88	2.85
	$A \rightarrow B$	0.88	0.11	0.94	0.09	0.84	0.19	4.32	1.01
	$A \leftarrow B$	0.02	0.98	1.00	0.37	0.98	0.31	5.05	1.16
	$A \updownarrow B$	0.01	0.03	0.99	0.24	0.99	0.13	5.11	0.67
$A \updownarrow C_2 \wedge B \rightarrow C_1$	$A \leftrightarrow B$	0.95	0.95	1.00	1.00	1.00	0.99	2.82	2.88
	$A \rightarrow B$	0.94	0.05	0.17	1.00	0.11	0.98	0.58	5.10
	$A \leftarrow B$	0.19	0.80	0.06	1.00	0.09	0.84	1.01	4.36
	$A \updownarrow B$	0.04	0.02	0.17	1.00	0.08	1.00	0.45	5.17
$A \updownarrow C_2 \wedge B \updownarrow C_1$	$A \leftrightarrow B$	0.95	0.95	1.00	1.00	1.00	0.99	2.83	2.87
	$A \rightarrow B$	0.91	0.08	0.80	1.00	0.78	0.99	3.21	5.10
	$A \leftarrow B$	0.05	0.95	0.93	1.00	0.89	0.65	4.61	3.36
	$A \updownarrow B$	0.05	0.01	1.00	1.00	0.91	0.99	4.73	5.01

Table 5.1: Micro-experiments for verifying accuracy of PIE in determining conflicts. Packet size and data rate was fixed at 1400 bytes and 6M respectively. We experiment with all possible combinations of carrier sensing and interference properties for a given two transmitter receiver pair.

accurately in all cases (compare the second and third column). Note that identifying both carrier sensing and LIR values accurately can characterize client performance under any scenario. For instance, in the scenario where the interference relationship is $A \rightarrow C_2 \wedge B \rightarrow C_1$, the links can achieve similar throughputs when they are carrier sensing and sharing the channel ($A \leftrightarrow B$) or when they are not carrier sensing and there is close to 40% loss rate for the links. PIE can provide this greater visibility, as to which phenomenon is actually taking place, which can then be leveraged by interference mitigation mechanisms.

Overall accuracy in larger testbed: We repeat the experiments reported in Table 5.1 for a large

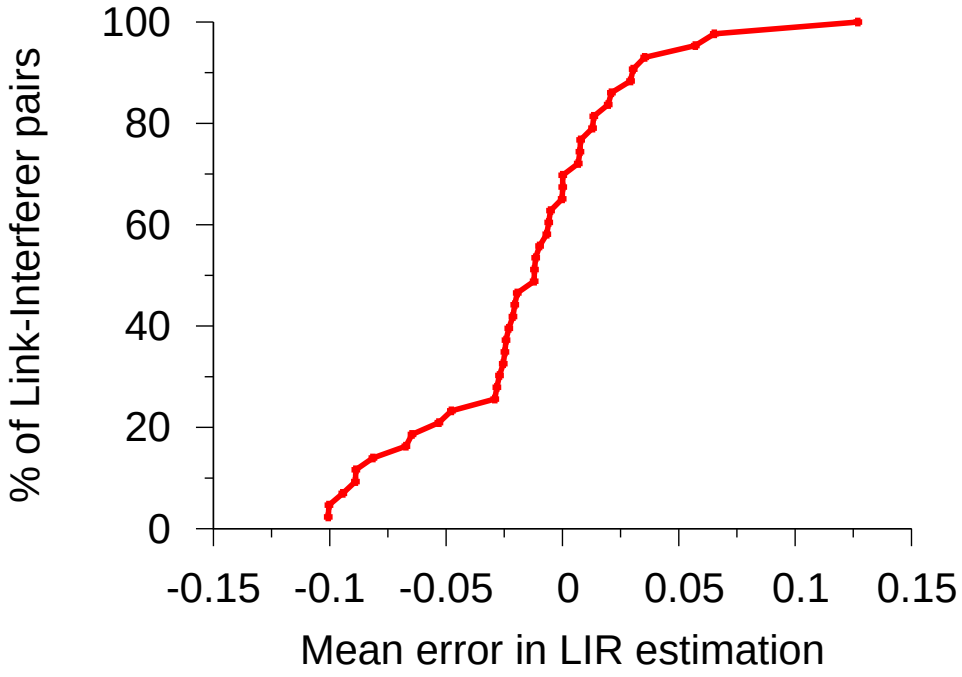


Figure 5.8: Distribution of error for PIE as compared to LIR. We note that in 95% of the interference scenarios PIE is within 0.1 of the actual LIR value.

number of link pairs in our testbed, comprising 30 nodes spread across five floors of our department building. We select links whose delivery ratio in isolation is greater then 0.9 in both directions [24]. Figure 5.8 compares the values of LIR achieved using unicast bandwidth test and PIE for 43 interference scenarios. We note that for 95% of the interference scenarios PIE is within 0.1 of the actual LIR value.

Finally, we would like to point out the inaccuracies that are introduced through approaches like broadcast bandwidth tests (BBT), that aim to collect interference information at minimal overhead. BBT will lead to the underestimation of interference when interference impacts the reception of ACKs rather than data packets. As LIR increases with decrease in interference, such underestimation of loss by BBT may lead to inflated LIR values as compared to UBT. Figure 5.9 does indeed

confirm that such cases do exist in reality and that they lead to the overestimation of LIR for a link-interferer pair.

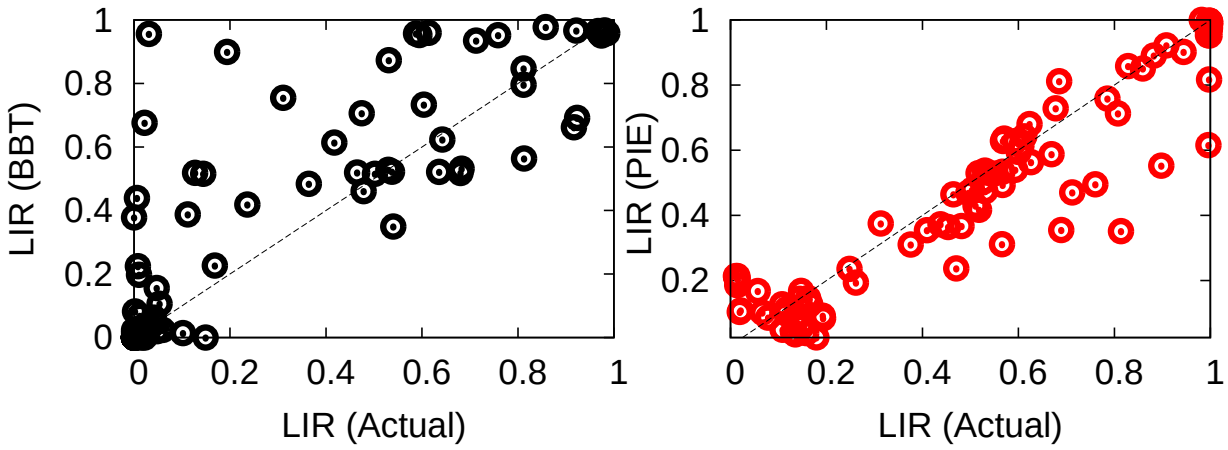


Figure 5.9: Scatter plot of delivery ratios obtained using bandwidth tests (unicast - LIR(Actual), broadcast - LIR(BBT)) and PIE on 43 link pairs. Note that LIR(BBT) may underestimate the loss rates as it does not take the ACK loss into account.

5.4.2.2 Dynamic interference settings

The previous experiments quantified PIE's accuracy as compared with the ground truth generated using unicast bandwidth tests. However, PIE is not only able to accurately capture interference under static conditions, but more importantly, under dynamic conditions that impose significant overhead to today's best known methods.

Handling client mobility: Any practical interference estimation mechanism must be able to handle client mobility, i.e. it should be able to update the conflict graph in real time to reflect the changing interference patterns that arise due to client movement. In order to evaluate PIE 's ability to handle mobile clients, we perform a micro experiment, where a mobile client is moving away from its AP towards a hidden interferer as shown in Figure 5.10. In this experiment, the client is moving at a pace of 0.25 m/s[6]. The bottom plot in the Figure shows the signal strength at the client from the AP and the interferer, while the middle and top plots show the throughput of the mobile

[6]Normal walking speed for mobile user.

client and the LIR estimate by PIE at each instant in the experiment. As shown in the Figure, the PIE's LIR estimate decreases as the client moves towards the interferer. Furthermore, it closely matches the trend shown by the instantaneous throughput during the experiment, which confirms PIE's accuracy in predicting the end user performance in dynamic wireless environments.

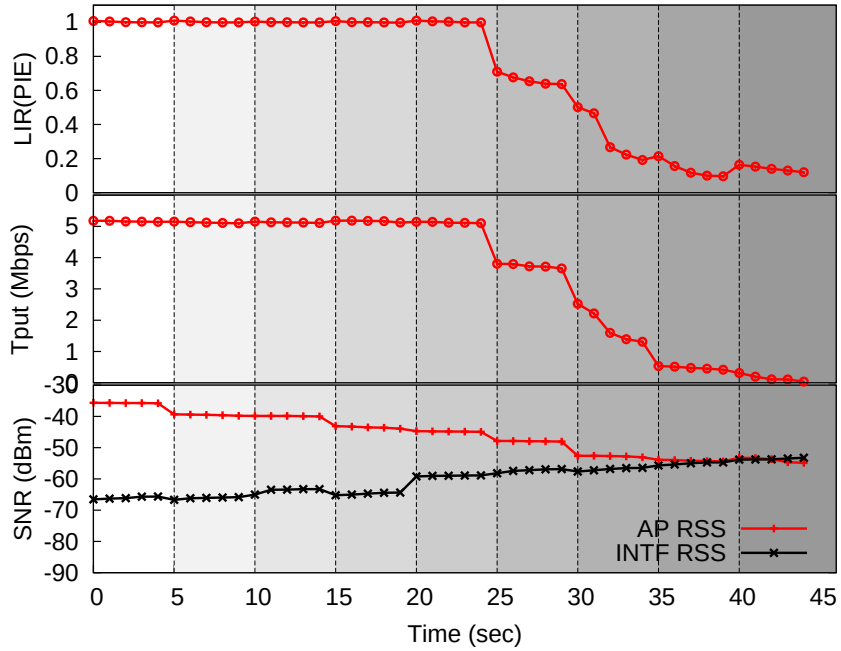


Figure 5.10: PIE 's ability to track the changing interference patterns for a mobile client. In this experiment, a mobile client is moving away from it's AP towards a hidden interferer. The bottom plot shows the signal strength at the client from the AP and the interferer. The middle plot shows the throughput achieved by the client at each instant. The top plot shows the LIR as measured by PIE.

Variable rate and packet sizes Prior research [161, 25] has shown that data rates and packet size impact the interference properties of wireless links. In order to evaluate PIE for different packet sizes and data rates, we repeat our canonical experiments with different packet sizes and data rates on multiple interferer-link pairs. To evaluate multiple data rates, we first activate a link in isolation and then activate an interferer, which forces the transmitter to adjust its data rate to minimize losses. In our experiments, APs use the popular auto-rate fallback (ARF) algorithm to adjust the data rate for their wireless transmissions. Figure 5.11 (left) shows the impact of data rate on the

delivery ratio of a link (LIR by UBT) and the estimate of LIR generated by PIE for each rate in the experiment.

Next, we fix the data rate and vary the packet size for a link under interference (right plot). As expected, LIR is worse for larger packet sizes, which are prone to more errors. We observe that the combination of data rate and packet size can result in varying interference properties and PIE is able to efficiently identify the impact of interference accurately in each such scenario (confirmed by the agreement with UBT). This also shows that using bandwidth tests or other active measurements may require performing an exponential number of tests with varying packet sizes and rates to determine the interference impact for any given traffic scenario. PIE, on the other hand, can passively determine the extent of interference for each scenario efficiently and accurately.

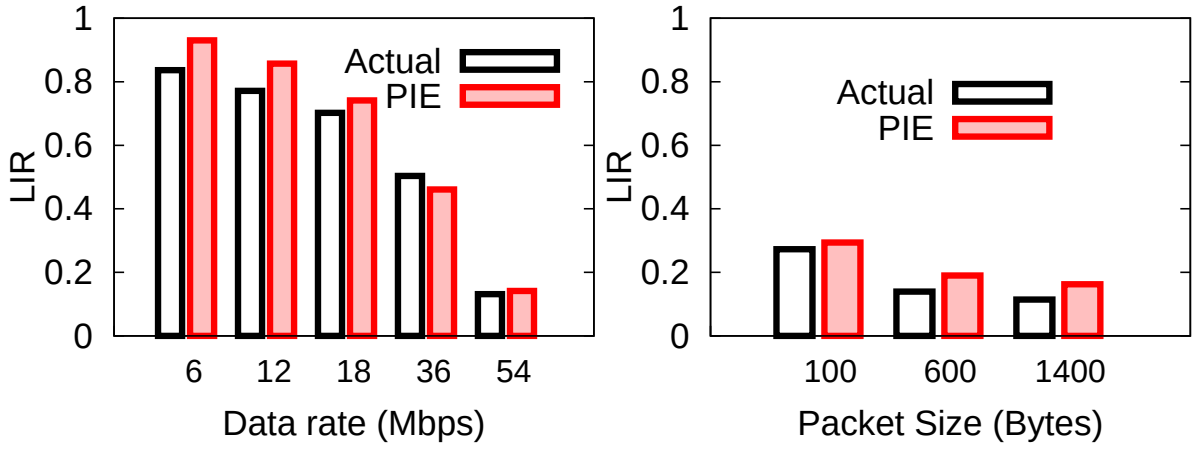


Figure 5.11: Impact of physical layer data rate and packet size on the delivery ratio of a link in a canonical hidden terminal topology. While varying data rate, packet size is fixed at 1400 bytes, and while varying packet size, data rate is fixed at 24Mbps. Note the significant drop in delivery ratio with rate while the impact of packet size is less pronounced. Confidence intervals were found to be tight and hence are omitted for clarity.

Classifying interferers accurately: PIEś fundamental operation relies on observing overlap in transmissions and correlating such events with packet loss. One could argue that PIEś accuracy is likely to be affected by scale since the probability of observing overlap in transmissions across the network increases with greater scale. Then the probability of identifying the interferer responsible

for loss becomes much harder. To answer this question we attempt to quantify the success of PIE in correctly identifying an interferer depending on the amount of time that it tends to overlap with the transmitter suffering the loss.

Setup: Consider a link (A, B) and two interferers C_1 and C_2. We compute the actual LIR of the link under C_1 and C_2 by performing individual unicast bandwidth tests, first with C_1 and then with C_2. According to the unicast tests, the LIR of the link under interference from C_1 and C_2 is 0.6 and 0.99 respectively, indicating substantial interference from C_1 and no interference from C_2. We term C_1 as the "Interfering" transmitter and C_2 as the "Non-interfering" transmitter. Our goal is to evaluate the accuracy of PIE in identifying the "Interfering" (C_1) and "Non-interfering" (C_2) transmitters, when both C_1 and C_2 are activated simultaneously. Both C_1 and C_2 follow a http traffic model, with sleep and active times being drawn from a 802.11 wireless trace [106]. We then identify the time periods (1s) in the experiment with varying overlaps between the transmission times of C_1 and C_2 and measure the LIR values for C_1 and C_2 according to PIE.

Figure 5.12 shows the LIR obtained by PIE for both the "Interfering" (C_1) and "Non-interfering" (C_2) transmitter as a function of the overlap in their wireless transmission times. As expected, when the overlap in transmission times is close to 100%, PIE is unable to distinguish between true and false interferers. When the overlap is less than 60% PIE can distinguish between the false and true interferer. In fact, notice that even for high overlaps (close to 75%), the median loss probability for false interferer is close to 0. Note that our analysis of the UCSD jigsaw traces reveals that more than 90% of the transmissions in the trace featured a transmission overlap less than 20%, thus indicating that PIE could be highly accurate in typical networks.

To further validate the previous result, we repeat the aforementioned experiments with an increasing number of active "Non-interfering" transmitters (LIR $>$ 0.9) in the system, each of them following a http traffic model of on-off traffic. We measure their impact on the accuracy of PIE in identifying the true interferer. For each set of active transmitters, Figure 5.13 shows the quartile LIR obtained for the "Interfering" transmitter C_1. Increasing the number of active transmitters has little impact on the accuracy of the LIR computed for the interfering transmitter. We also observe that the mean LIR for the "Non-Interfering" transmitters is less than 0.1 for all scenarios. This

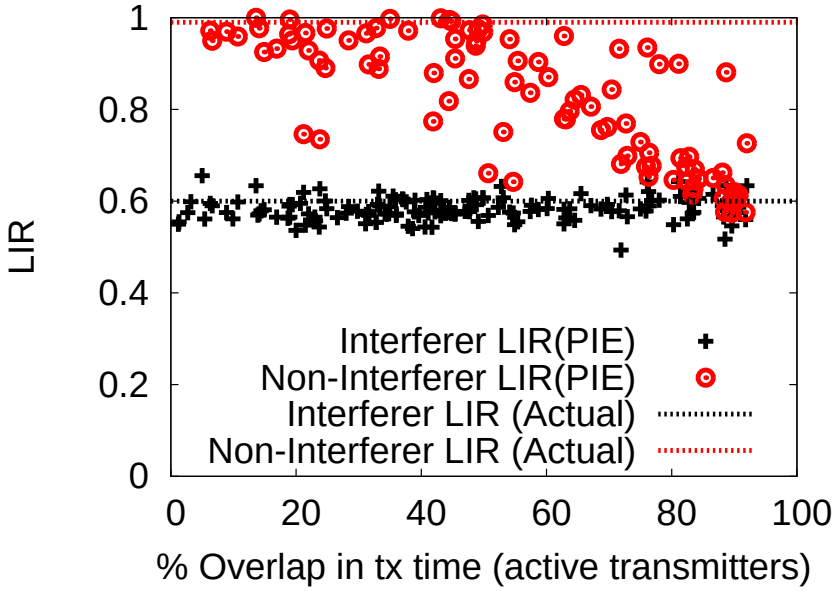


Figure 5.12: Ability of PIE to identify true interferers from a set of active transmitters. We plot the LIR measured by PIE for both the true interferer and the non-interfering transmitter as a function of the overlap in transmission times. Clearly, when the overlap in transmission times is close to 100%, PIE is unable to distinguish between true and false interferers. If the overlap fraction is less then 60%, PIE can distinguish the false and true interferers accurately.

shows that as long as there is some diversity in the transmission times, PIE can accurately identify the cause of interference.

Multiple interferer experiments: To validate the previous result with multiple interferers, we repeat the aforementioned experiments with a larger topology. In our experiments, we try to emulate the structure of our in-building WLANs by placing one testbed AP node near each production AP in the environment. We present results from a representative topology that randomly distributes client nodes into offices. The topology has 7 APs and 8 clients. Clients connect to the AP with the strongest signal strength. Each transmitter follows a http on-off model for transmitting data with the on and off times derived from the UCSD trace. We classify all interferers for which the UBT LIR is less than 0.8 ($>$ 20% loss) as strong (interfering) transmitters and rest are classified as weak (non-interfering) transmitters.

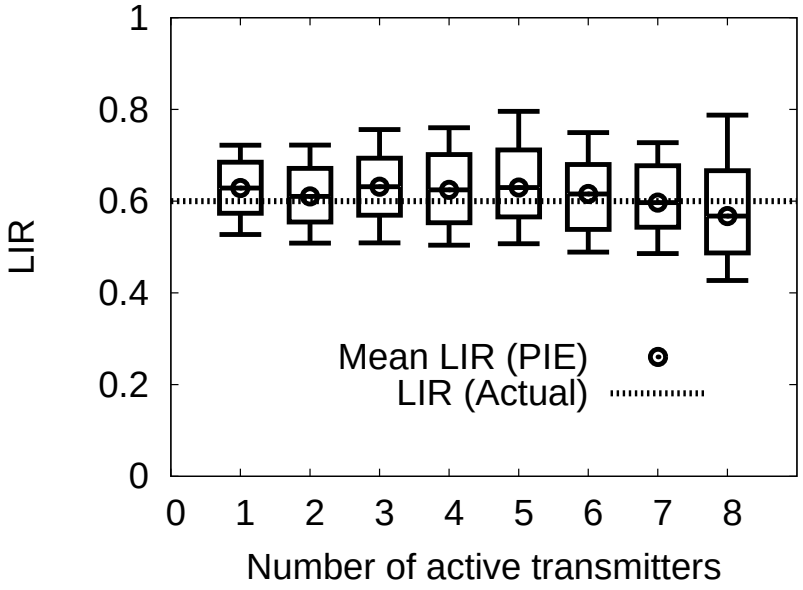
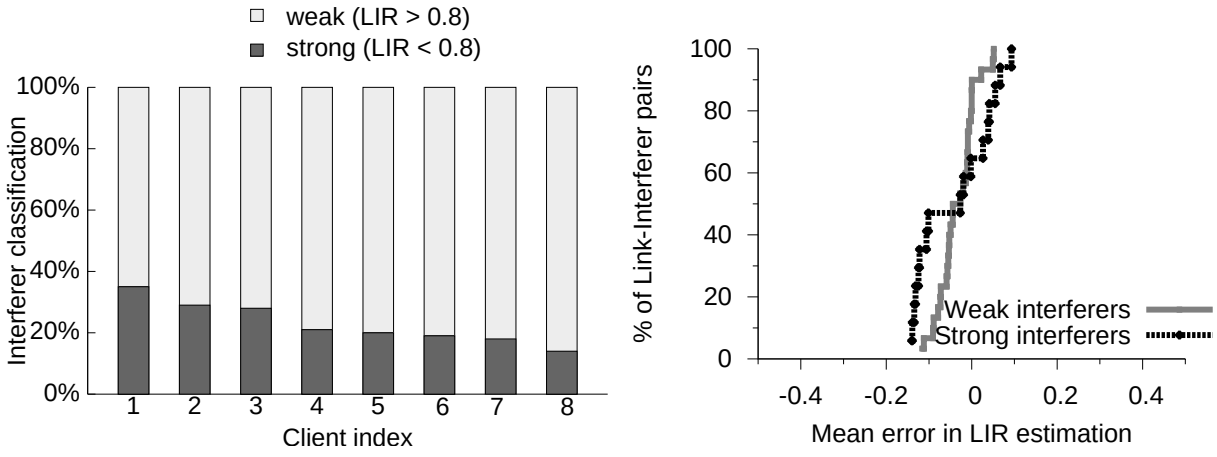


Figure 5.13: Ability of PIE to distinguish between interfering and non-interfering transmitters, as a function of the number of active transmitters. The quartile LIR remains stable and equal to the actual value.

First, we evaluate the ability of PIE to identify the strong interfering transmitters from a mix of multiple simultaneously active transmitters. We keep one strong interferer (C_1) and vary the number of non-interfering transmitters in the system and measure the value of LIR reported for all the transmitters by PIE . For each set of active transmitters, Figure 5.13 shows the quartile LIR obtained for the interfering transmitter C_1. Increasing the number of active transmitters has little impact on the accuracy of the LIR computed for the interfering transmitter. We also observed that the mean LIR for the non-interfering transmitters was less than 0.1 for all scenarios. This shows that as long as there is some diversity in the transmission times, PIE can accurately identify the cause of interference.

Next, we evaluate the overall accuracy of PIE in identifying multiple strong and weak interferers in the system. Figure 5.14 (a) shows the number of strong and weak interferer per client as determined by UBT in our topology. Figure 5.14 (b) shows the ability of PIE to identify multiple strong and weak interferers in this topology. As shown in the Figure, the LIR values estimated by PIE are within +/- 0.15 of the actual LIR determined by pairwise bandwidth tests using unicast

traffic (UBT). Summarizing, PIE is able to accurately identify the exact impact of each interferer on every client in the system even in the presence of multiple interferers. We show the overall impact of such an accurate conflict graph on application level performance for wireless clients in the system in Section 5.5.



(a) Per client interferers

(b) PIE error in multi-link conflict graph

Figure 5.14: Accuracy of PIE for a 8 client, 7 AP topology. (a) Distribution of strong (LIR $<$ 0.8) and weak (LIR $>$ 0.8) interferers for the clients in the topology. (b) CDF shows the error in PIEś estimation of LIR for a link-interferer pair as compared to pairwise bandwidth test (UBT). PIE identifies both multiple strong and weak interferers accurately (all estimates are withing +/- 0.15 of UBT LIR values). PIE is able to identify the extent of interference accurately in the presence of multiple strong and weak interferers.

5.4.3 Agility of PIE and overhead

PIE can be easily integrated in today's centrally controlled WLANs, requiring software-only modifications to the central controller. Upon the availability of the PIE software, which we intend to release through sourceforge, it took us less than an hour to deploy and start collecting real time interference information in each one of our testbeds. However, as is apparent from the design section, there are a number of knobs in PIE 's design that are likely to affect its accuracy. In this section, we study appropriate values for the polling interval, and measure PIE 's convergence time

under different traffic loads. Finally, we discuss PIE 's potential overhead, which is assessed to be minimal.

5.4.3.1 Polling interval

Any online interference estimation mechanism must identify conflicts in real time to be useful. In PIE , the controller periodically polls the APs for transmission summaries and then determines link conflicts. Higher polling periods can provide more information to the controller, thereby improving the quality of interference estimation. However, having a higher polling period also makes the system less responsive, which may be critical to dynamic interference scenarios. Here we evaluate the performance of PIE with different polling periods and determine the minimum period for which PIE can provide stable LIR values. We define a LIR value reported by PIE to be stable when the 90th and 10th percentiles of the LIR estimates are within 0.1 of the mean LIR value. Figure 5.15 demonstrates that a value of 100 ms provides a good compromise between reactivity and accuracy.

5.4.3.2 Convergence time

The time for PIE to converge to the correct measure of interference will not only depend on the polling interval but also on the amount of traffic passively observed. Figure 5.16 shows the convergence time for a canonical hidden terminal link as a function of traffic load on the link and the interferer. Lower traffic loads lead to longer convergence times because of the reduced frequency of interference events. Note, however, that LIR values would correspond to perceived client performance degradation only under relatively heavy loads, in which case PIE could capture events in 100 ms.

5.4.3.3 PIE overhead

Agility is likely to come at increased overhead. In this section we quantify PIE's communication, and computational overhead on APs and the controller.

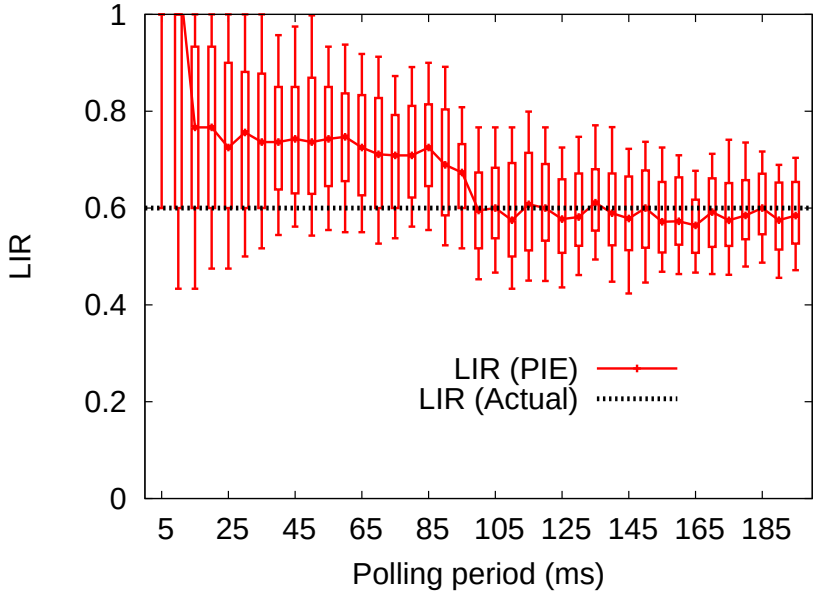


Figure 5.15: Impact of polling period on the accuracy of the interference measures produced by PIE. The LIR value stabilizes for polling periods greater then 100ms. The experiment time was adjusted to ensure same sample size for different polling periods.

Communication overhead: Each AP in PIE maintains a per packet summary that it forwards to the controller at the end of each polling period. Depending on the amount of information transmitted to the controller every polling period (100ms), one could question PIE's feasibility due to its communication overhead.

In reality this overhead is actually quite reasonable and could easily be accommodated in today's enterprise WLANs. Each AP creates a summary of packets. For each one of them, it computes the difference with the previous packet timestamp (32 bits), packet duration (32 bits), transmit rate (4 bits), retry (1 bit), packet success report (1 bit), source-destination identifier (16 bits) and signal strength (10 bits). This amounts to a total of 96 bits per packet. Using an average packet size of 600 bytes, and a medium constantly busy at 54 Mbps, the AP will have to store a summary for 1125 packets. This results in 9 KBytes sent from each AP every 100 ms, i.e. 1 Mbps, easily sustained by the AP. On the controller side, this translates to 200 Mbps of traffic, for controllers supporting 200 APs (typical product in today's market). Given that most centrally

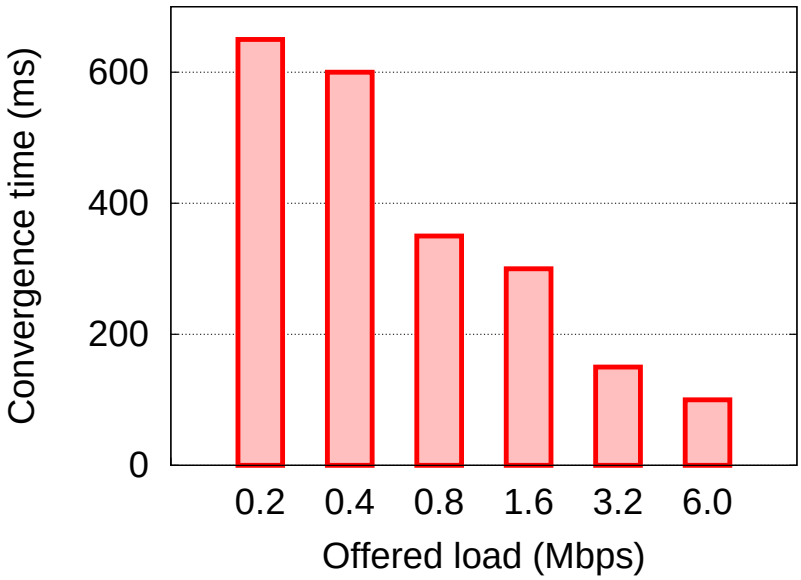


Figure 5.16: Convergence time for a canonical hidden terminal link as a function of traffic load on the link and the interferer. Confidence intervals were found to be tight and hence are omitted for clarity. Both the link and the interferer are operating on a data rate of 6Mbps. Lower traffic loads take longer to converge because the frequency of interference events is reduced.

controlled WLANs operate on Gigabit Ethernet, even under such pessimistic computations the communication overhead would not exceed 20% of the available capacity.

Computation overhead: If communication is not the fundamental overhead, how is the controller impacted by the additional processing of the packet summaries? To answer this question we describe the process at the controller. Once the controller receives the summaries from the APs, it sorts the packet summaries with respect to the reception timestamps at the APs. It then iterates over the sorted list and computes the overlaps and impact of overlap on the packet loss. It then updates the carrier sensing and conditional loss probabilities for different links in the system. Carrier sensing and interference properties are maintained using EWMA. In our implementation, such an operation takes less than $10ms$ on a single core machine generating the conflict graph for a 20 link topology. WLAN controllers tend to be much more powerful machines that could perform

these operations much faster. Also notice that the aforementioned functionality could be easily parallelized and implemented on multiple cores.

5.4.4 Convergence with real traffic patterns

Time taken for PIE to converge on the accurate estimate of link interference depends on two key factors: i) the polling period used by PIE to collect statistics from the APs, and ii) the actual amount of traffic that is captured by the APs in a given polling period. In order to understand the convergence of PIE under realistic traffic patterns, we replay a real WLAN trace on our wireless testbed and measure the time taken by PIE to converge on an accurate estimate of LIR for different link-interferer pairs in our testbed topology.

Playback of real wireless traces: Section 3.6.2 describes our methodology for replaying realistic traces on our testbed. We briefly summarize our replay mechanism here. From the public Sigcomm 2004 conference traces [139], (collected on the wired controller of the WLAN setup during SIGCOMM 2004), we extract HTTP transactions that is categorized into a series of HTTP session files. Each session file consists of a set of timestamped operations starting with a connect, followed by a series of sends and receives (called transactions), and finally a close. The HTTP session files are then replayed on our testbed using the mechanism described in [52]. During the replay, timing between HTTP sessions and transactions is enforced as per the timing in the real trace. In our experiments, each client emulated the behavior of one real client from the trace, faithfully imitating its HTTP transactions.

We partitioned the original trace into one hour time periods, and each one hour time period is classified as heavy, medium or light traffic period depending on the total fraction of time the wireless medium is busy in that particular time period [138]. Time periods where wireless medium is busy more than 50% of the time are classified as heavy traffic periods. On the other hand, time periods where medium is busy less than 20% of the time, are categorized as light traffic periods. All other periods, where medium is busy between 50% - 20% of the time, are classified as medium traffic periods. Each client in our topology randomly picks sessions from the target load period and replays the HTTP transactions as explained above.

Representative topology: In our experiments, we try to emulate the structure of our in-building WLANs by placing one testbed AP node near each production AP in the environment. We present results from a representative topology that randomly distributes client nodes into offices. The topology has 7 APs and 8 clients. Clients connect to the AP with the strongest signal strength. We perform our experiments on the 5GHz band using 802.11a to prevent interference from our in-building WLAN that operates on the 2.4GHz band using 802.11g standard.

Metrics: The goal of this experiment is to measure the convergence time of PIE under realistic traffic patterns. Convergence time of PIE is defined as the minimum time period required by PIE to arrive at an LIR estimate that is within $\pm$ 0.1 of the actual LIR value measured using saturated traffic for a link-interferer pair. As discussed before, convergence time is a function of the number of interference events that can be captured by PIE in a particular time period, which in turn depends on the exact traffic patterns of the link and the interferer. Hence, replaying realistic trace on the representative topology allows us to evaluate the quickness of PIE for practical traffic scenarios.

Results: Figures 5.17, 5.18 and 5.18 show the convergence time for each link-interferer pair when realistic traffic patterns from different load periods (heavy, medium and light) are replayed on the representative topology. Bottom plot of each figure compares the actual LIR value for a link-interferer pair computed using PIE under saturated traffic, with the LIR value that PIE converges to, under the realistic traffic patterns. As shown in the Figure 5.17, for heavy traffic scenarios, PIE converges within 540 ms for 80% link-interferer pairs. On the other hand, under medium (Figure 5.18) and light (Figure 5.19) traffic load periods, it takes 720 and 900 ms for 80% of the link-interferer pairs to converge to their accurate LIR values. Figure 5.20 shows the distribution of convergence time of PIE for realistic trace replay. As expected, PIE 's convergence is fastest for heavy traffic scenarios (median 400 ms), followed by medium (median 620 ms) and light (median 700 ms) traffic scenarios. Further, about 80% of the link-interferer pairs converge to their accurate estimates within 540, 720 and 900 ms for heavy, medium and light traffic loads receptively.

Summary: We replay realistic trace from Sigcomm 2004 on our representative topology to evaluate the convergence time for PIE under realistic traffic patterns. PIE converges within 600 ms for most links under heavy traffic scenarios, while medium and light traffic scenarios may require

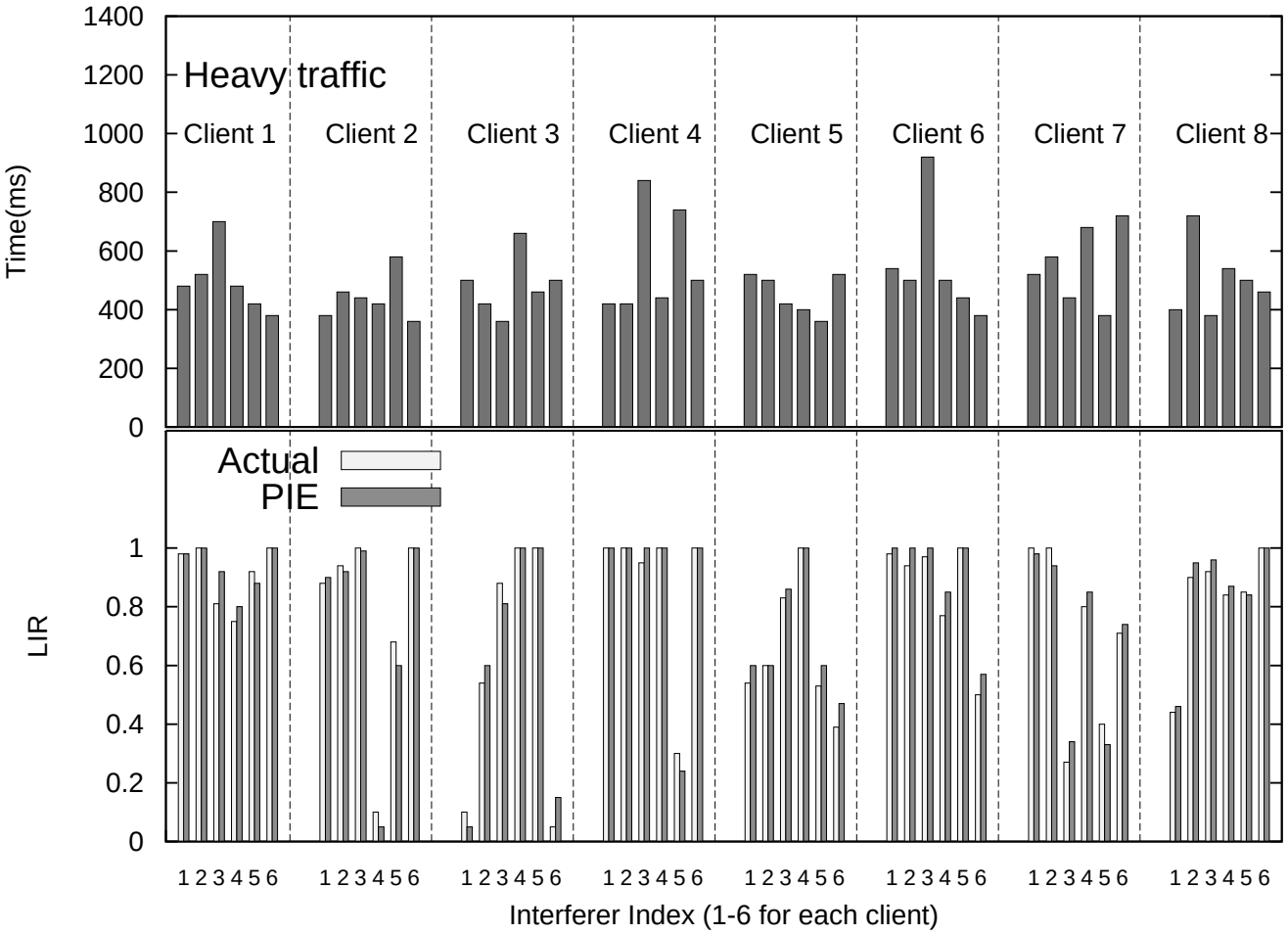


Figure 5.17: Convergence time and accuracy for PIE on a 7 AP - 8 Client topology under realistic patterns replayed from a period of heavy client activity. Top figure shows the convergence time for each link-interferer pair and the bottom figure shows its corresponding accuracy when traffic traces are replayed on our representative topology. As shown in the figure, for heavy traffic scenarios, PIE converges within 540 ms for 80% for link-interferer pairs.

upto 800 ms of monitoring period to reach at an accurate interference estimate. This indicates that in realistic scenarios, PIE should be able to generate accurate interference estimates for different traffic loads if it conservatively uses a polling period of around 900 ms to compute its interference estimate. Further, the wireless controller running PIE can also dynamically tune the polling period

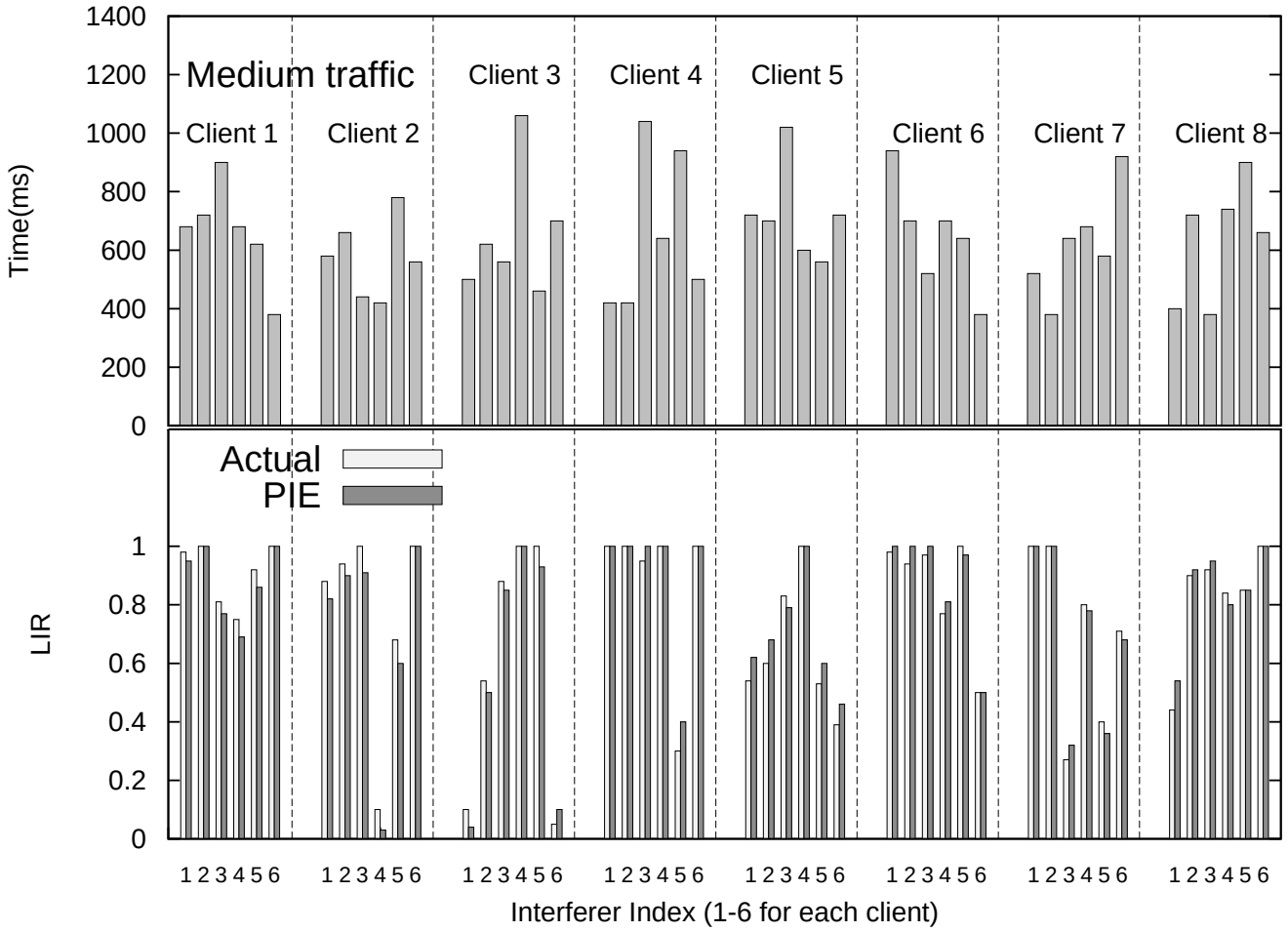


Figure 5.18: Convergence time and accuracy for PIE on a 7 AP - 8 Client topology under realistic patterns replayed from a period of medium client activity. Top figure shows the convergence time for each link-interferer pair and the bottom figure shows its corresponding accuracy when traffic traces are replayed on our representative topology. As shown in the figure, for medium traffic scenarios, PIE converges within 720 ms for 80% of the link-interferer pairs.

depending on the amount of traffic load in the system. If traffic load is above a certain threshold, a lower polling period could be used, that will allow the wireless controller to detect and react to interference quicker. On the other hand, if the traffic load is low, controller can increase the polling period to accurately capture the interference estimate.

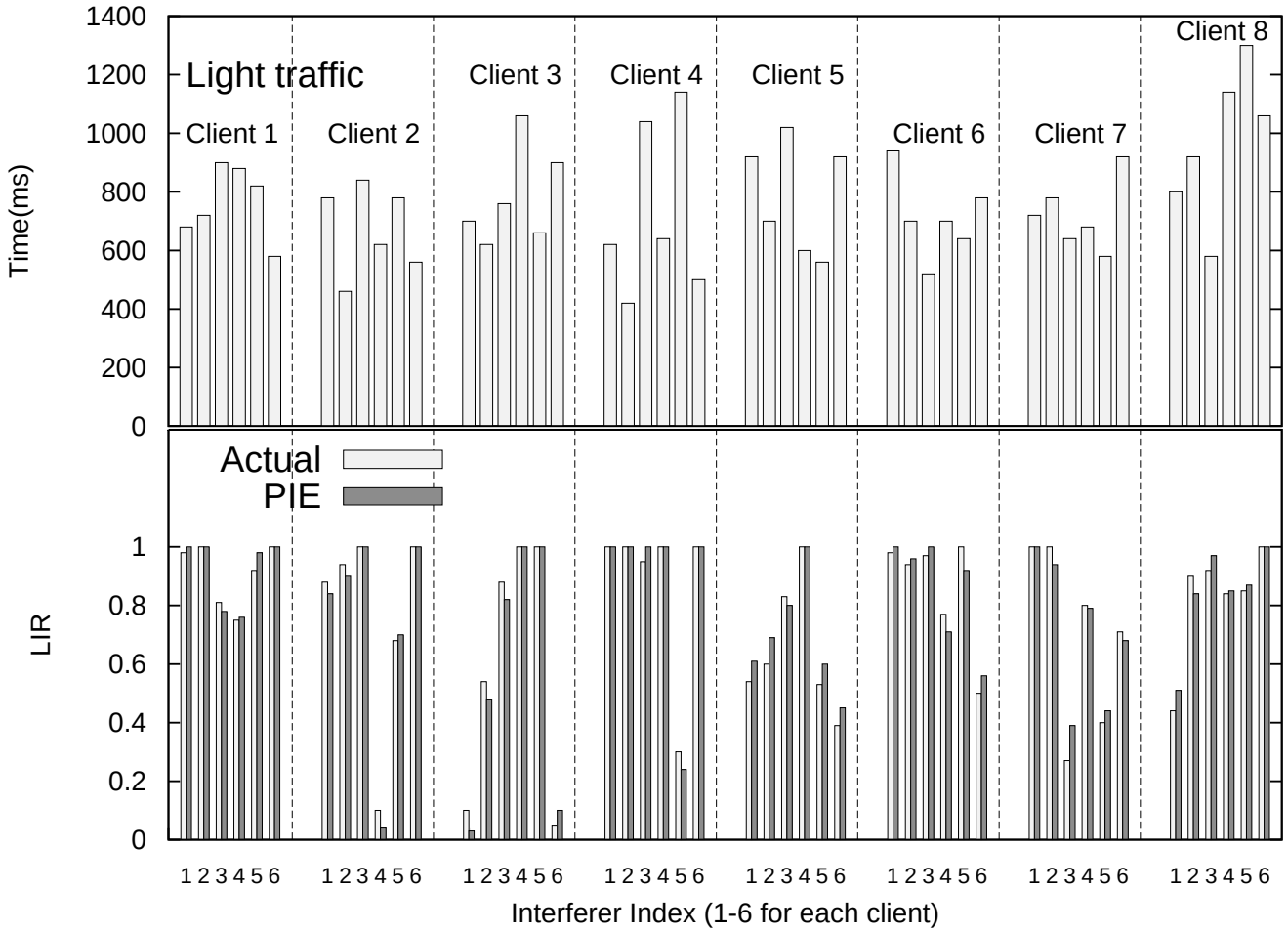


Figure 5.19: Convergence time and accuracy for PIE on a 7 AP - 8 Client topology under realistic patterns replayed from a period of light client activity. Top figure shows the convergence time for each link-interferer pair and the bottom figure shows its corresponding accuracy when traffic traces are replayed on our representative topology. As shown in the figure, for light traffic scenarios, PIE takes 900 ms for 80% link-interferer pairs to converge within $\pm$ 0.1 of their actual value.

5.5 Applications of PIE

Being able to track interference in a highly dynamic environment may be considered as an admirable academic exercise. In this section, we will prove that access to such information can

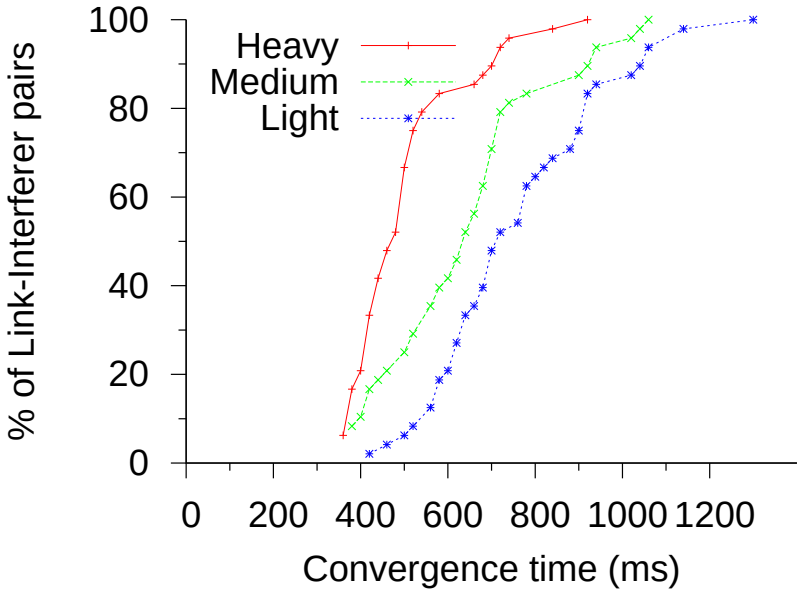


Figure 5.20: Distribution of convergence time for all link-interferer pairs under realistic traffic scenarios. Traffic scenarios are classified as heavy, medium and light depending on the total traffic load imposed by the clients. As expected, PIE 's convergence is fastest for heavy traffic scenarios (median 400 ms), followed by medium (median 620 ms) and light (median 700) traffic scenarios.

better enable a number of real time mechanisms that have been proposed for the performance optimization of wireless networks. To that end, we have integrated PIE with three such mechanisms (channel selection, dynamic packet scheduling, and power control) and tested them on two different testbeds. Our results clearly demonstrate that all these functions become a viable tool in the hands of network operators as long as we can supply reliable interference information in real time. We use the same topology as described in Section 5.4.4. In mobility experiments, each client moves along a corridor at $\sim$0.25 m/s.

5.5.1 Channel assignment

Efficiently assigning channels to access points (APs) in an enterprise WLAN can significantly affect the network performance and capacity [111, 141]. We implement a conflict aware channel assignment heuristic (Randomized Compaction), proposed in [141], that takes a conflict graph as

Conflict graph	Mechanism (Num Channels)	System Tput(Mbps)	Fairness
NA	Single (1)	9.2	0.52
NA	Random (3)	17.1	0.58
UBT	Conflict aware (3)	24.6	0.72
PIE	**Conflict aware (3)**	**24.9**	**0.71**

Table 5.2: Performance of conflict-aware channel assignment (using conflict graph generated by PIE and bandwidth tests) as compared with single channel and LCCS (least congested channel search) assignments. Under static conditions, PIE leads to similar results as UBT, offering significant improvement compared to single channel and LCCS assignments. Note that UBT being an active technique has significantly higher measurement overhead and is not practical.

input and performs channel assignment with the objective to minimize interference. We compare the performance of the conflict-aware channel assignment scheme when based on the conflict graph generated by PIE and that of unicast bandwidth tests.

Table 5.2 shows the total system throughput and Jain's fairness achieved by each channel assignment mechanism. Bandwidth tests are performed with unicast traffic at data rate and packet size of 6Mbps and 1400 bytes. Experiments are performed under static settings for a fair comparison with bandwidth tests. We consider the conflict graph generated by bandwidth tests as the true interference information. Results are averaged over 20 runs. We note that conflict aware channel assignment significantly improves system throughput over LCCS [60] (least congested channel search) and single channel assignments. Moreover, the performance of the heuristic is similar with PIE and bandwidth tests, illustrating PIE 's ability to generate high quality conflict graphs in real time.

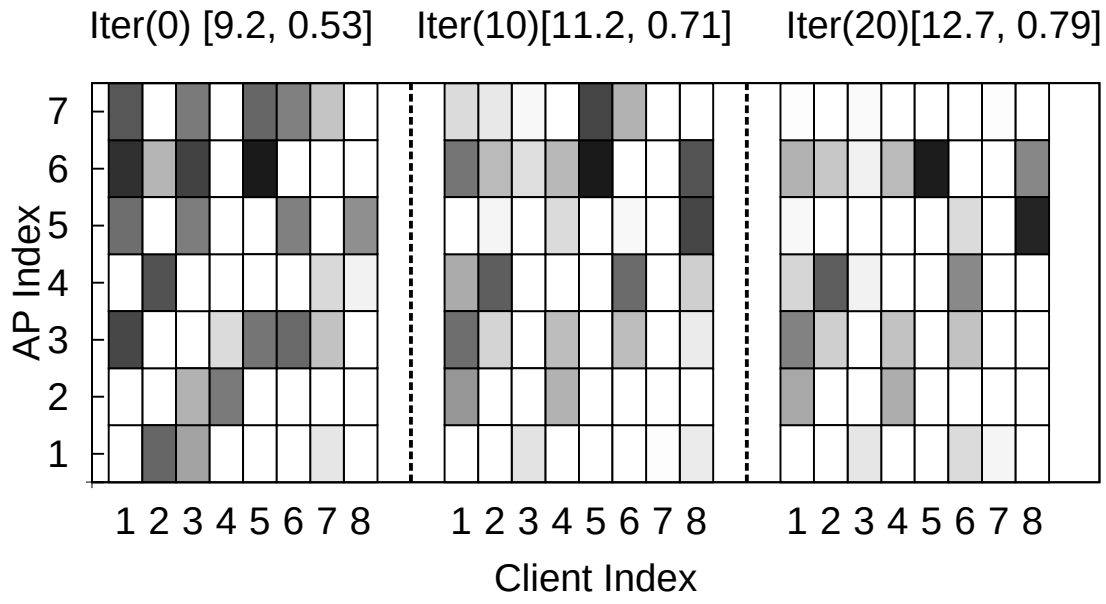


Figure 5.21: Performance of an iterative power control mechanism that uses PIE. Each matrix represents the conflict graph, with overall capacity and fairness index listed in the title. Intensity of darkness is proportional to the extent of interference. The final state corresponds to reduced interference, improved overall network capacity and fairness.

5.5.2 Transmit Power Control

We implement a simple centralized power control heuristic that uses the dynamic conflict information produced by PIE to reduce interference in the system. We measure the performance of the system through LIR_{all}, i.e. the sum of LIR values, for all link-interferer pairs in the system. Our goal is then to maximize this value by iterating over different power levels of the transmitters.

In each iteration of power control, we identify the most dominant interferer, as the AP that sources links with the minimum cumulative LIR. We reduce its transmit power and recompute the conflict graph using PIE. If the new conflict graph has lower cumulative LIR, then we discard the new power settings and reduce the power level of the next strongest interferer. In this way, we always move to a new set of power levels only if it increases the overall performance of the system. We quit when there is no improvement in the overall LIR value for n iterations.

Figure 5.21 shows the impact of such a power control mechanism. We present three matrices that capture the interference caused by each AP (row) to each client (column) in the network (the darker the cell, the stronger the interference). The title of each matrix further captures the iteration, the overall network capacity, and the fairness index. The leftmost matrix corresponds to the default power level setting, while the middle and right columns indicate the intermediate and final stages of the power level settings achieved by the aforementioned power control heuristic. We clearly see that our simple power control mechanism reduces the overall conflict in the system (matrix cells get increasingly lighter), while increasing overall network capacity and fairness. The point of this evaluation is not on the power control mechanism itself, since there are a number of solutions that could achieve such an objective more effectively (like [132]). Our focus is to demonstrate the effectiveness of PIE when used for power control.

5.5.3 Centralized scheduling

Accurate, fast and scalable conflict graph construction is critical for realizing centralized data plane mechanisms. In a recent work on centralized data path scheduling (Centaur [149]), authors relied on micro-probing [24], an online mechanism that performs micro experiments to determine link conflicts. Although micro-probing can generate an accurate conflict graph in very short time scales (4 seconds for a 10 link topology), it may still be inefficient in high mobility scenarios, especially given the need for silencing the network during the measurement of the conflict graph. We re-evaluate the performance of Centaur using the conflict graph generated by PIE and contrast it to bandwidth tests for consistency. We show that PIE improves the performance of Centaur under high mobility and varying traffic properties (variable packet sizes and data rates).

Table 5.3 shows CENTAURś performance when operating on conflict information from PIE and bandwidth tests respectively, in one static and one mobile scenario. The UBT conflict graph is generated using 6 Mbps and a fixed packet size of 1400 bytes for static client locations. Due to the overhead of repeating bandwidth tests, we use this graph for the mobility scenario too. One can clearly see that exploiting real time conflict information in scheduling is not only increasing the overall network throughput but also the fairness index across clients. More interestingly, the

Scenario	Mechanism	System Tput(Mbps)	Fairness
Static	DCF	11.2	0.64
	Centaur (UBT)	12.6	0.88
	Centaur (PIE)	**13.0**	**0.84**
Mobile	DCF	10.1	0.61
	Centaur (UBT)	10.4	0.71
	Centaur (PIE)	**12.4**	**0.95**

Table 5.3: Performance of centralized scheduling (Centaur) using PIE 's conflict graph. UBT and PIE lead to equivalent performance under static settings. The introduction of mobility confirms PIE's superiority to provide real time information. Note that UBT has very high measurement overheads compared to PIE .

inaccuracies in the conflict graph generated using bandwidth tests almost negate the benefits of centralized scheduling under mobility. We performed similar experiments with auto-rate and observe that Centaur with PIE 's conflict graph provides 32% overall system throughput gain as compared to using the conflict graph generated using bandwidth tests under static scenarios (6Mbps, 1400 bytes).

5.5.4 Wireless troubleshooting

Beyond PIE's ability to enable real time performance optimization in enterprise WLANs, its real time nature allows it to serve as a diagnosis tool that could be used proactively by a network operator to avoid performance problems. We test this property by running PIE in two production 802.11b/g WLANs (W_1 and W_2), co-located with our two testbeds.

These WLANs differ from each other in many significant ways as follows. $WLAN_1$ spans 5 floors of a building and uses 9 APs manufactured by vendor A. The network administrator was responsible for conducting RF site surveys, identifying locations to place the APs, and manually assigning the channel of operation of each AP to minimize interference. Exactly 3 APs were placed

on channels 1, 6, and 11 in $WLAN_1$ to make the level of inter-AP interference relatively low. In contrast, $WLAN_2$ occupies a single floor of a different building, uses 21 APs manufactured by a different vendor, B, and features a controller in charge of dynamic channel assignment. The number of APs on each channel, thus, varies over time. In $WLAN_2$ the vendor was responsible for conducting the RF site surveys and making AP placement decisions.

We select testbed nodes closest to the production APs to provide transmission reports to the PIE controller, sniffing the transmissions on the operational network. We use those reports to measure the carrier sense and interference relationships between different links in the production WLAN. We set the polling period to 1 second, thus capturing interference accurately even under low traffic loads. PIE reveals two performance issues:

1) Hidden terminals: Performance degradation beyond a certain level due to interference can significantly impact client performance. We set LIR_{thresh} equal to 0.7 to identify those links that suffer more than 30% reduction in their LIR under interference and classify them as hidden terminals.

2) Rate anomaly: Rate anomaly is a well documented problem [73] in wireless environments. If a transmitter of a link operating at a high data rate (say 54 Mbps), carrier senses the transmitter of another link operating at a low rate (say 6Mbps), then the link operating at higher rate will experience significant slowdown in throughput (by a factor of 1/10 in this case). We classify a given link pair as a case of rate anomaly, when the ratio of their transmission rates is less then 0.2.

Both these issues are observed in both production networks. Table 5.4 shows the extent of hidden interference and rate anomaly problem in the two WLANs. The extent of hidden interference is rather limited (8% for $WLAN_1$ and 11% for $WLAN_2$). For comparison, Jigsaw [44] also reports that 5% of their links observe an LIR of less than 0.8. While limited on average, however, we do still observe, across both WLANs, that hidden interference can lead to up to 70% LIR degradation for as many as 4% and 3% of the links in WLANs 1 and 2 respectively.

In terms of rate anomaly issues, we observe that for about 20% of carrier sensing link pairs, the transmission rates differ by more than 80%. This could be one of the reasons for sudden performance slowdown experienced by perfectly good quality links in WLANs. We note that PIE

was able to identify these scenarios in real-time (1s), and could thus allow for remediation actions, such as scheduling [149] for hidden terminals and time based fairness mechanisms [156] for links experiencing rate anomaly problems.

WLAN	HT-Links ($LIR < 0.7$)	Anomaly-Link pairs ($Ratio < 0.2$)
WLAN1	31 / 386	231 / 1087
WLAN2	53 / 464	305 / 1391

Table 5.4: Performance issues observed in three production WLANs. The extent of hidden terminal interference ranges from 8% to 11% but can be significant for a small number of links. Rate anomaly affects approximately 20% of the links in both networks.

In this chapter, we presented a detailed evaluation of a passive, real time interference estimation mechanism (PIE). We show that PIE is accurate in estimating link interference and can also adapt to changing interference patterns in real time. This enables PIE to be especially effective in realistic wireless environments, where client mobility, variable transmission rates, and bursty traffic result in ever changing interference scenarios, thereby limiting the usefulness of static bandwidth test mechanisms. Further, we show that PIE is completely passive, does not require client support and does not require any network downtime, making it attractive for use in real WLAN settings. We have integrated PIE with interference mitigation mechanisms like centralized scheduling, transmit power control and channel assignment and we show that PIE can enable these mechanisms to function efficiently and dynamically by providing an accurate conflict graph in real time. Finally we use PIE to monitor two production WLANs and show that PIE can diagnose subtle performance issues in real systems. In the next chapter, we conclude this thesis by summarizing our work in designing and implementing control and data plane mechanisms for interference mitigation in enterprise WLANs.

Chapter 6

Conclusions

Radio interference remains a core concern among WLAN users, network administrators, and operations staff alike. This dissertation examines proposals that leverage the centralized architecture of enterprise WLANs to manage and mitigate radio interference. In this chapter, we first present a summary of our work (Section 6.1), followed by a discussion on avenues for future work (Section 6.2). We then discuss the relevance of the mechanisms presented in this dissertation in view of continuously evolving wireless technologies that may impact WLAN planning and deployment (Section 6.3). In conclusion, we present some thoughts on the process underlying this research (Section 6.4).

6.1 Summary

Enterprise WLANs have made a dramatic shift towards centralized architectures in the recent past. The reasons for such a change have been ease of management and better design of various control and security functions. In this dissertation, we present the design, implementation, and evaluation of data and control plane mechanisms geared towards managing interference in enterprise WLANs. We leverage the centralized architecture of enterprise WLANs in designing efficient interference mitigation mechanisms.

6.1.1 Centralized Data Plane

While it is natural for control plane mechanisms in enterprise WLANs to be centralized in nature, it is not immediately obvious whether data plane mechanisms, i.e., channel contention and

access for competing transmitters, should also be centralized. However, in order to facilitate the convergence of different services like voice, data, and video in a single WLAN, there needs to be control on the data plane of the wireless medium to enforce multiple service levels, bandwidth contracts, traffic shaping, and spectrum utilization.

We take a fresh, implementation and deployment-oriented view in understanding data path choices in enterprise WLANs. We propose CENTAUR – a hybrid data path for enterprise WLANs, that combines the simplicity and ease of DCF with a limited amount of centralized scheduling from a unique vantage point. We perform extensive measurements to characterize the impact of various design choices, like scheduling granularity on the performance of a centralized scheduler, and identify regions where such a centralized scheduler can provide the best gains.

Our detailed evaluation with scheduling prototypes deployed on two different wireless testbeds indicates that DCF is quite robust in many scenarios, but centralization can play a unique role in 1) mitigating hidden terminals — scenarios which may occur infrequently, but become pain points when they do and 2) exploiting exposed terminals – scenarios which occur more frequently, and limit the potential of successful concurrent transmissions. Our mechanisms do not require client cooperation and can support legacy 802.11 clients.

6.1.2 Centralized Control Plane

The wireless controller in a centralized WLAN has a unique vantage point, offering a global view of the entire WLAN. This can be used to identify different wireless contention domains and could also be leveraged to implement smart centralized mechanisms that configure the power levels and wireless channels of the APs to manage contention in enterprise WLAN settings. For example, the centralized controller can facilitate the desired service (bit-rate) to any particular client by adjusting the power levels for its corresponding AP. In order to provide such multiple service levels in enterprise WLANs, we investigated the feasibility of using fine grained power control in enterprise WLANs. However, we observe that multipath, fading, shadowing, and external interference from wireless devices, make the implementation of fine grained power control challenging in practical settings.

Our work suggests that a few, 3-5, discrete power level choices are sufficient to implement any robust power control mechanism in typical indoor WLAN environments [148]. Through our work, we also design an empirical model based transmit power control mechanism, Model-TPC, that determines these appropriate number and choices of power levels that are adequate in any setting and uses only those power levels for faster convergence on the right transmit power settings. Further, through collaborative measurements made by different clients over time, the centralized controller can create a location-dependent model for power control, which can be downloaded to clients during association. This unified model can enable the centralized controller to facilitate per client power control, which can be useful for satisfying diverse bandwidth requirements for individual clients.

An accurate, fast, and scalable mechanism for detecting potentially interfering links is critical for realizing efficient data and control plane mechanisms. Such a tool for interference estimation can enable WLAN managers to improve the network performance by dynamically adjusting operating parameters like channel of operation and transmit power of access points, but also diagnose and proactively fix problems. Prior work on interference estimation employs active probing techniques and suffers from three main problems: a) it incurs moderate to significant measurement overhead and cannot be employed to continuously obtain interference information as it evolves over time, b) it offers limited visibility into the root cause of interference, and c) it often requires specific client support.

Motivated by these observations, we design and implement a Passive Interference Estimator (PIE) that can dynamically generate fine grained interference estimates across an entire WLAN. The most attractive feature of PIE is that it imposes no measurement traffic, and yet provides an accurate estimate of WLAN interference as it changes with client mobility, dynamic traffic loads, and varying channel conditions. Our experiments conducted on on two different testbeds show that PIE is able to not only provide high accuracy but also operate beyond the limitations of prior tools, providing a true solution to performance diagnosis and real time WLAN optimization, as manifested through its use in multiple WLAN optimization applications, namely channel assignment, transmit power control, and data scheduling.

6.2 Future Work

This dissertation lays the groundwork for an exciting set of future research problems. We describe some of the potential problems for future research in this area.

6.2.1 Centralized data plane

Our existing work in the centralized data plane can be extended in various ways, as described below.

6.2.1.1 Extending CENTAUR

Through our efforts in implementing a centralized data plane for enterprise WLANs, we learned quite a few valuable lessons and insights that can enhance the performance of a centralized scheduling framework.

Customized hardware platform: The controller is a key component of the entire centralized scheduling architecture and needs to operate at line speeds. A desktop PC architecture is, clearly, not optimal for such a task, as packets arriving from the edge router have to undergo inefficiencies of desktop-style packet processing. A more, router-style design of the controller (with fast processing on the line cards, and only complex tasks in the central processor) will be critical to meet our goals of scheduling at line speeds, especially when this hardware needs to scale up to a few hundred APs deployed in larger enterprises. Note that the major WLAN vendors like Cisco, Aruba, and Meru, already have such high performance controllers in production. Integrating CENTAUR with such a powerful controller can minimize the impact of scheduling overheads and enhance the benefits of our scheduling approach.

Uplink scheduling: The current implementation of CENTAUR schedules only downlink traffic. While downlink traffic dominates most enterprise WLANs [162], CENTAUR can still be extended to support the scheduling of uplink traffic. One way to implement uplink scheduling is to divide transmission slots into uplink and downlink slots, depending on the ratio of uplink and

downlink slots. Then the wireless controller can potentially schedule clients in uplink slots, depending on the conflict between competing clients. However, supporting uplink scheduling will require some client changes to determine the uplink load on each client and also to coerce the clients to transmit only in their scheduled uplink slots. While such uplink scheduling can be supported under the generic CENTAUR framework, a thorough implementation and evaluation of such an approach is required.

Scheduling algorithms: CENTAUR schedules packets in the order of their arrival. Better scheduling algorithms can be implemented that buffer incoming packets and schedule them in an order that maximizes the overall system throughput. While such an optimization can improve scheduling gains, it will likely increase the scheduling complexity and processing time for each packet, which may be undesirable for the scalability of the system. Exploring the tradeoff between scheduling complexity and potential gains for any given WLAN setting can be an exciting avenue for future research in data plane centralization.

6.2.1.2 Implementing centralized policy

CENTAUR currently schedules traffic in a best effort fashion and does not implement any prioritization of different flows. However, it is a natural next step to augment the scheduler to provision for quality of service. This will involve two steps: 1) determining low level traffic policies from high level policies defined by the system administrator and 2) taking traffic priority into account while scheduling packets at the wireless controller.

Prior work in policy design has primarily been in the areas of 1) access/security control and 2) configuration diagnosis and management. This work can explore a more active approach towards policy-driven performance management. Defining performance policies through low-level constructs of traffic components can be fairly tedious and administrators will benefit if the policies can be expressed in a high-level language. Such a design will, however, require an automated mechanism to convert such high-level policies into traffic aggregates by inferring behavioral properties of the traffic. Work by Karagiannis et al. [86] illustrates an approach to behavioral traffic classification on the Internet and can be a good starting point for one component of this work. Recent work

in enforcing policy at different layers of the network stack [80, 42], can also provide good insights into the design and implementation of a centralized policy manager that is tailored for enterprise WLANs.

6.2.2 Centralized control plane

Control plane mechanisms have been studied in detail and are routinely implemented in production wireless controllers. We describe how our work in transmit power control and interference detection can be extended to facilitate a new set of sophisticated control plane mechanisms.

6.2.2.1 Extending Model-TPC

Model-TPC can be used as a plug-in to previously proposed power control mechanisms, to make them implementable in real settings. The current work in determining the set of feasible power levels in an environment can be further fine-tuned in the following ways to provide better results under diverse settings.

Better statistical tools: We have used NKLD as a statistical tool to measure the distance between two RSSI distributions. Although it works well for our environments and is easy to compute in a real-time fashion, there are other statistical tools, like moment based estimators, that capture the spread of the two distributions better and may be more effective in distinguishing between two probability distributions. Comparing the performance of NKLD with moment based estimators is an avenue of future work for us.

Cell breathing and other applications: We evaluate Model-TPC by integrating it with a joint power-rate adaptation mechanism. The flexibility in estimating and building the empirical model allows for its applicability to a wide range of power control algorithms and other topology control algorithms, and might find interest outside the scope of 802.11 networks. For example, Model-TPC can be used to practically implement load balancing mechanisms like cell-breathing [67] in indoor environments. Cell-breathing requires APs to adapt their transmit powers to force clients to associate with suitable lightly loaded APs, that can keep the load balanced amongst different APs in the system. Such a mechanism can significantly benefit from the per-client feasible power levels

that can estimated by Model-TPC and then used by the APs to implement cell-breathing efficiently, especially in indoor settings.

6.2.2.2 Extending PIE

Our work in implementing and evaluating PIE has provided us with valuable insights into the functioning of a passive interference estimation mechanism. Below we outline some key extensions to PIE that can further enhance its performance under a wider range of deployment scenarios.

Uplink conflicts: In PIE, we use the APs to sniff packets in the wireless medium. This allows us to avoid the additional overhead associated with the deployment and management of extra sniffers in the enterprise building. We note that as a result of limiting sniffing at the APs, coverage of client traffic can be lower as compared to a dense sniffer deployment (as in [44]).

Hence PIE can miss some uplink conflicts as it is not able to capture the traffic from some clients efficiently. This can be resolved in two ways: 1) running sniffing software at the clients and periodically reporting the statistics to the controller and 2) inferring missing packets as outlined in [106]. While the first solution would require client changes, it might be feasible in some enterprise settings in which clients are already mandated to install some enterprise software (e.g. firewalls) to function properly. Extending and evaluating PIE for uplink conflicts can be an interesting area for future research.

Scalability: As discussed in Chapter 5, the worst cast communication overhead of PIE for a 802.11g AP is $\sim$ 1 Mbps on the wired backplane. Although this overhead is manageable for a 802.11g AP, it can significantly increase for a 802.11n AP, where data rates can be relatively higher ($\sim$ 600 Mbps). Under such high data rate scenarios, PIE can exert significant computation and communication overhead at a single wireless controller if it supports a large scale wireless network ($\sim$ 200 APs). However, we note that the functionality of PIE could be easily parallelized and implemented on multiple cores. Additionally, the entire wireless network can be manually partitioned into obvious non-interfering regions (like different wings of a building that are well isolated), where each region of the network is under the control of a separate wireless controller.

We believe implementing and evaluating such partitioning mechanisms to facilitate scalability of PIE is an important future step in the research in passive interference estimation.

Cross layer diagnosis: While PIE focuses only on MAC level interference (packet collisions, asymmetric carrier sensing), prior research in measuring enterprise WLANs [43] has shown that client performance can also be significantly impacted by other higher layer mechanisms like TCP contention window and standard incompatibilities between the AP and client wireless drivers. Extending PIE to perform a cross layer diagnosis of poor client performance can have much more practical applicability than PIE in its current form. We have already augmented PIE to record some high level information, like association and disassociation requests. We hope that PIE will provide a natural platform for extending the reach of real-time fault diagnosis in enterprise WLANs.

6.3 Relevance to future trends in wireless networks

Wireless networks are continuously evolving with the availability of new frequency bands for unlicensed use (e.g. 60 GHz band [150] and 700 MHz whitespaces [29]) and with the advent of smarter physical layer transmission and decoding schemes (e.g. Multiple Input Multiple Output (MIMO [124]). Such advances in wireless technology can improve the end client performance by reducing the impact of interference in practical wireless deployments. We briefly discuss the relevance of the mechanisms proposed in this dissertation in view of such key trends in wireless networks.

High throughput wireless networks: Enterprise WLANs are rapidly upgrading to the high throughput wireless standard, 802.11n, that should allow wireless links in the system to operate at speeds of up to 600 Mbps. Such high speed wireless links can impose severe bandwidth and processing constraints at the wireless controller, through which all the traffic is typically routed in a centralized WLAN architecture. There are two current approaches to mitigate such resource constraints at the wireless controller: 1) upgrade the wireless controller and the wired backplane to handle high traffic volumes (upto 10 Gbps links may be required for supporting 802.11n APs) or 2) partition the APs in the system into different groups on the basis of their physical location in the building (e.g.

APs in different wings or floors of the enterprise building can be grouped together), and use a dedicated wireless controller for managing each such group of APs. In the second approach, any one wireless controller in the WLAN would only need to handle the traffic for a subset of APs, thereby avoiding the severe bandwidth and processing power constraints faced by a single controller that manages the entire WLAN. The second approach can also scale better as wireless speeds increase and has already been adopted by many centralized WLAN vendors (e.g. Triple Distribution System from Meru [9], User-centric Architecture from Aruba [4]) for handling 802.11n clients. Finally, under both the aforementioned-mentioned approaches, the wireless controller (or multiple wireless controllers) in the enterprise WLAN should be able to employ mechanisms like CENTAUR and PIE to efficiently manage contention for the set of APs under its control.

Emerging physical layer mechanisms: The advent of smarter physical layer mechanisms (e.g. MIMO [124], PPR [77], SIC [68], ZigZag [61]), also promises to improve client performance by using sophisticated techniques, like path diversity and robust coding, for efficiently transmitting and decoding data under interference. Such physical layer mechanisms can typically be implemented as hardware and/or software updates to the wireless APs in the system. On the other hand, our mechanisms are primarily implemented at the centralized wireless controller with feedback from the APs and should provide complementary gains on top of those provided by smart physical layer schemes.

Availability of new spectrum bands: The newly-available frequency bands (60 GHz [150], 700 MHz [29]) can provide more orthogonal wireless channels as compared to 2.4 GHz or 5 GHz band, that can potentially be used to reduce the overall interference by an intelligent channel assignment. But at the same time, there is an overwhelming trend in the industry to 1) increase AP density to improve overall network capacity (see the Aruba white paper [162]) and 2) use single channel deployments for minimizing handoff latency for emerging VoIP and media streaming applications (see Meru Virtual Cell [164] and Extricom [8] solutions). Further, any future WLAN deployment will still need to support clients that operate in the popular 2.4 GHz frequency band, using only three orthogonal channels. Such factors imply that co-channel interference is still likely to be an

important concern in future WLANs, and mechanisms, like CENTAUR and PIE, that estimate and mitigate such co-channel interference should still be relevant for future WLAN deployments.

6.4 Concluding remarks

This dissertation focuses on leveraging the centralized architecture of enterprise WLANs to design practical interference estimation and mitigation mechanisms that yield tangible performance gains under realistic scenarios. Overall, the contributions of this dissertation are multifold. First, we collect detailed measurements in real wireless deployments to characterize and validate the key problems addressed in this dissertation. We show that interference can negatively impact the performance of clients in production WLANs, although the exact impact of interference varies dynamically with time and depends on the client's location and traffic patterns of other competing wireless transmitters. Second, we have taken a careful implementation and deployment-driven approach to solve the key challenges in enterprise WLANs. This meant spending substantial time in both designing elegant mechanisms and overcoming engineering challenges to implement such mechanisms on enterprise-scale wireless testbeds. Moreover, our mechanisms adhere to practical design constraints like support for legacy clients, which makes them especially attractive for use in current deployments. Finally, we have evaluated our schemes using large scale experiments with realistic traffic on multiple wireless testbeds and show that they provide consistent gains under diverse topology and traffic patterns. Such large scale experimentation also provided us with a crucial feedback loop in identifying the key performance bottlenecks in our initial solutions and have enabled us to shape our final solutions for practicality and performance. Ultimately, the overall purpose of this dissertation has been not only to explore key research problems that are crucial for performance of wireless networks, but also to carefully design mechanisms that solve such problems and could be integrated into current and future enterprise WLANs.

APPENDIX
Measurement study of interference in an enterprise WLAN

The key goal of this dissertation is to manage interference in enterprise WLANs by leveraging the centralized architecture of such deployments. We outline the design and implementation of interference mitigation (CENTAUR - Chapter 3, Model-TPC - Chapter 4) and interference estimation (PIE - Chapter 5) mechanisms, that can efficiently manage contention in enterprise settings. The design and performance of such mechanisms is impacted by the exact extent (e.g. fraction of wireless clients suffering from hidden and terminals) and type (e.g. uplink vs. downlink or symmetric vs. asymmetric) of interference. For example, if downlink hidden and exposed terminals are the dominant cause of problems for wireless clients, then we can avoid client modifications for handling uplink traffic and focus on scheduling just downlink traffic at the wireless controller (as done in CENTAUR). Likewise, it is important to understand the extent of symmetric or asymmetric exposed terminals, that may determine the exact strategy of solving such problems [1].

We perform systematic active experiments using testbed clients and production APs to uncover the extent of interference in two functional WLANs (Section 3.1). In this appendix, we extend our interference analysis to production clients that use productions APs in a functional enterprise WLAN setting. We perform detailed analysis on publicly available wireless traces from a production WLAN, deployed in the Computer Sciences building of University of California, San Diego (UCSD). It comprises of 45 APs spread across four floors. Details of the monitored WLAN could be found in [44].

We present detailed results regarding the occurrence of hidden and exposed terminals, extent of symmetric and asymmetric carrier sensing by wireless transmitters and the extent of rate anomaly

[1] CENTAUR uses fixed backoffs with batching to solve the dominant symmetric exposed terminal problems as determined by active experiments on our wireless testbed.

problem in the production WLAN. We perform our analysis on the detailed client activity traces provided by the Jigsaw [44] monitoring infrastructure that uses 180 wireless sniffers to passively capture client traffic in the UCSD WLAN.

A.1 Study goals

In order to understand the interference patterns in a real production WLAN, we analyze sender and receiver side interference in the UCSD WLAN trace. As described in Section 5.2, we can identify sender and receiver side conflicts by analyzing the timestamps of packet transmissions and loss statistics for different links. This study also allows us to test the accuracy of our mechanisms by comparing some of our results with those presented in [44]. Specifically, our goal is to understand the following key areas regarding interference in a production WLAN.

- *Carrier sensing properties:* What are the carrier sensing properties of wireless transmitters in a real WLAN ? What percentage of transmitters perform symmetric or asymmetric carrier sensing ? What is the extent of symmetric (or asymmetric) carrier sensing in uplink/downlink direction ?

- *Rate anomaly:* Does rate anomaly exist in real WLANs ? If so, what is the exact extent of rate anomaly in uplink/downlink direction ?

- *Traffic load:* How does traffic load vary across time and across different APs and clients ? Is it similar across different channels ? What fraction of traffic is uplink/downlink in nature ?

- *Hidden terminals:* What is the extent of hidden terminal problem in the WLAN ? Is the problem more prominent in the uplink or downlink direction ? How many interferers are typically impact a given AP-Client pair.

- *Exposed terminals:* What is the extent of exposed terminal problem ? Can we actually detect exposed terminals in the presence of carrier sensing ?

While it is difficult to generalize observations based on one single trace, we believe that this study does allow us insight into the actual problems caused by interference in a production WLAN

and also provides us with an opportunity to compare some of our results with [44], thereby providing an accuracy check for our interference estimation mechanisms. Finally, uncovering interference problems in a real trace also provides much better motivation to design efficient interference estimation and mitigation mechanisms, a key goal of this dissertation. Next we present our analysis of the Jigsaw trace for interference patterns. As outlined in Chapter 5, we broadly categorize interference in wireless networks into sender and receiver side interference. We begin our analysis with sender side interference.

A.2 Sender side interference

As described in Section 5.2, we determine carrier sensing relationships on the basis of the order of timestamp overlaps observed by the monitors. For a given pair of wireless transmitters, we compute overlap probability as the fraction of competing transmissions from the two transmitters that simultaneously occupy the wireless medium. The no-overlap probability is defined as 1 - overlap probability. Note that if the overlap probability for a pair of transmitters is high, it indicates that the transmitters do not carrier sense each other, resulting in overlapping transmissions most of the times when they compete for wireless channel. On the other hand, if the overlap probability is very low, it may indicate that the transmitters are carrier sensing each other, which serializes and prevents any overlap in their transmissions. Note a very low fraction of wireless transmissions from carrier sensing transmitters may still overlap if both the transmitters choose the same backoff period and access the channel in the same slot [106].

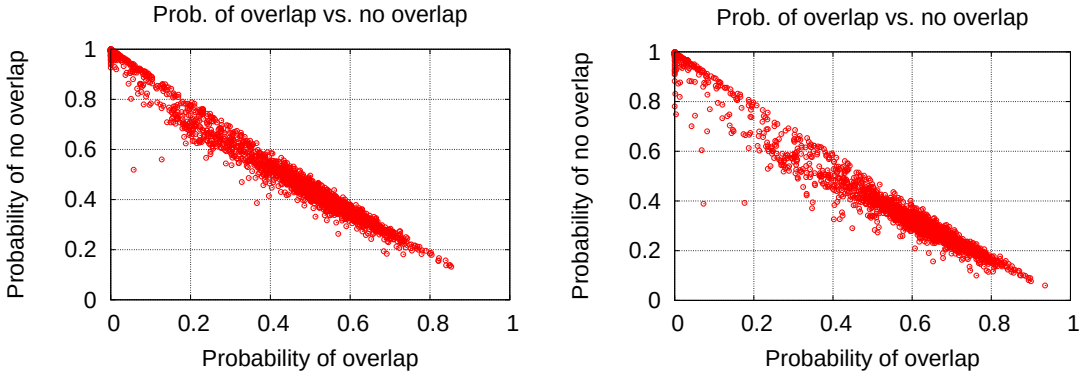
A.2.1 Carrier sense relationships

Figure A.1 (a) and (b) show the scatter of plot of overlap and no overlap probability of packets of a given transmitter pair, whose starting timestamps are between a value of $320\mu sec$ and $160\mu sec$ respectively [2]. As shown in the Figure A.1, the points lying close to (0,1) belong to the transmitter pairs whose packets never overlap even if they are in contention. On the other hand, those pairs

[2]It was shown in [106] that the start times for competing transmissions in a WLAN typically differ by a duration of average contention window in 802.11. Since this is a heuristic, we analyze the trace with two such δ_t, $320\mu sec$ and $160\mu sec$. Testing with two values also allows us to test the sensitivity of the heuristic to the choice of δ_t.

close to (1,0) see a high number of packet overlaps and hence are not mutually carrier sensing. The points in between belong to the transmitter pairs which fall in the category of asymmetric carrier sensing. Note that depending on the exact arrival timings of the packets, there may not be an overlap even though the transmitters are not carrier sensing. But we believe that if we observe a sufficient number of packets, the impact of exact packet arrival times would be negligible. In order to validate this assumption, we perform our analysis with a smaller contention window of $160\mu sec$. Although the exact number of transmitter pairs that qualify as contending pairs decreases, but we observe the same pattern in the scatter plot.

Figure A.2 shows the distribution of overlap probability for all transmitter pairs for which there were at least 100 packets in contention. Low overlap probability indicates mutual carrier sensing. Note that very few downlink transmitter pairs (APs) actually have overlap probability less than 0.2, indicating the APs usually are not carrier sensing each other. This can be attributed to careful planning while deploying APs which attempts to maximize coverage by minimizing coverage overlaps between APs. Note that pairs whose left and right overlap probability differ by less than 0.3, are likely not carrier sensing each other resulting in overlaps in both directions. While pairs whose left and right overlap probability differ by greater than 0.3 are likely experiencing asymmetric CS. Note that about 80% downlink transmitter pairs lie in mutual no CS range, while the distribution of uplink and mixed transmitter pairs is more uniform. Table A.1 shows the fraction of downlink, uplink and mixed links that do no carrier sense or perform one way carrier sense in the USCD WLAN trace. The table is derived after applying a 0.3 threshold on the distribution presented in Figure A.2(b). One interesting thing to note is that a very high percentage (85%) of overlapping downlink transmitter pairs (where both transmitteres are APs) belong to the no carrier sense range as compared to only (30%) of purely uplink transmitter pairs lying in the no carrier sensing range. This can be attributed to the careful planning that is undertaken while deploying the APs, while clients are located in a random fashion giving rise to many scenarios where asymmetric carrier sensing is possible.



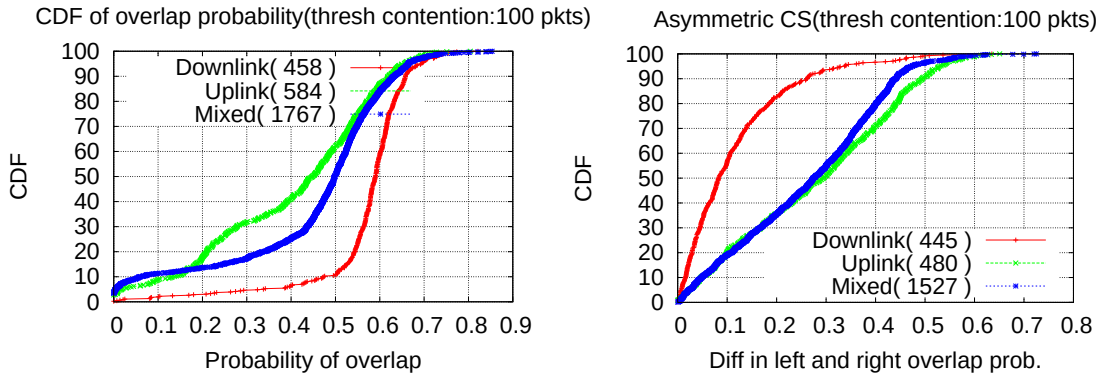
(a) Contention window 320 μsec

(b) Contention window 160 μsec

Figure A.1: Carrier sense properties for Jigsaw trace. Scatter plot of overlap and no overlap probability for all transmitter pairs for which we observe at least 100 packets in contention. Two packets whose starting timestamps differ by less than the contention period parameter are assumed to be likely in contention. Contention period is assumed to be 320 μsec in (a) and 160μsec in (b). For a given pair of wireless transmitters, we compute overlap probability as the fraction of competing transmissions from the two transmitters that simultaneously occupy the wireless medium. The no-overlap probability is defined as 1 - overlap probability. Note that if the overlap probability for a pair of transmitters is high (close to (1,0)), it indicates that the transmitters do not carrier senses each other, resulting in overlapping transmissions most of the times when they compete for wireless channel. On the other hand, if the overlap probability is very low (close to (0,1)), it indicates that the transmitters are carrier sensing each other, which serializes and prevents any overlap in their transmissions. Note a very low fraction of wireless transmissions from carrier sensing transmitters may still overlap if both the transmitters choose the same backoff period and access the channel in the same slot.

A.2.2 Rate anomaly

Rate anomaly can also lead to poor client performance. A passive conflict graph mechanisms can also generate information on rate anomaly that can be used to drive channel assignment or scheduling to provide time fairness to these clients. First, we analyze the impact of such rate anomaly problems in the UCSD WLAN deployment. Figure A.3 (a) shows the scatter plot between

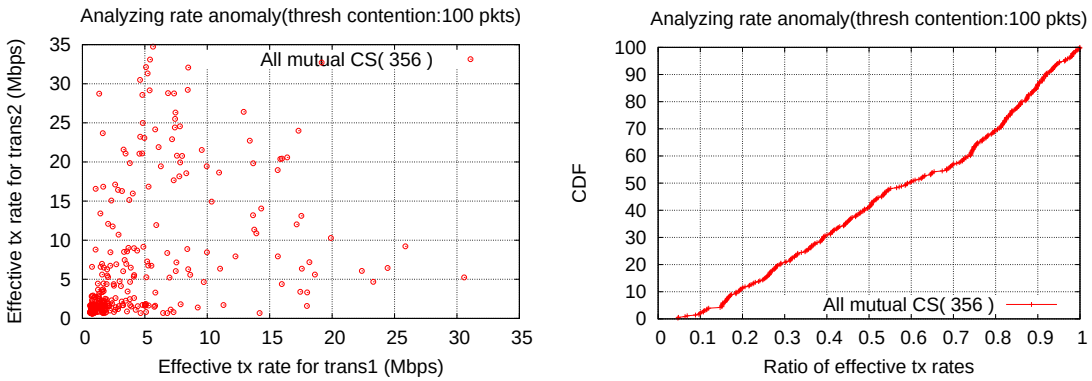


(a) CDF of overlap probability

(b) Diff in left and right overlap

Figure A.2: Carrier sense properties for Jigsaw trace. (a) CDF of overlap probability, separated by the type of transmitter pairs. Transmitter pairs are classified as downlink (both AP), uplink (both client) and mixed (one AP and one client). Note that very few downlink transmitter pairs (APs) actually have overlap probability less than 0.2, indicating the APs usually are not carrier sensing each other. This can be attributed to careful planning while deploying APs, aiming to maximize coverage by minimizing coverage overlaps between APs. (b) CDF of difference between left and right overlap probability. Note that pairs whose left and right overlap probability differ by less than 0.3, are likely not carrier sensing each other resulting in overlaps in both directions. While pairs whole left and right overlap probability differ by greater than 0.3 are likely experiencing asymmetric CS. Note that about 80% downlink transmitter pairs lie in mutual no CS range, while the distribution of uplink and mixed transmitter pairs is more uniform.

the effective rate of two links that are in mutual carrier sensing range. Figure A.3 (b) shows the distribution of the ratio of the effective transmission rates of every transmitter pair that carrier senses each other. Pairs close to (0,0) are the ones which have significant disparity between the transmission rates. In 40% of such transmitter pairs, the effective rates of two transmitters differ by a factor of 2X or more. This indicates that rate anomaly may play an important role in client performance for a significant fraction of transmitter pairs that are in carrier sensing range.



(a) Scatter plot

(b) Probability distribution

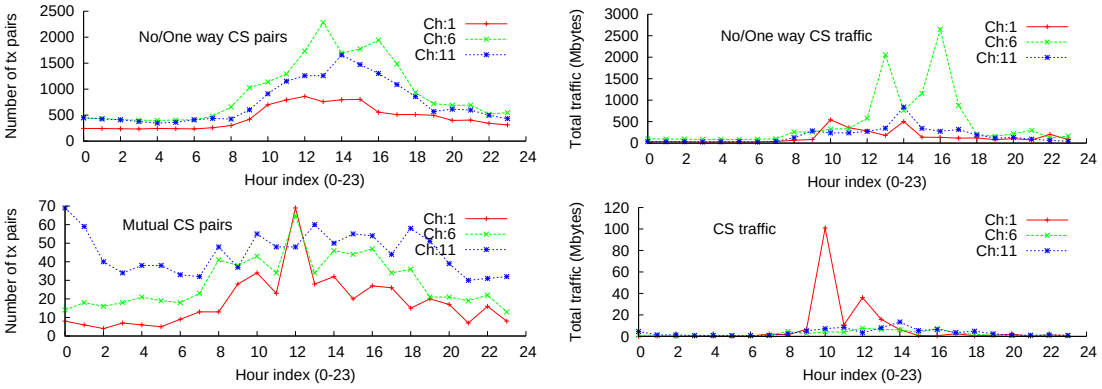
Figure A.3: Analyzing the extent of rate anomaly problem in the Jigsaw trace. (a) shows the scatter plot between effective rates of each transmitter pairs that are identified as being in the carrier sense range of each other. (b) plots the distribution of the ratio of these effective rates. Pairs closer to X = 1 indicate no rate anomaly, while those close to X = 0 indicate significant gap between the effective rates of two transmitters. Note that for about 40% of all transmitter pairs (142 out of 356), the effective tx rates of two contending transmitters differ by a factor of 2 or more.

Type	No CS (Overlap Diff < 0.3)	One way CS (Overlap Diff > 0.3)	Overlapping pairs
All	60%	40%	3287
Downlink	85%	15%	457
Uplink	30%	70%	582
Mixed	35%	65%	2248

Table A.1: Fraction of downlink/uplink/mixed links that do no carrier sensing or one way carrier sensing in Jigsaw trace.

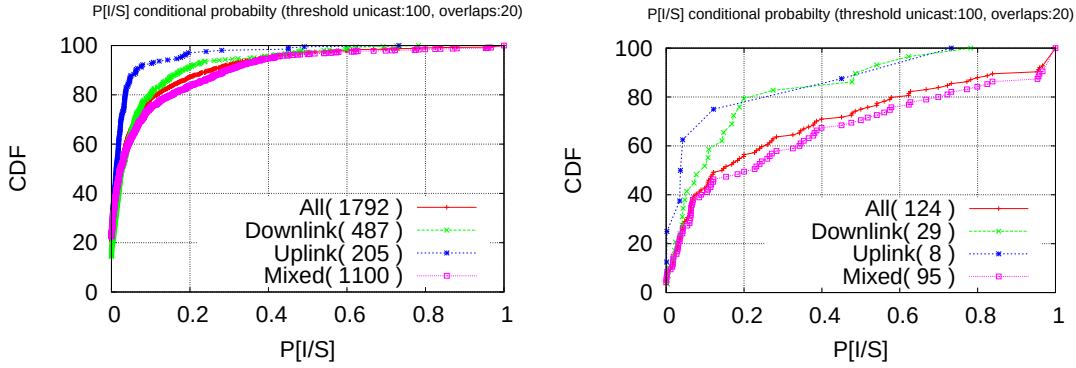
A.2.3 Traffic properties

Figure A.4 (a) and (b) shows the variation of total contending transmitter pairs and total traffic in the system binned by the hour of the day. We separate out transmitter pairs depending on their carrier sense properties. Clearly the density of transmitter pairs is highest around hours of 2 to 4



(a) Tx pair count (b) Total traffic

Figure A.4: Analyzing the traffic distribution for Jigsaw trace. (a) shows the temporal variation of number of transmitter pairs in the trace with time. We separate out transmitter pairs depending on their carrier sense properties. Clearly the density of transmitter pairs is highest around hours index of 14-16, which indicates the afternoon time. (b) shows the temporal variation of total traffic in the system due to these transmitter pairs. Note that the three channels show different loads at different hours. Using a dynamic conflict graph, can this load can be more evenly distributed using a traffic aware channel assignment mechanism.



(a) Conditional loss probability (b) Strongest interference per link

Figure A.5: Analyzing hidden interference for Jigsaw trace. (a) shows the distribution of loss probability due to hidden interference. (b) shows the distribution of loss due to strongest interferer for each link.

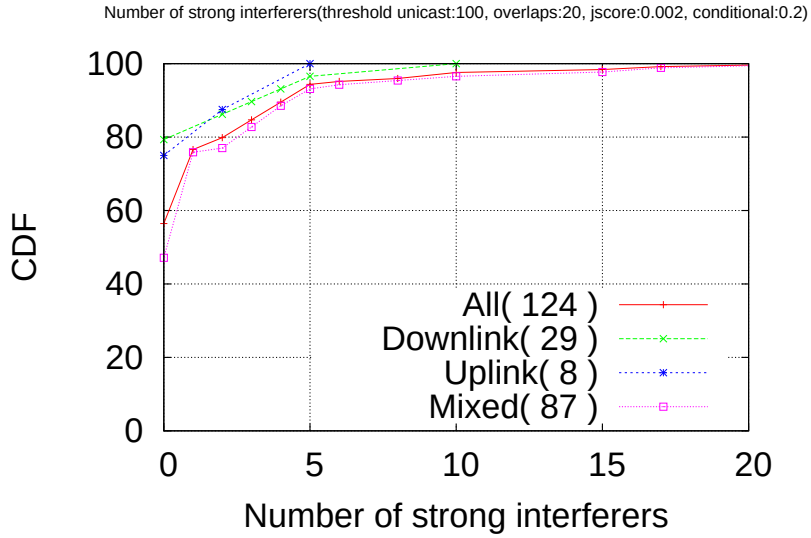


Figure A.6: This graph shows the number of significant hidden interferers per link that inflict at least a 20% additional loss at the receiver (on top of the background loss). We also take the JScore into account to filter out interferers that may be wrongly classified as sources of interference. We use a JScore threshold of 0.004 $nats$ to filter out valid interferers.

pm , which indicates heavy activity during the afternoon time. (b) shows the temporal variation of total traffic in the system due to these transmitter pairs. Note that the three channels show different loads at different hours. Using a dynamic conflict graph, can this load can be more evenly distributed using a traffic aware channel assignment mechanism.

A.3 Receiver side interference

Table A.2 shows the inferencing that can be performed on the basis of passive information obtained in the form of packet timestamps. As shown in Table A.2, two transmitters that are not in carrier sensing range, but interfere with each others clients, indicate hidden interference. Similarly, two transmitters that are mutually carrier sensing, but do not actually interfere with each others clients, indicate classic exposed terminal problem.

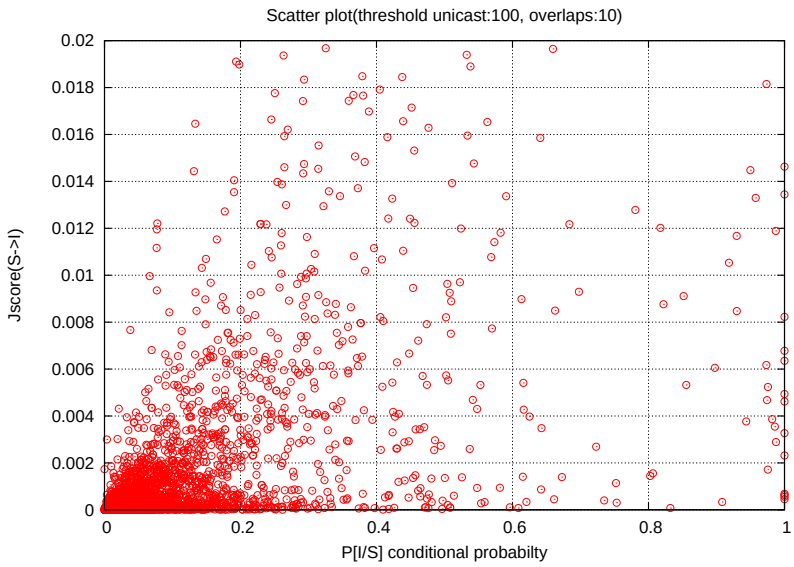


Figure A.7: Comparing the JScore and Conditional probability of different interferers in the Jigsaw trace. We consider only those interferers where JScore is greater than 0.004 and conditional loss probability is greater than 0.2.

$\text{overlap}(I, L) \Uparrow \wedge\, \text{loss}(L) \Uparrow \implies$	AP $I \rightarrow L$
$\text{overlap}(I, L) \Uparrow \wedge\, \text{loss}(L) \Downarrow \implies$	AP $I \mid L$
$\text{overlap}(I, L) \Downarrow \wedge\, \text{loss}(L) \Uparrow \implies$	Non conclusive
$\text{overlap}(I, L) \Downarrow \wedge\, \text{loss}(L) \Downarrow \implies$	Non conclusive

Table A.2: Premise for identifying whether a potential interferer I negatively impacts a link L. In this table, $\rightarrow$ indicates interference relationship, $\mid$ indicates no interference and the last two scenarios are inconclusive. Further, $\Uparrow$ and $\Downarrow$ indicates higher and lower side of the measures. When the overlap between the transmissions of the interferer I and link L is high (high overlap is denoted by the $\Uparrow$), and still the loss for link L is low, it indicates that I does not negatively impact the performance of L ($\implies$ AP $I \mid L$). Similarly, if the overlap is high and the loss is high, then we infer the I interferes with the link L ($\implies$ AP $I \rightarrow L$). Finally, if the overlap between interferer I and link L is low, we cannot assess the impact of the interferer on the link and hence the inference is inconclusive in those scenarios.

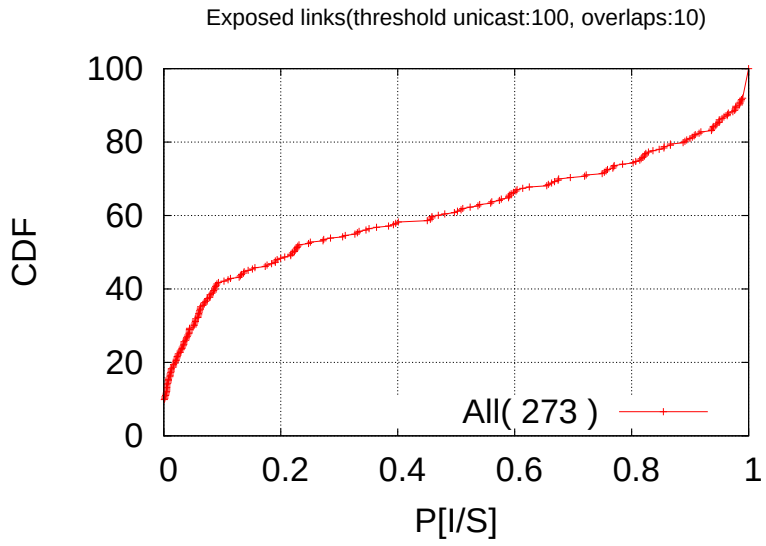


Figure A.8: Distribution of loss rates for exposed links. Even when two links carrier sense, they can choose the same slot for transmission with a probability of $\frac{2}{CW}$, where CW is the contention window of the transmitters. In this graph, we compute the loss rate of those packets which are transmitted in the same slot by two transmitters that are considered to be in carrier sensing range. Links with less than 20% loss indicate classic exposed terminal problem, where transmission could have proceeded simultaneously (with only 20% loss, 80% throughput), but instead shares the channel resulting in about 50% throughput.

A.3.1 Hidden interference

The authors in [44] identify 472 links that suffer from hidden interference in the monitored WLAN, out of which 56% of the links (264 links) are downlink in nature. The authors also define the interference loss rate $X = P_i \times (n_x/n)$ as the fraction of all transmissions that are lost due to simultaneous transmissions and show that about 15% of all links can have interference loss rates of more than 10%. Figure A.5 (a) shows the distribution of losses for all links for all potential interferers. Each sender, interferer pair is categorized as downlink, uplink and mixed category as described before. We observe that all types of transmitter pairs show similar loss tendencies, with the uplink sender, transmitter pairs showing slightly less losses than downlink and mixed categories. About 18% of interferers inflict more than 20% losses on their respective victim links.

Figure A.5 (b) shows the impact of the strongest hidden interferer on the loss rate of a link. Here each link is just considered once, while in (a) there is a point for each (link, interferer) pair if at least 20 packets overlap for the link and the interferer. Notice that the impact of the strongest interferer can be quite significant. About 40% of the links can suffer upto 20% of losses or more due to their strongest hidden interferer.

However, the exact impact of each interferer on the given link also depends on the frequency with which their packets overlap with the packet transmissions on a given link. In order to characterize the significant interferers, we make use of JScore presented in [84], which also takes into account the frequency of interference. Figure A.7 shows the scatter plot between JScore and conditional loss probability. As shown in the Figure, there can be lot of interferers for which conditional probability is high but JScore is low, indicating that such interference occurs very infrequently, hence may not be practically significant. So we consider only those interferers where JScore is greater than 0.004 and conditional loss probability is greater than 0.2. After filtering out only significant interferers, Figure A.6 shows the number of interferes per link that classify as strong relevant interferers. Notice that about 60% of the links do not have any strong interferers. However about 20% of links can have at least one such strong interferer. This indicates that although hidden interference is limited to few links but it can cause significant losses on those links.

A.3.2 Exposed interference

Figure A.8 shows the loss rates due to interferers which can be sensed by the sender. Even when two transmitters carrier sense, they can choose the same slot for transmission with a probability of $\frac{2}{CW}$, where CW is the contention window of the transmitters. In this graph, we compute the loss rate of those packets which are transmitted in the same slot by two transmitters that are considered to be in carrier sensing range. Links with less than 20% loss indicate classic exposed terminal problem, where transmission could have proceeded simultaneously, but instead shares the channel resulting in about 50% throughput for each.

Downlink hidden interference We extend their analysis to investigate the probability of hidden interference between two downlink flows in the WLAN. We restrict our analysis to the 264 downlink flows that suffer from hidden interference. More precisely, we analyze the jigsaw trace to find $P[I \mid S]$ for scenarios where both the unicast transmission (s,r) and the other simultaneous transmissions originate from the APs. If we denote the set of production Access Points in the Jigsaw trace as $AP = AP_1, AP_2 \ldots$, then we require that

$$s \in AP \wedge \forall (s_i, r_i) \in L, s_i \in AP \tag{A.1}$$

We recompute the interference loss rates for the downlink transmissions that suffer from hidden interference from other APs. Figure A.9 shows the distribution of interference loss rates (X) and the probability of loss due to simultaneous downlink transmissions $P[I/S]$. As shown in the figure, only 5% of all downlink transmissions suffer from interference loss rate of greater than 0.1. However, about 11% of downlink transmissions have a probability of loss due to hidden interference greater than 0.1. Note that each downlink transmission might be interfered by multiple simultaneous downlink transmissions, which might reduce $P[I/S]$, if some interferers are weaker than the others. In order to illustrate this, we also plot the $P[I/S]$ for only the strongest interferer for each downlink transmission. About 30% of all links may suffer a loss of greater than 10% when operated in parallel with its most significant interferer. We believe, that depending on the traffic patterns and load on the WLAN, the performance of such downlink clients can be severely degraded under normal DCF mode of operation.

A.4 Summary

Following are the key observations from the analysis of Jigsaw trace. We also report some key implications of these observations on the design of interference mitigation and estimation mechanisms.

- *Observation:* Only 10% of all contending transmitter pairs perform mutual carrier sense. About 40% of links are outside the carrier sensing range. Remaining links fall in the category

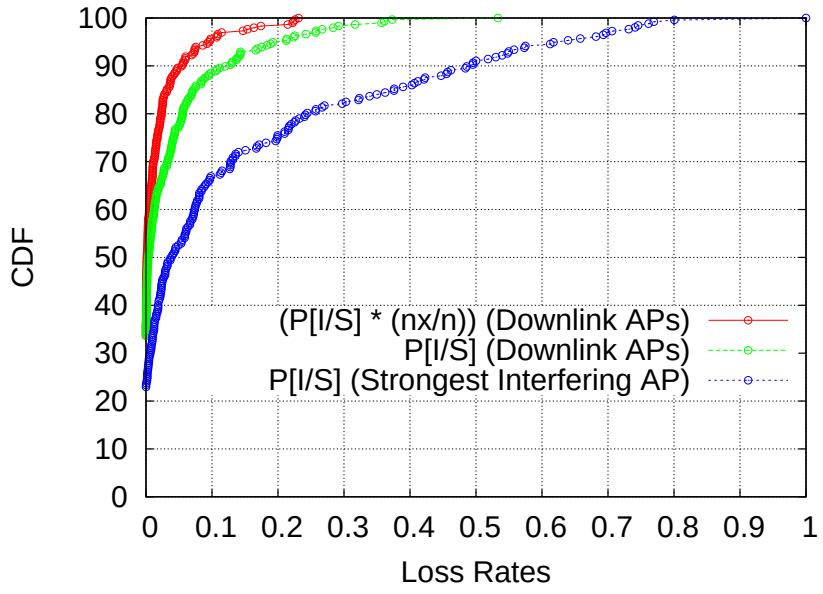


Figure A.9: Impact of downlink hidden terminals in a production WLAN.

of asymmetric carrier sensing. About 85% of the downlink transmitter pairs do no carrier sense each other.

  - **Implication:** Since majority of downlink transmitter pairs do not carrier sense each other, so there is high probability that clients falling in the interference range of two downlink transmitters might suffer from hidden terminal problem.

- *Observation:* Traffic distribution across channels is not uniform and varies significantly depending on the time of the day.

  - **Implication:** Traffic aware channel assignment can provide better load balancing across different channels by dynamically adjusting the channel of operation for different APs according to load on each channel. A conflict graph augmented with traffic information can be useful for dynamic traffic aware channel assignment.

- *Observation:* About 40% of all transmitter pairs indicate a factor of two or more difference in their effective data rates. This indicates significant rate anomaly in the network, which may slow down clients using higher data rates (for example 802.11n clients).
  - *Implication:* Annotating conflict graph with effective transmission rates of different links can be helpful in diagnosing slow or variable client performance. Moreover, once rate anomaly is detected, we can switch those clients under centralized channel access to ensure the performance of high throughput clients is not impacted by slower clients.
- *Observation:* About 35% of the links have a hidden interferer that can cause at least 20% packet loss. About 20% of links have a interferer that can cause 60% packet loss.
  - **Implication:** Hidden terminal occurs infrequently, but when it occurs, it can cause significant performance loss for the victim clients.
- *Observation:* About 60% of the links have no hidden interferer, 20% have one strong interferer and 3% of the links have two interferer.
  - **Implication:** Majority of hidden terminal links have only one strong interferer, which is an important factor for simplifying the interference estimation and mitigation mechanisms. For example, a mechanism like CENTAUR is especially effective in mitigating such interference from single strong interferers.
- *Observation:* About 44% of the links in which the sender senses the interferer have a loss rate of less than 20% when both the sender and the interferer are active simultaneously. These links can be classified as potentially exposed terminal links.
  - **Implication:** Exposed terminal links can be identified without disabling the carrier sensing of the transmitters, as transmissions from even carrier sensing transmitters still overlap when the transmitters choose the same value for random backoff. Such exposed terminal links can be put under centralized channel access to enhance their performance using a mechanism like CENTAUR .

Summarizing, this study shows the value of estimating and mitigating interference in production WLANs. It shows that providing a fine grained conflict graph, from a mechanism like PIE, can allow us to solve hidden and exposed terminals, using mechanisms like CENTAUR and Model-TPC.

LIST OF REFERENCES

[1] Aerohive Networks Inc. URL: http://www.aerohive.com/. 2

[2] Aruba Mobility Controller. URL: http://www.arubanetworks.com/products/mobility_controllers.php. 2

[3] Aruba Networks Inc., A Closer Look at Wireless Intrusion Detection: How to Benefit from a Hybrid Deployment Model. White paper. Retrieved 2010. URL: http://www.arubanetworks.com/pdf/technology/whitepapers/wp_Hybrid_WIDS.pdf. 2

[4] Aruba Networks Inc., User-Centric Distributed Enterprise Network. White paper. 2009. URL: http://www.arubanetworks.com/technology/distributed_enterprise_network.php. 166

[5] Business WLAN market analysis. http://www.researchandmarkets.com/reports/c19271. 4

[6] Cisco 4400 series wireless lan controllers. http://www.cisco.com/en/US/products/ps6366/index.html. 2

[7] Cisco Wireless Control System. http://www.cisco.com/en/US/products/ps6305/. 2, 25

[8] Extricom Inc. URL: http://www.extricom.com. 166

[9] Fulfilling the promise of 802.11n in the enterprise without compromise. White paper, 2007. URL: http://www.merunetworks.com/pdf/whitepapers/WP_Fulfilling_the_Promise_802.11n_v4.pdf. 166

[10] High Resolution Timers Home Page. http://high-res-timers.sourceforge.net/. 51

[11] IEEE 802.11e. URL: www.ieee802.org/11/. 18, 22

[12] Infonetics research. http://www.infonetics.com/. 1

[13] Intel pro/wireless network connection for mobile. http://www.intel.com/network/connectivity/products. 52

[14] Madwifi open source driver. http://madwifi-project.org. 123

[15] Meru Controllers. URL: http://www.merunetworks.com/products/controllers.php. 2

[16] The network simulator - ns-2. URL: http://www.isi.edu/nsnam/ns/. 19, 20, 31

[17] ONOE rate control. http://madwifi.org/browser/trubk/ath_rate/onoe. 108

[18] Soekris Engineering. URL: http://www.soekris.com. 52

[19] Supported RADIUS attributes on the wireless LAN controller. URL: http://www.cisco.com/en/US/products/ps6307. 2

[20] XIRRUS: High Performance WiFi. URL: http://www.xirrus.com/. 2

[21] A. Munaretto, M. Fonseca, K.A. Agha, and G. Pujolle. Fair time sharing protocol: A solution for IEEE 802.11b hot spots. In *Lecture Notes in Computer Science*, 2004. 19

[22] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *ACM SIGCOMM*, 2004. 33

[23] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of rf interference. In *ACM CoNEXT*, 2008. 28, 33, 34, 47, 64, 69, 124

[24] N. Ahmed and S. Keshav. Smarta: A self-managing architecture for thin access points. In *ACM CoNEXT*, 2006. 28, 33, 34, 37, 116, 135, 154

[25] Nabeel Ahmed, Usman Ismail, Konstantina Papagiannaki, and Srinivasan Keshav. Measuring multi-parameter conflict graphs for 802.11 networks. *Mobile Computing and Communications Review*, 2009. 73, 137

[26] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *ACM MobiCom*, 2005. 28, 30, 88, 107

[27] D.W. Allan. Time and frequency (time domain) characterization, estimation and prediction of precision clocks and oscillators. In *IEEE Transactions*, 1987. 94

[28] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad hoc wireless networks. In *ACM MobiCom*, 2004. 27, 28

[29] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. White space networking with Wi-Fi like connectivity. In *ACM SIGCOMM*, 2009. 165, 166

[30] T. P. Baker and G. M. Scallon. An architecture for real-time software systems. In *Tutorial: Hard real-time systems*, 1989. 23

[31] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. Characterizing user behavior and network performance in a public wireless LAN. *SIGMETRICS Performance Evaluation Review*, 2002. 35

[32] Aruna Balasubramanian, Ratul Mahajan, Arun Venkataramani, Brian Neil Levine, and John Zahorjan. Interactive WiFi connectivity for moving vehicles. In *ACM SIGCOMM*, 2008. 70

[33] G. Beccari, S. Caselli, and F. Zanichelli. A technique for adaptive scheduling of soft real-time tasks. *Real-Time Systems*, 2005. 22

[34] Y. Bejerano and R. Bhatia. MiFi: a framework for fairness and QoS assurance in current IEEE 802.11 networks with multiple access points. In *IEEE INFOCOM*, 2004. 18, 21, 38

[35] Vaduvur Bharghavan, Alan J. Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In *ACM SIGCOMM*, 1994. 41

[36] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. In *IEEE Journal on Selected Areas in Communications*, 2000. 33

[37] Ioannis Broustis, Jakob Eriksson, Srikanth V. Krishnamurthy, and Michalis Faloutsos. Implications of power control in wireless networks: A quantitative study. In *Passive and Active Measurement Conference (PAM)*, 2007. 29

[38] Ioannis Broustis, Konstantina Papagiannaki, Srikanth V. Krishnamurthy, Michalis Faloutsos, and Vivek Mhatre. MDG: measurement-driven guidelines for 802.11 WLAN design. In *ACM MobiCom*, 2007. 28, 30

[39] Kan Cai, Michael Blackstock, Michael J. Feeley, and Charles Krasic. Non-intrusive, dynamic interference detection for 802.11 networks. In *IMC*, 2009. 33

[40] Joseph Camp, Joshua Robinson, Christopher Steger, and Edward Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *ACM MobiSys*, 2006. 20

[41] Gene D. Carlow. Architecture of the space shuttle primary avionics software system. *Communications ACM*, 1984. 23

[42] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. Ethane: taking control of the enterprise. In *ACM SIGCOMM*, 2007. 163

[43] Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Péter Benkö, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker. Automating cross-layer diagnosis of enterprise wireless networks. In *ACM SIGCOMM*, 2007. 28, 33, 36, 116, 123, 165

[44] Yu-Chung Cheng, John Bellardo, Péter Benkö, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *ACM SIGCOMM*, 2006. 8, 28, 33, 36, 42, 46, 116, 122, 123, 129, 156, 164, 168, 169, 170, 178

[45] T.M. Cover and J.A. Thomas. *Elements of Information Theory*, 1991. 99

[46] Mark E. Crovella, Robert Frangioso, and Mor Harchol-balter. Connection scheduling in web servers. In *USENIX Symposium on Internet Technologies and Systems*, 1999. 22

[47] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM*, 1989. 23, 24

[48] Dan Duchamp and Neil F. Reynolds. Measured performance of a Wireless LAN. In *IEEE Conference on Local Computer Networks*, 1992. 35

[49] David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *ACM SIGCOMM*, 1996. 35

[50] Stephen William Edge. An adaptive timeout algorithm for retransmission across a packet switching network. In *ACM SIGCOMM*, 1984. 50

[51] Jakob Eriksson, Sharad Agarwal, Paramvir Bahl, and Jitendra Padhye. Feasibility study of mesh networks for all-wireless offices. In *ACM MobiSys*, 2006. 7

[52] Jakob Eriksson, Sharad Agarwal, Paramvir Bahl, and Jitendra Padhye. Feasibility study of mesh networks for all-wireless offices. In *ACM MobiSys*, 2006. 46, 70, 146

[53] Charles Reis et al. Measurement-based models of delivery and interference in static wireless networks. In *ACM SIGCOMM*, 2006. 28, 33, 34, 82, 90

[54] Fehmi Ben Abdesslem et al. On the feasibility of power control in current IEEE 802.11 devices. In *PERCOMW 2006*. 81, 84

[55] Violeta Gambiroza, Bahareh Sadeghi, and Edward W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *ACM MobiCom*, 2004. 18, 20

[56] Yan Gao and Dah ming Chiu. Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications. In *ACM SIGMETRICS*, 2006. 33

[57] Michele Garetto, Theodoros Salonidis, and Edward W. Knightly. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. In *IEEE INFOCOM*, 2006. 32, 33

[58] Michael Gastpar and Martin Vetterli. On the capacity of wireless networks: The relay case. In *IEEE INFOCOM*, 2002. 32, 33

[59] Ye Ge, Jennifer C. Hou, and Sunghyun Choi. An analytic study of tuning systems parameters in IEEE 802.11e enhanced distributed channel access. *Computer Networks*, 2007. 18, 22

[60] J. Geier. Assigning 802.11b access point channels. *Wi-Fi Planet 2004*. 152

[61] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *ACM SIGCOMM*, 2008. 18, 24, 26, 166

[62] Javier Gomez, Andrew T. Campbell, Mahmoud Naghshineh, and Chatschik Bisdikian. Paro: supporting dynamic power controlled routing in wireless ad hoc networks. *Wireless Networks*, 2003. 29, 31

[63] Matthias Grossglauser and David Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of IEEE INFOCOM*, 2001. 32

[64] P. Gupta and P. Kumar. Capacity of wireless networks. In *IEEE Transactions on Information Theory*. 33, 80

[65] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, March 2000. 32

[66] David Hadaller, Srinivasan Keshav, Tim Brecht, and Shubham Agarwal. Vehicular opportunistic communication under the microscope. In *ACM MobiSys*, 2007. 18

[67] Mohammad T. Hajiaghayi, Sayyed Vahab Mirrokni, and Amin Saberi. Cell breathing in wireless LANs: Algorithms and evaluation. *IEEE Transactions on Mobile Computing*, 2007. 163

[68] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *ACM MobiCom*, 2008. 18, 24, 26, 166

[69] Prashanth Hande, Sundeep Rangan, and Mung Chiang. Distributed uplink power control for optimal SIR assignment in cellular data networks. In *IEEE INFOCOM*, 2006. 112

[70] Mor Harchol-Balter, Bianca Schroeder, Nikhil Bansal, and Mukesh Agrawal. Size-based scheduling to improve web performance. *ACM Transactions Computer Systems*, 2003. 22

[71] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. In *ACM MobiCom*, 2004. 35

[72] Felix Hernandez-Campos and Maria Papadopouli. A comparative measurement study of the workload of wireless access points in campus networks. In *IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, 2005. 35

[73] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *IEEE Infocom*, 2003. 156

[74] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec, IEEE 802.11 Standard. *IEEE Standard 802.11*, 2003. 123

[75] V. Bharghavan J. Monks and W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *IEEE INFOCOM*, 2001. 28, 29, 32, 80, 84, 87

[76] Jain R. et. al. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR-301, DEC Research Report, September 1984. 66

[77] Kyle Jamieson and Hari Balakrishnan. PPR: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, 2007. 18, 25, 166

[78] Amit P. Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *ACM SIGCOMM E-WIND*, 2005. 28, 33, 35

[79] Moustafa Youssef Jihwang Yeo and Ashok Agrawala. Characterizing the IEEE 802.11 Traffic: The Wireless Side. In *International Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo)*, 2005. 28, 33, 35

[80] Dilip A. Joseph, Arsalan Tavakoli, and Ion Stoica. A policy-aware switching layer for data centers. In *ACM SIGCOMM*, 2008. 163

[81] Bong jun Ko and Vishal Misra. Distributed channel assignment in multi-radio 802.11 mesh networks. In *WCNC*, 2007. 28

[82] Eun-Sun Jung and Nitin H. Vaidya. A power control mac protocol for ad hoc networks. *Wireless Networks*, 2005. 28, 29, 31

[83] Abdul Kabbani. Distributed low-complexity maximum throughput scheduling in wireless backhaul networks. 2006. 18, 20

[84] Srikanth Kandula, Ranveer Chandra, and Dina Katabi. What's Going On? Learning Communication Rules in Edge Networks. In *ACM SIGCOMM*, 2008. 179

[85] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. In *ACM MobiCom*, 2001. 18, 19, 38

[86] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. BLINC: multilevel traffic classification in the dark. In *ACM SIGCOMM*, 2005. 162

[87] P. Karn. MACA — A new channel access method for packet radio. In *Proceedings of ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990. 29, 31, 41

[88] Anand Kashyap, Samrat Ganguly, and Samir R. Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *ACM MobiCom*, 2007. 28, 32, 33

[89] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. Embracing wireless interference: Analog network coding. In *ACM SIGCOMM*, 2007. 26

[90] Vikas Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *IEEE INFOCOM*, 2003. 30

[91] A. A. Khan, C. L. McCreary, and M. S. Jones. A comparison of multiprocessor scheduling heuristics. In *International Conference on Parallel Processing, volume II*, 1994. 22

[92] Kyu-Han Kim and Kang G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. In *ACM MobiCom*, 2006. 33

[93] Ramana Rao Kompella, Sriram Ramabhadran, Ishwar Ramani, and Alex C. Snoeren. Cooperative packet scheduling via pipelining in 802.11 wireless networks. In *ACM SIGCOMM E-WIND*, 2005. 21

[94] Ketan Kotecha and Apurva Shah. Adaptive scheduling algorithm for real-time operating system. 2008. 22

[95] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *ACM MobiCom*, 2002. 35

[96] B. Krishnamachari, S. Wicker, R. Bejar, and C. Fernandez. On the complexity of distributed self-configuration in wireless networks. *Telecommunication Systems*, 2003. 27, 28

[97] A. Ksentini, M. Naimi, A. Nafaa, and M. gueroui. Adaptive service differentiation for qos provisioning in IEEE 802.11 wireless ad hoc networks. In *PE-WASUN*, 2004. 18, 22

[98] S. Kullback. *Information theory and statistics*, 1959. 100

[99] Anurag Kumar, Eitan Altman, Daniele Miorandi, and Munish Goyal. New insights from a fixed-point analysis of single cell ieee 802.11 wlans. *IEEE/ACM Transactions on Networking*, 2007. 32, 33

[100] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. Algorithmic aspects of capacity in wireless networks. In *SIGMETRICS*, 2005. 32

[101] K. Leung and L. Wang. Controlling QoS by Integrated Power Control and Link Adaptation in Broadband Wireless Networks. In *European Transactions on Telecommunications*, 1999. 108, 109

[102] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *ACM MobiCom*, 2001. 32, 33

[103] Shu lin Wu, Yu-Chee Tseng, and Jang ping Sheu. Intelligent medium access for mobile ad hoc networks with busy tones and power control. *IEEE Journal on Selected Areas in Communications*, 2000. 29

[104] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 1973. 22, 23

[105] Qingwen Liu, Shengli Zhou, and Georgios B. Giannakis. Cross-layer scheduling with prescribed qos guarantees in adaptive wireless networks. *IEEE Journal on Selected Areas in Communications*, 2005. 18, 21, 22

[106] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Analyzing the mac-level behavior of wireless networks in the wild. *ACM SIGCOMM*, 2006. 28, 33, 35, 36, 116, 120, 122, 123, 139, 164, 170

[107] Filipe Mazzini, Geraldo Mateus, and James Macgregor Smith. Lagrangean based methods for solving large-scale cellular network design problems. *Journal of Wireless Networks*, 2003. 27, 28

[108] Vivek P. Mhatre, Konstantina Papagiannaki, and Francois Baccelli. Interference mitigation through power control in high density 802.11 WLANs. In *IEEE INFOCOM*, 2007. 29

[109] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for WLANs. *Mobile Computer Communications Review (MC2R)*, 2005. 27

[110] A. Mishra, V. Brik, S. Banerjee, and W. Arbaugh. A client-driven approach for channel management in wireless LANs. In *IEEE Infocom*, April 2006. 27, 28

[111] Arunesh Mishra, Vladimir Brik, Suman Banerjee, Aravind Srinivasan, and William A. Arbaugh. A client-driven approach for channel management in wireless lans. In *IEEE INFOCOM*, 2006. 151

[112] Allen K. Miu, Hari Balakrishnan, and Can E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *MobiCom*, 2005. 85

[113] A.K.-L. Mok. Fundamental design problems of distributed systems for the hard real-time environment. *PhD Thesis*, 1983. 23

[114] Thomas Moscibroda and Roger Wattenhofer. Minimizing interference in ad hoc and sensor networks. In *DIALM-POMC*, 2005. 29

[115] Alaa Muqattash and Marwan Krunz. Power controlled dual channel (PCDC) medium. In *IEEE INFOCOM*, 2003. 29

[116] Alaa Muqattash and Marwan Krunz. A single-channel solution for transmission power control in wireless ad hoc networks. In *ACM MobiHoc*, 2004. 29

[117] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *ACM MobiCom*, 2000. 38

[118] Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *European Wireless Conference*, 2002. 28, 29, 30, 32

[119] Naveen Kumar Santhapuri, Justin Manweiler, Souvik Sen, Romit Roy Choudhury, Srihari Nelakuditi, Kamesh Munagala. Message in Message (MIM): A Case for Shuffling Transmissions in Wireless Networks. In *ACM Hotnets*, 2008. 18, 20

[120] Sergiu Nedevschi, Rabin K. Patra, Sonesh Surana, Sylvia Ratnasamy, Lakshminarayanan Subramanian, and Eric Brewer. An adaptive, high performance mac for long-distance multihop wireless networks. In *ACM MobiCom*, 2008. 20

[121] Giao T. Nguyen, Randy H. Katz, Brian Noble, and Mahadev Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *WSC: 28th conference on Winter simulation*, 1996. 35

[122] Nan Ni and Laxmi N. Bhuyan. Fair scheduling and buffer management in internet routers. In *Proc. IEEE INFOCOM*, 2002. 24

[123] Dragos Niculescu. Interference map for 802.11 networks. In *IMC*, 2007. 28, 33, 34, 37, 116

[124] C. Oestges and B. Clerckx. MIMO wireless communications. In *Academic Press*, 2003. 165, 166

[125] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *IMC*, 2005. 28, 33, 34, 122, 132

[126] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transaction on Networking*, 1993. 23, 24

[127] Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *NSDI*, 2007. 20

[128] M.B. Pursley, H.B. Russell, and J.S. Wysocarski. Energy-efficient transmission and routing protocols for wireless multiple-hop networks and spread-spectrum radios. In *EUROCOMM*, 2000. 29, 31

[129] Daji Qiao, Sunghyun Choi, Amit Jain, and K. G. Shin. Adaptive transmit power control in IEEE 802.11a wireless LANs. In *Vehicular Technology Conference*, 2003. 29

[130] Daji Qiao, Sunghyun Choi, Amit Jain, and Kang G. Shin. MiSer: an optimal low-energy transmission strategy for IEEE 802.11a/h. In *ACM MobiCom*, 2003. 28, 29, 31, 32

[131] Lili Qiu, Yin Zhang, Feng Wang, Mi Kyung Han, and Ratul Mahajan. A general model of wireless interference. In *ACM MobiCom*, 2007. 33, 34

[132] Kishore Ramachandran, Ravi Kokku, Honghai Zhang, and Marco Gruteser. Symphony: synchronous two-phase rate and power control in 802.11 WLANs. In *ACM MobiSys*, 2008. 28, 29, 30, 32, 154

[133] Krithi Ramamritham and John A. Stankovic. Scheduling algorithms and operating systems support for real-time systems. In *Proceedings of the IEEE*, 1994. 22

[134] Bhaskaran Raman and Kameswari Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *ACM MobiCom*, 2005. 20

[135] Ananth Rao and Ion Stoica. An overlay mac layer for 802.11 networks. In *ACM MobiSys*, 2005. 20

[136] T. Rappaport. *Wireless Communications: Principle and Practice*. Prentice Hall, 1996. 27, 28

[137] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 1998. 29

[138] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, and John Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *ACM SIGCOMM E-WIND*, 2005. 28, 35, 36, 146

[139] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, John Zahorjan, and Ed Lazowska. CRAWDAD data set uw/sigcomm2004. 46, 70, 146

[140] Zvi Rosberg. Asymptotically optimal transmission power and rate control for CDMA channels with multiple user classes. In *IEEE INFOCOM*, 2005. 112

[141] Eric Rozner, Yogita Mehta, Aditya Akella, and Lili Qiu. Traffic-aware channel assignment in enterprise wireless LANs. In *ICNP*, 2007. 151

[142] Bianca Schroeder and Mor Harchol-Balter. Web servers under overload: How scheduling can help. *ACM Transactions Internet Technol.*, 2006. 22

[143] Anmol Sheth, Christian Doerr, Dirk Grunwald, Richard Han, and Douglas C. Sicker. Mojo: a distributed physical layer anomaly detection system for 802.11 WLANs. In *ACM MobiSys*, 2006. 33, 35

[144] Anmol Sheth and Richard Han. SHUSH : Reactive Transmit Power Control for Wireless MAC Protocols. In *WICON*, 2005. 28, 29, 32, 80, 84, 87

[145] Anmol Sheth, Sergiu Nedevschi, Rabin Patra, Sonesh Surana, and Eric Brewer. Packet loss characterization in wifi-based long distance networks. In *IEEE INFOCOM*, 2007. 20

[146] Jingpu Shi, Omer Gurewitz, Vincenzo Mancuso, Joseph Camp, and Edward W. Knightly. Measurement and modeling of the origins of starvation in congestion-controlled mesh networks. In *IEEE INFOCOM*, 2008. 20

[147] M. Shreedhar and George Varghese. Efficient fair queueing using deficit round robin. In *ACM SIGCOMM*, 1995. 24

[148] Vivek Shrivastava, Dheeraj Agrawal, Arunesh Mishra, Suman Banerjee, and Tamer Nadeem. Understanding the limitations of transmit power control for indoor WLANs. In *IMC*, 2007. 15, 160

[149] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. CENTAUR: realizing the full potential of centralized WLANs through a hybrid data path. In *ACM MobiCom*, 2009. 15, 116, 119, 154, 157

[150] P. Smulders. Exploiting the 60 ghz band for local wireless multimedia access: prospects and future directions. *Communications Magazine, IEEE*, 2002. 165, 166

[151] Sharad Agarwal Srikanth, Sharad Agarwal, Srikanth V. Krishnamurthy, Randy H. Katz, and Son K. Dao. Distributed power control in ad-hoc wireless networks. In *PIMRC*, 2001. 31

[152] Tamer Elbatt Srikanth, Srikanth V. Krishnamurthy, Dennis Connors, and Son Dao. Power management for throughput enhancement in wireless ad-hoc networks. In *IEEE International Conference on Communications*, 2000. 29

[153] D. Stiliadis and A. Varma. Providing bandwidth guarantees in an input-buffered crossbar switch. In *IEEE INFOCOM*, 1995. 24

[154] John A. Stine and Gustavo de Veciana. A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states. *IEEE Journal on Selected Areas in Communications*, 2004. 18, 22

[155] M. Subbarao. Dynamic power-conscious routing for manets: An initial approach. In *IEEE Vehicular Technology Conference*, 1999. 28, 31, 88

[156] Godfrey Tan and John Guttag. Time-based fairness improves performance in multi-ra te wlans. In *Proc. of USENIX*, 2004. 18, 19, 157

[157] Diane Tang and Mary Baker. Analysis of a metropolitan-area wireless network. In *Mobile Computing and Networking*, 1999. 35

[158] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *ACM MobiHoc*, 2005. 18

[159] Ilenia Tinnirello, Sunghyun Choi, and Youngsoo Kim. Revisit of RTS/CTS exchange in high-speed IEEE 802.11 networks. In *WOWMOM*, 2005. 44

[160] N. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a Wireless LAN. In *ACM MobiCom*, 2000. 18, 19, 38

[161] Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan. Harnessing Exposed Terminals in Wireless Networks. In *NSDI*, 2008. 18, 20, 26, 33, 37, 45, 60, 66, 137

[162] White-paper from Aruba Networks. Advanced RF management for wireless grids. http://www.arubanetworks.com/pdf/rf-for-grids.pdf. 7, 25, 46, 161, 166

[163] White-paper from Meru Networks. Microcell deployments: Making a bad problem worse for pervasive wireless LAN deployments. http://www.merunetworks.com/pdf/whitepapers/. 25

[164] White-paper from Meru Networks. Virtual cells: The only scalable multi-channel deployment. http://www.merunetworks.com/pdf/whitepapers/. 25, 166

[165] Grace R. Woo, Pouya Kheradpour, Dawei Shen, and Dina Katabi. Beyond the bits: cooperative packet recovery using physical layer information. In *ACM MobiCom*, 2007. 18, 25

[166] Yang Xiao, Frank Haizhon Li, and Sunghyun Choi. Two-level protection and guarantee for multimedia traffic in IEEE 802.11e distributed WLANs. *Wireless Networks*, 2009. 18, 22

[167] Hongyu Yang, Lixia Xie, and Jizhou Sun. Intrusion detection solution to WLANs. *IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, 2004. 2

[168] Chi-Hsiang Yeh. IPMA: An Interference/Power-aware MAC Scheme for Heterogeneous Wireless Networks. *ISCC*, 2003. 80, 84, 88

[169] Jihwang Yeo, Moustafa Youssef, and Ashok Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSe*, 2004. 28, 33, 35, 123

[170] Yoo S-H, Choi J-H, Hwang J-H, and Yoo C. Eliminating the performance anomaly of 802.11b. In *In Lecture Notes in Computer Science*, 2005. 19

[171] Yang Ziao and Yi Pan. Differentiation, QoS guarantee, and optimization for real-time traffic over one-hop ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 2005. 22