PERFORMANCE IMPROVEMENTS FOR UNPLANNED HIGH DENSITY WIRELESS LANS

BY MESUT ALI ERGIN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Marco Gruteser

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2010

ABSTRACT OF THE DISSERTATION

Performance Improvements for Unplanned High Density Wireless LANs

by Mesut Ali Ergin

Dissertation Director: Marco Gruteser

Chaotic unplanned IEEE 802.11 WLAN deployments are becoming the norm and such residential deployments have many nearby access points (APs) and stations on the same channel, either due to lack of coordination or insufficient available channels. Thus, inter-cell interference in these high-density settings is common but not well-understood. Our evaluations for such interfering deployments reveal that up-to two-thirds of the WLAN system capacity may be lost in a typical large-apartment building with 50 interfering WLANs

In this thesis, we first report on our analysis of high-density unplanned WLANs' performance under realistic scenarios. We find that with a typical TCP-dominant workload, cumulative system throughput is characterized by the number of actively interfering APs rather than the number of clients. We verify that due to TCP flow control, the number of backlogged stations in such a network equals twice the number of active APs. Thus, a single AP network proves very robust even with over one hundred clients, while multiple interfering APs lead to a significant increase in collisions that reduces throughput and affects multimedia traffic.

Based on our analysis, we suggest a practical contention window adaptation technique, $WiPhi$, using information on the number of nearby APs rather than clients. We also point out the need for collision-resilient rate adaptation in such a setting. Together these techniques can largely recover the loss in cumulative throughput in a setting with strongly interfering APs.

We then propose an alternative ISP-level solution, $HeedNet$, recovering lost performance by scheduling the IP packets of the bulk traffic at the ISP edge-router towards interfering APs. It requires no changes to the MAC protocol and the APs of the network, making it a viable solution for ISPs. We evaluate $HeedNet$ via simulations and an actual deployment to show that a significant portion of the lost system capacity can be regained (more than 2.2X improvement compared to legacy). $HeedNet$ also increases the fairness, reducing starvation among WLANs. Additionally, we show that $HeedNet$ improves the performance of the non-scheduled (i.e., non-bulk) traffic considerably, such as VoIP, due to the reduced-collision rate environment it creates.

Acknowledgements

I would like to express my sincere gratitude to my academic advisor, Professor Marco Gruteser, for his guidance and support throughout my Ph.D. study. For many times in the progress of my research work, it was his enlightening thoughts and patient advices that helped me overcome the difficulties I faced with. Special thanks go to Professor Dipankar Raychaudhuri and Ivan Seskar for providing me the support and their wisdom, which was essential for completing this thesis. I am also grateful to Professors Richard P. Martin, and Predrag Spasojević for serving in my proposal and defense committees.

It has been my great pleasure to work together with many friends at WINLAB. I express my sincere thanks to Kishore Ramachandran, Sanjit Kaul, Sachin Ganu, Gayathri Chandrasekaran, Kemal Karakayali, Omer Ileri, Umut Akyol, Aliye Ozge Kaya, and Mete Rodoper. I am indebted to my sister Aysegul Ergin and my parents for their unbounded care and love throughout my life. I am also very thankful to my precious wife Nurcan for her understanding and support in all the years of my study.

Dedication

To My Father

Mr. Mehmet Osman Ergin

Table of Contents

Abstract ii

Acknowledgements iv

Dedication v

List of Tables ix

List of Figures x

1. Introduction 1

2. Background 5

2.1. WLAN Interference Management 5

2.1.1. Transceiver parameter optimization 6

2.1.2. Channel assignment 6

2.1.3. Association control and load balancing 6

2.2. Rate Adaptation 7

2.3. Scheduling in WLANs 8

3. Analysis of Interfering WLAN System Performance 9

3.1. Experiment Setup and Traffic Models 9

3.1.1. Experimental and Simulation Setup 9

3.1.2. Validation Experiment 11

3.1.3. Traffic model 12

3.2. Analysis of the System Performance 14

3.2.1. Multi-cell Network with Realistic Workload 14

3.2.2. The Effect of the Number of APs 16

3.2.3. The Effect of Traffic Workload, Dynamic Station Arrival, and Number of Stations 17

3.2.4. Discussion 19
3.3. TCP Analysis in Multi-Cell Networks 20
3.3.1. Gong and Marbach's Model for Multi-cell Networks 20
3.3.2. Verification via Experiments and Simulations 21
3.3.3. Discussion 22
3.4. Effect of AP Density on Multimedia over WLAN 24
3.4.1. Performance over Legacy IEEE 802.11a 25
3.4.2. Performance over IEEE 802.11e (WME) 26

4. $WiPhi$: Improving Performance with CW Adaptation 29
4.1. CW_{min} Adaptation Using Active AP Count 29
4.2. Implementation of CW_{min} Adaptation 30
4.3. Simulation Results 32
4.4. Discussion: Relationship to Collision-resilient Bit-rate Adaptation 33
4.5. Conclusions 34

5. $HeedNet$: Improving Performance with ISP Participation 36
5.1. Overview 36
5.2. Severity of Interference 38
5.2.1. Practical Densities and Interference 38
5.2.2. Design of the High-Density Residential Scenarios 39
5.2.3. Simulation Setup 41
5.2.4. Baseline Results 42
5.3. HeedNet Design and Algorithms 44
5.3.1. HeedNet Components and Operation 44
5.3.2. Assumptions and Notation 45
5.3.3. Selecting Potential Interferers 46
5.3.4. Determining Schedule Intervals 47
5.3.5. Ordering APs for Scheduling 48
5.3.6. Evaluation of Algorithms 50
5.4. Performance Evaluation 52

5.4.1. Recovering Lost Performance 53
5.4.2. Effects on Fairness 54
5.4.3. Delay Behavior with HeedNet 55
5.4.4. Buffering Requirements 56
5.5. Implementation and Deployment 56
5.5.1. Testbed and Design 57
5.5.2. Scenarios Tested and Results 59
5.6. Discussion and Limitations 61

6. Conclusions 63
6.1. Conclusion 63

References 65

Curriculum Vita 69

List of Tables

3.1. Attribute Summary for ORBIT Experiments 10
3.2. Attribute Summary for Simulations 11
3.3. Validation Experiment Results for Bianchi's Model, QualNet Simulator and ORBIT Testbed 12
3.4. Empirical VoIP Performance 25
3.5. Simulated VoIP Performance 26
5.1. Statistics from the apartment dataset 40
5.2. Parameter Summary for Simulations 42

List of Figures

3.1. ORBIT testbed experiment setup with configurable number of access points and up to 400 nodes 10

3.2. (a) WLAN application workload characterization. (b) ON-OFF model for emulating realistic WWW access [1]. 13

3.3. Dynamic client arrival patterns from traces of the 62^{th} IETF meeting [2] showing rapid changes in number of associated users. 14

3.4. System performance for the most general case: *multi-cell with many clients using realistic traffic and arrival patterns*. 15

3.5. Experiment with 320 stations and 12 APs showing association patterns and distribution to access points. 16

3.6. Investigating the effect of the number of APs to system performance. Everything else is kept the same as the previous experiment in Section 3.2.1. 17

3.7. Results from the ORBIT experiments investigating the effects of three different factors on system performance 18

3.8. (a) Empirical and (b) simulated collision-caused packet error rate (PER) with up to 16 clients and 2 APs. (c) Collision-PER predicted by Bianchi's Model. . . 22

3.9. Accuracy of the model relative to simulation results. All simulation results exhibited very small variance characteristics, hence errorbars are not shown here. 23

3.10. Collision-rate behavior of TCP under varying CW$_{min}$ values. 24

3.11. Simulated VoIP MOS results for varying intensities of multimedia traffic carried over one and four AP networks using IEEE 802.11e (WME). 27

3.12. Default value for AIFSN parameter of IEEE 802.11e *Best Effort* queue results in decreasing throughput for bulk data traffic, even in the absence of any multimedia traffic. 28

4.1. Simulations with 32 STAs showing potential gains from incorporating AP-Count-based CW adaptation into WLAN with inter-cell interference (in (b), throughput curves for "Optimal CW_{min}" and "Nearest Practical CW_{min}" almost overlap). 32
4.2. Empirical IEEE 802.11 bitrate distribution with the Samplerate algorithm. . . . 33
5.1. (a) Interfering residential APs (b) Less interference after ISP router schedules outgoing packets. 37
5.2. System capacity with increasing APs. The same workload causes more collisions thus reduced system throughput. 38
5.3. Residential WLAN observation experiment we conducted in Manhattan, NY. Hundreds of WLANs seen from a single location, with tens of them having the potential to strongly interfere with each other. 39
5.4. Residential setup used in simulation scenarios with 5-units in the small (SSA), 18-units in the medium (MSA), and 136-units in the large-sized apartment (LSA). One AP and two clients are placed randomly in 3D space of each unit and 35% of the units are assumed to have WLANs that are on the same channel. 41
5.5. Topology used for assessing the high-density residential scenarios. All WLANs are assumed to be served by a single ISP router with packets to and from ten different Internet servers. 42
5.6. Severity of High-Density Residential WLAN Interference. Errorbars indicate 95-percentiles of the resultsets. 43
5.7. Overview of HeedNet Operation with Functional Units 45
5.8. Evaluation of $pick\_interval()$ 51
5.9. Evaluation of $order\_APs()$ 52
5.10. HeedNet's Performance on High-Density Residential WLAN Interference Management. Errorbars indicate 95-percentiles of the resultsets. 53
5.11. HeedNet Under Microscope: Behavior of a TCP flow in Legacy (top) and HeedNet (bottom) system. 54
5.12. Distribution of delay between packet arrivals at the STA 55
5.13. Delay behavior for different file-sizes. 56
5.14. Buffering requirement on IER 57

5.15. Testbed Deployment for HeedNet 58
5.16. HeedNet implementation on the router using Click 59
5.17. Results from the scenarios tested on the ORBIT testbed. 60

Chapter 1

Introduction

The expanding ubiquity of Wi-Fi networks and the growing dependence on it in both enterprise and residential domains require a careful management of their deployments. Despite their demanding users and heavy workload, Wi-Fi networks for enterprise domains are technically more suitable for management in terms of device control and policy implementation. Unplanned high-density WLANs, such as residential Wi-Fi networks, however, are simply chaotic in deployment, capacity, coverage, and interference planning [3]. Lack of any coordination between neighboring wireless devices and the restrictions of many ISP contracts add to the management puzzle of the unplanned Wi-Fi deployments-at-large.

RF interference has been cited as the source for more than half of all residential Wi-Fi problems [4]. Mitigation of co-channel interference in residential domains is not straight-forward, since the current channel assignment and load balancing approaches for centrally managed WLANs [5–10] are not suitable for distributed deployment. Also, given the small number of available non-overlapping channels in the unlicensed spectrum and the rate at which these channels are utilized, these solutions at their best may only allow an even exposure of users to interference. On the other hand, consumer-grade wireless access points that are in the market today only have basic capabilities to test and select the quietest channel for operation during boot time [4]. Beyond this optimistic mechanism, researchers proposed state-of-the art distributed algorithms for dynamic channel hopping [11], distributed contention window adjustments [12], and transceiver parameter optimizations with CCA-thresholds [13] or transmit power/bit-rates [14]. However, they all necessitate changes in the software of the access point. Such changes, unfortunately, create a barrier in front of the wide-adoption of these interference management techniques.

In these unplanned high-density environments, little is known about the effects of inter-cell interference on IEEE 802.11 system performance. Detailed analytical and simulation models exist for the MAC protocol scalability [15,16] and experimental studies have characterized scalability under TCP and UDP workloads [17] in the single cell case. However, system scalability in the common *unplanned* multi-cell case remains largely unexplored. Multi-cell networks have been studied through measurement campaigns in real-world campus [18] or conference settings [19, 20] and recent measurements in a dense conference deployment have detected performance anomalies [20, 21], but the data does not allow a detailed analysis of root causes.

In this thesis, we present a systematic analysis of the effect of inter-cell interference in such unplanned, high density WLANs through detailed experiments and simulations. Our work complements previous real-world measurements through experimentation with over hundred IEEE 802.11 enabled nodes in a repeatable laboratory setting with controlled interference. Thus, it allows in-depth analysis through simulations and repeatable experiments, with precisely known configurations. The insights provided by our analysis allow us to enhance the existing IEEE 802.11 performance models to be used for understanding the performance of current high-density wireless deployments. Further, we propose two solutions that target the root-cause of the inefficiency we have observed from our analysis: *interference-driven collisions*. The first solution we propose, *WiPhi*, a new CW adaptation scheme, is a MAC layer approach with software ugrade requirements on the AP. For those deployments where changes to AP are prohibitively costly, we propose *HeedNet*, an ISP-based scheduling method that can control the interference at customer premises. Both solutions can recover most of the losses we demonstrate in this thesis. In summary, our contributions with this dissertation include:

- Analyzing system performance using a realistic TCP dominated workload in unplanned multi-cell WLANs by conducting experiments on ORBIT testbed [22] as well as QualNet simulator [23]. Results show that a single-cell network remains remarkably robust even with 125+ clients; the collision rate remains low. This extends Choi et.al.'s empirical results [17] for 16 clients to a much larger network, with realistic client association patterns, and bursty traffic mixes. We also show that, in an unplanned multi-cell network, however, the collision rate increases significantly.

- Providing novel insights into the behavior of TCP in multi-cell WLANs. Due to TCP flow control, the number of backlogged stations equals twice the number of active access points, meaning that network efficiency is determined by *the number of interfering access points*, not *the number of clients*. In addition, we show that TCP can not regulate the flows in the IEEE 802.11 network for optimal system operating point (i.e. max. throughput) across different contention window settings.

- Quantifying the effect of inter-cell interference on multimedia traffic and on throughput loss due to inefficient rate adaptation. Even with Wireless Multimedia Extensions based on the IEEE 802.11e standard [24], VoIP users can still experience substantial performance degradation in unplanned deployments. This deterioration starts to occur even when the number of interfering APs is *relatively small* (three). Video streaming in the network makes the system performance worse for VoIP users.

- Identifying a practical distributed interference mitigation technique ($WiPhi$): *contention window adaptation based on the number of active access points*, not the number of clients in the network. We also show that an additional 20% gain would be possible with collision-resilient rate adaptation.

- Providing realistic residential high-density WLAN scenarios for performance evaluations with medium and large-sized apartment buildings, based on recent metro-area apartment surveys and channel usage statistics from millions of APs.

- Design of an alternative scheduling solution requiring no MAC layer changes and AP modifications ($HeedNet$). Algorithms we present can efficiently determine the WLANs to schedule together, find the appropriate scheduling interval, and order the APs for scheduling in a feasible way. Using simulation experiments, we show that $HeedNet$ could easily be implemented as an extension to the ISP edge-routers serving interfering APs and can provide more than 2X improvement in system capacity while maintaining a better per-WLAN fairness than legacy.

- Demonstrating that our ISP-based scheduling solution is performing better than legacy in terms of its delay characteristics, since it provides lower inter-packet delay distribution, faster download completion, and more deterministic packet delivery to the client. Also, we show that $HeedNet$ requires very modest amount of buffer space on the ISP router.

- Implementation of $HeedNet$ using Click modular router on a general-purpose server. Our deployment using ORBIT infrastructure with off-the-shelf APs and the implementation proves that the gains from $HeedNet$ are practical and a potential ISP deployment is feasible.

- Showing $HeedNet$ benefits to non-scheduled WLAN traffic on our testbed deployment. Non-scheduled traffic, such as VoIP, enjoys reduced collision-rate environment $HeedNet$ creates, improving its delay and loss behavior.

Our analysis provide researchers evaluating high density WLAN deployments an opportunity to use existing IEEE 802.11 performance models for their system capacity assessments. Using extensive simulations and implementation, we show that our solutions $WiPhi$, and $HeedNet$ can recover most of the capacity loss due to interference-related collision losses.

Chapter 2

Background

Related work, most relevant to our study, that models the interaction between TCP and IEEE 802.11 is presented in [17,25,26]. Choi et al. [17] and Bruno et al. [25] show that for both downlink and uplink traffic, cumulative TCP throughput does not degrade in single-cell WLANs. In [26], a discrete time model explaining the interaction between TCP and IEEE 802.11 is presented for a network topology consisting of multiple source destination pairs along with an infrastructure-like network. In this thesis, we used their model to extend it for multi-cell infrastructure networks. In addition, the authors of [27], study TCP fairness in the presence of simultaneous uplink and downlink traffic and observed significant unfairness among TCP flows. They model the interaction between TCP and IEEE 802.11 but do not consider the effect of MAC congestion. We differ from all these studies since we study the interaction between TCP and IEEE 802.11 for high-density multi-cell WLANs.

Several studies in the past have also proposed tuning the contention window (CW) to maximize utilization [28–30]. In [28], the authors determine that finding a balance between the bandwidth loss associated (i) with collisions and (ii) with the time spent by the nodes backing off (idle period) is possible. Heusse et al. [29] use an AIMD algorithm to tune the CW so as to maintain the idle period at a desirable level. Hu et al. [30] analyze the ability of IEEE 802.11e EDCA [24] to maximize bandwidth utilization and provide service differentiation. However, all these studies focus on the single-cell WLAN and address the situation of multiple competing clients. We believe that we are the first to apply CW_{min} adaptation to mitigate interference from adjacent BSSs in unplanned deployments.

2.1 WLAN Interference Management

There are a number of complementary techniques to address interference-based performance degradation in IEEE 802.11 WLANs. They can be grouped into the following broad categories:

2.1.1 Transceiver parameter optimization

This category includes transmit power, carrier sense threshold and receiver sensitivity adjustments. In [31], the authors propose tuning the carrier sense threshold at the receiver to mitigate interference effects and a dynamic power management technique to reduce interference in unplanned deployments is proposed in [3]. In [13], a combination of receiver sensitivity and clear-channel-assessment adaptation is proposed. However, the authors themselves point out the suboptimal behavior of their approaches in uncoordinated environments. More recently, [14] proposes a distributed algorithm to jointly adjust transmit power and IEEE 802.11 bit-rate to reduce interference while not sacrificing performance.

2.1.2 Channel assignment

This category includes static and dynamic channel assignment techniques to mitigate interference. In [11], the authors show that static channel assignment techniques cause unfairness in unplanned deployments and then describe a decentralized channel hopping scheme that improves fairness by distributing interference evenly among neighboring BSSs. We argue that this solution may not fully be able to mitigate contention in environments where the number of interfering APs are higher than the number of orthogonal channels. The CFAssign-RaC algorithm presented in [5] jointly address the issues of channel assignment and load balancing in centrally administered WLANs. Given the uncoordinated nature of the deployments we consider, this solution is not directly applicable here.

2.1.3 Association control and load balancing

These approaches balance client load across a set of APs [5, 6, 32] by changing the point of association of the clients. For that reason, they have an inherent assumption of requiring coordination and orthogonality of channels across APs. Recently, [21] propose an association management solution to prevent WLANs from accepting more clients than they can serve efficiently. However, their solution mandates that clients incur delays on the order of minutes.

Some WLAN equipment vendors [8–10] claim that their lightweight APs and the WLAN controllers coordinating these APs support high density deployments through load balancing and other algorithms. However, neither the underlying methodology nor success of these proprietary solutions in heterogeneous, unplanned environments is known.

2.2 Rate Adaptation

Existing rate adaptation studies can be grouped into (i) frame-error based adaptation; (ii) SNR based adaptation; and (iii) throughput based adaptation. Auto Rate Fallback (ARF) [33] and Adaptive Auto Rate Fallback (AARF) [34] mechanisms increase transmission rate after a fixed or dynamic number of successful transmissions at a given rate and switches back to a lower rate after one or two consecutive failures. ONOE [35], a frame-error based algorithm used in the MADWIFI driver for Atheros-based wireless NICs aims at selecting the highest bit-rate with less than 50% frame loss rate. Adaptive Multi-Rate Retry (AMRR) [34], a modification of ONOE, adaptively raises the threshold for rate increases to prevent frequent attempts at bit rates higher than the optimal one in an SNR-limited channel.

Receiver Based Auto Rate (RBAR) [36] is another published rate adaptation algorithm whose goal is to optimize the application throughput. The RBAR algorithm mandates the use of the RTS/CTS mechanism: a pair of Request To Send/Clear To Send control frames are exchanged between the source and the destination nodes prior to the start of each data transmission. The receiver of the RTS frame calculates the transmission rate to be used by the upcoming data frame transmission based on the Signal To Noise Ratio (SNR) of the received RTS frame and on a set of SNR thresholds calculated with an apriori wireless channel model.

The SampleRate [37] algorithm selects the rate that minimizes mean packet transmission time. Initially, the loss-less packet transmission times are calculated for each bit rate and an initial rate is chosen (36Mbps). Hereafter, for each successfully sent packet, the transmission time is updated (using an exponentially weighted moving average (EWMA)) based on the number of retransmissions, packet length and protocol timing overheads. The algorithm also periodically attempts transmission at bitrates whose loss-less transmission time is lower than the measured time on the current rate. If these sample transmissions indeed show lower mean transmission time, the algorithm switches the rate.

The Opportunistic Auto Rate (OAR) protocol [38], which can be layered on top of any of the above rate adaptation mechanisms can optimize individual, as well as network throughput, by sending multiple back-to-back frames under favorable channel conditions.

2.3 Scheduling in WLANs

Scheduling based channel access has been extensively studied in cellular networks [39]. In the 802.11 context, major research has been done in distributed scheduling techniques for multi-hop or ad-hoc networks [40] since centralization of data, though recognized as providing more control, is harder to implement, and therefore less common.

Centralized scheduling in enterprise WLANs is studied in CENTAUR [41] in which a central control element makes scheduling decisions when individual frames should be transmitted by APs that are part of the enterprise. CENTAUR shows that a selective amount of data-path centralization is useful in enterprise WLANs in directly mitigating performance loss due to downlink hidden and exposed terminal scenarios. Further, such a mechanism can indirectly help improve the performance of the entire WLAN environment.

Meru Networks [42] has proposed cellular-like coordination of various APs and scheduling mechanisms to provide a certain degree of deterministic channel access in enterprise WLANs. Mechanisms like TXOP in 802.11e [43] and packet aggregation in 802.11n also provide uninterrupted channel access to wireless transmitters for extended periods of time. However such mechanisms are orthogonal ways for implementing epoch based scheduling.

Chapter 3

Analysis of Interfering WLAN System Performance

In this chapter we provide a systematic analysis of the effect of inter-cell interference in unplanned, high density WLANs through detailed experiments and simulations. The next section explains our evaluation methodology and the details of the realistic traffic workload we have used followed by our characterization of the effect of node density (both client and AP density), traffic variability, and client arrival pattern, on high-density WLAN system performance. Then, performance evaluations are provided with large-scale realistic experiments on ORBIT testbed [22] as well as evaluations of multimedia performance (esp. VoIP) through experiments and simulations.

3.1 Experiment Setup and Traffic Models

We leverage the publicly accessible ORBIT testbed [22] to carry out systematic and controlled experiments. In our evaluations, we use the network topology shown in Figure 3.1. The main components of this integrated wired/wireless IP network are wired nodes hosting application servers, wireless access points (AP) and stations (STA). We focus on application behavior in the wireless access segment, which consists of multiple, interfering basic service sets (BSS) on the same channel, in close proximity. All nodes remain in communication range emulating future very high-density deployments. Evaluating the effect of hidden nodes is beyond the scope of this thesis. Pairwise SNR profile of all nodes in our testbed allows communication using IEEE 802.11a at 54 Mbits/s rate with less that 1% packet error rate.

3.1.1 Experimental and Simulation Setup

Table 3.1 provides a list of the parameters we used on ORBIT testbed. To ensure that our results are representative of real-world behavior, we first carried out calibration tests comparing throughput of ORBIT machines configured as APs (that use the MadWifi driver [44]) with

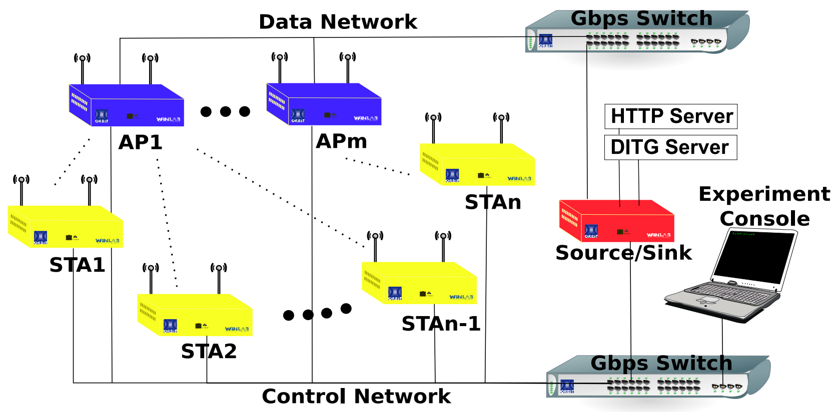


Figure 3.1: ORBIT testbed experiment setup with configurable number of access points and up to 400 nodes

commercial Cisco (1200 series) and D-Link APs. We did not observe a significant difference (less than 5%). We also configured the APs as bridges and used traffic sniffers (via tcpdump) on both the wired and wireless segments. To allow other researchers to replicate, and build on our experiments in the ORBIT environment, we have published relevant traffic scripts, tools, and frame dump traces through the CRAWDAD archive [2] as *rutgers\ap_density* traceset [45].

For our simulations, we chose QualNet [23] (the commercial successor of GloMoSim) due to its accurate physical layer interference model, which can affect higher layer performance comparisons [46]. In particular, QualNet's SINR calculation taking into account the cumulative interference power from all concurrent senders is very important for measuring the effect of MAC collisions in our high-density simulations. Note that the default ns-2 model may underestimate collisions, since it only keeps track of the strongest interferer, not the sum of all interference signals.

Table 3.1: Attribute Summary for ORBIT Experiments

Attribute	**Value**
Radio Nodes	1GHz VIA C3 Processor, 512MB RAM, 20GB HDD
Wireless Interfaces	2 X Atheros AR5212 based mini-PCI 802.11a/g
Wireless Output Power	18 dBm
PHY/MAC/Freq. Used	IEEE 802.11a / Operating at UNII Band Channel 52
PHY Link Speed (Fixed)	up to 54Mbits/s
MAC Payload Size	1300 bytes
MAC retries	10
O/S Used	Linux 2.6.18
Driver Software	MadWifi svn.21XX [44]

Table 3.2: Attribute Summary for Simulations

Attribute	Value
PHY	5.2GHz, Two Ray Ground, Tx. power of 18 dBm, Rx. Sensitivity of -78dBm (at 36Mbps)
MAC	Basic access (RTS off), Variable CW_{min} (mostly 15), CW_{max} of 1023, 16 μs SIFS, 34 μs DIFS, 9 μs Slot time, 28 bytes MAC header, 14 bytes ACK frame size, 1300 bytes MAC payload, 10 retries
NET	IPv4, IP Queue size of 75000 bytes
TCP	NewReno with RFC 1323, Max. segment size of 1300 bytes, Send/Rcv buffer size of 110KB, Delayed ACK disabled

Before running our simulations, we made sure that two bugs in the particular version of the simulator: (i) losing slot synchronization when resuming backoff (already described in [47]), and (ii) improper resetting of CW values for IEEE 802.11e EDCA [24] access queues, were fixed. The parameters that we used in our simulations are outlined in Table 3.2[1].

Metrics: For performance evaluation, we choose to use application-specific metrics. Thus, for the performance of TCP-based applications, we focus on system throughput and throughput fairness (Jain's fairness index [48]). For multimedia applications such as VoIP, we utilize both quantitative metrics such as application-level packet drop rate, latency and jitter as well as a standard subjective quality metric, namely mean opinion score (MOS)[2].

The rationale behind conducting our experiments in a controlled laboratory setting (such as ORBIT) rather than a real deployed WLAN is as follows. First, it allows detailed instrumentation to understand MAC-level behavior without the use of large numbers of sniffers. Second, experiments are repeatable that is they are not dependent on time-varying shadowing and interference patterns. These allow both easier investigation of root causes and directly comparing alternative solutions. Finally, the high density placement of 400 nodes allows us to experiment with densities that may be expected in future years rather than focusing solely on today's system performance.

3.1.2 Validation Experiment

Throughout the dissertation, we will use collision-dominated packet error rate (PER) and system throughput measurements obtained from Bianchi's IEEE 802.11 model that assumes that all stations are backlogged [15], the QualNet simulator [23], and the ORBIT wireless testbed [22],

[1]Note that we used a modified MAC retry value of 10 to match those used in our ORBIT experiments.

[2]MOS is a subjective score (ranging from 0 to 5) used to evaluate voice quality as perceived by an average user of the system. Details can be found in ITU G.107 and G.113.

Table 3.3: Validation Experiment Results for Bianchi's Model, QualNet Simulator and ORBIT Testbed

	Avg. PER (%)	**Avg. Throughput (Mbits/s)**
Bianchi's Model	10.17	23.14
QualNet Simulator	11.02	22.28
ORBIT Testbed	9.86	24.67

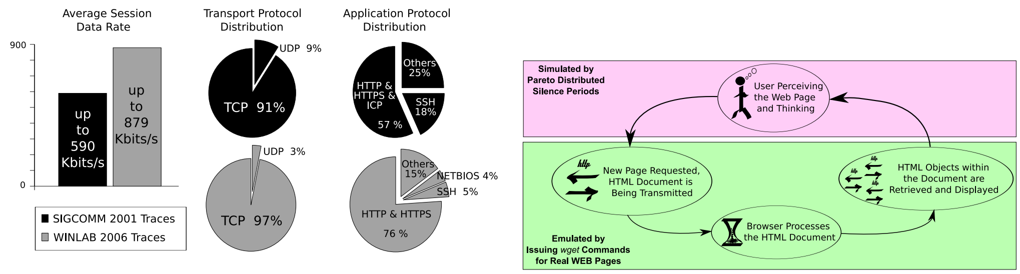
to compare the performance of the wireless networks of interest[3]. To be able to validate such a comparison, we conduct a saturation data transfer experiment on an IEEE 802.11a network of two active nodes using 36 Mbits/s fixed rate, sending 1500-byte UDP packets, for a duration of five minutes. Average PER and throughput results obtained from MATLAB simulations of Bianchi's model, the QualNet simulator and the ORBIT testbed for this configuration are reported in Table 3.3. From this and other similar experiments, for the configurations tested, we have repeatedly observed that results from all three methods are very close, validating their potential comparability.

3.1.3 Traffic model

Workloads: Since inter-cell interference patterns are affected by end-user workloads, we designed a synthetic office workload in addition to bulk TCP-only workloads. The office workload is based on several hours of sniffer traces obtained in our academic office/lab environment from a single access point serving up to 50 students and faculty. These measurements indicate that 97% of packets use the TCP protocol and about 75% of traffic is generated by web traffic, as illustrated in Figure 3.2(a). In the figure, all percentages are based on the number of bytes communicated in the WLAN and only application protocols with $> 2\%$ contribution are individually referred. These measurements are reasonably consistent with, except for a 20% increase in web traffic, with an earlier analysis of SIGCOMM 2001 conference traces covering 4 APs and 195 stations [19], which is also shown in the figure.

Thus, 75% of the synthetic workload consists of bursty web traffic, following the self-similar ON-OFF traffic model described in [1]. We directly emulate the HTML transfer, browser processing, and HTML object retrieval phases using the HTTP 1.1 compliant *GNU*

[3]Note that whenever Bianchi's saturation model is used for comparisons, corresponding experiments on simulator and the testbed will incur bulk transfer workloads to satisfy backlogged-station requirements of the model.



(a) (b)

Figure 3.2: (a) WLAN application workload characterization. (b) ON-OFF model for emulating realistic WWW access [1].

wget page retrieval tool [49] to access a local webserver serving web pages and objects obtained from an academic web server. The model we used for this emulation is illustrated in Figure 3.2(b). The user's thinking time X (inactive time) between page accesses follows a Pareto distribution:

$$Pr(X > x) = (\frac{x}{x_m})^{-k}$$

with shape parameter $k = 1.5$ and lower bound $x_m = 1$, as suggested in [1]. We concentrate on TCP downlink traffic, since it represents typical access point usage for web browsing. Also, earlier results [25] showed that the direction of TCP traffic is not significant. This is mostly due to the equal frame flow-rate requirement of TCP in data segment and ACK directions.

The remaining share of the workload comprises a mix of VoIP traffic (over UDP/IP) using the G.711 codec with H.323 signalling (3% of overall volume), and TCP packet transfers with exponentially distributed interarrival times (21% of the overall volume on average) as background traffic. These flows are emulated through the D-ITG traffic generator v.2.4.3 [50]. In the experiments, each station is assigned a traffic generation profile to satisfy the workload distribution outlined above, and keeps this profile until the end of the experiment.

User Arrival Pattern: Another factor that might potentially affect system performance is client association dynamics. To measure performance in a more realistic manner, we extracted the user arrival patterns from WLAN traces of the 62^{nd} IETF meeting [20]. In particular, we use the arrival pattern of the users returning from lunch between 12:30pm to 1:00pm, as illustrated in Figure 3.3. The IETF WLAN comprised over 150 APs and more than 700 users. Note the significant variance in user associations, for example at 43 minutes into the trace, the number of users on channel 11 quickly increases from approximately 50 to over 250 within a two minute window.

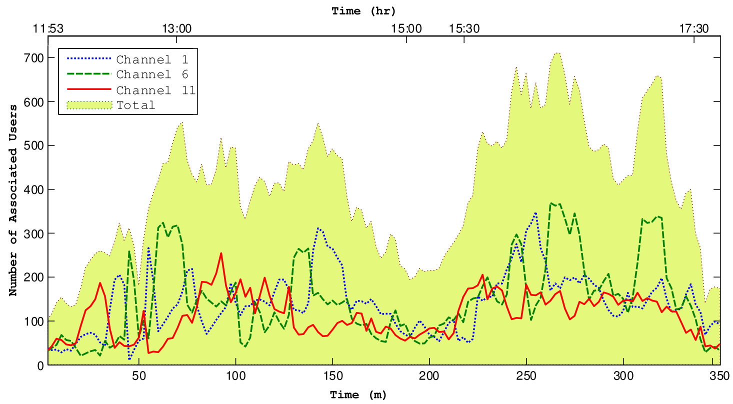


Figure 3.3: Dynamic client arrival patterns from traces of the 62^{th} IETF meeting [2] showing rapid changes in number of associated users.

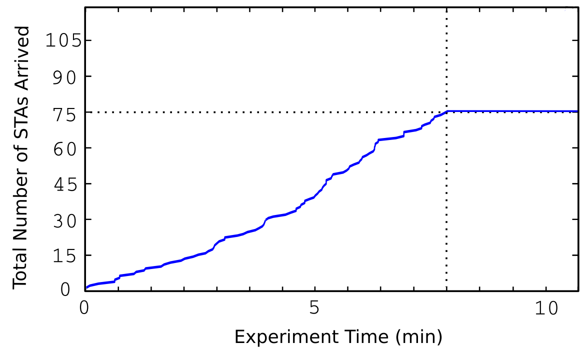
3.2 Analysis of the System Performance

In this section, we study the system performance of a multi-cell wireless network deployment, where the cells interfere with each other. We systematically examine the effect of access point density, traffic variability, user arrival pattern and station density to understand root causes of performance problems. We begin with an experiment that emulates the characteristics of short-term conference deployments [20, 21], with multiple APs, a web-dominated traffic workload, and a dynamic station arrival pattern. Emulating such a scenario in a controlled environment will allow us to isolate the effect of these factors on network throughput.

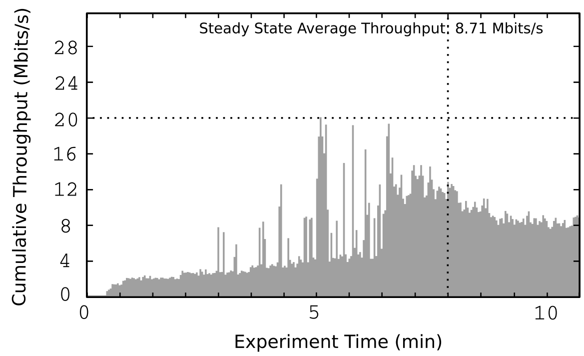
3.2.1 Multi-cell Network with Realistic Workload

This 11 minute experiment comprises four APs and 75 STAs, selected randomly from the 400-node ORBIT main radio grid. Once APs are operational on UNII 5GHz Band Channel 52, using IEEE 802.11a, stations start to associate with the network by following the dynamic client arrival pattern described in Section 3.1.3. This arrival pattern is illustrated in Figure 3.4(a). All APs use the same SSID, thus STAs select the AP with the highest Received Signal Strength Indication (RSSI) at their position. In this experiment, the four APs have 32, 13, 13, and 17 stations associated, respectively.

Figure 3.4(b) summarizes the system performance in terms of the cumulative system throughput for the network. We attribute the throughput spikes, especially with about 30 STAs, to our



(a) Station (STA) arrival dynamics



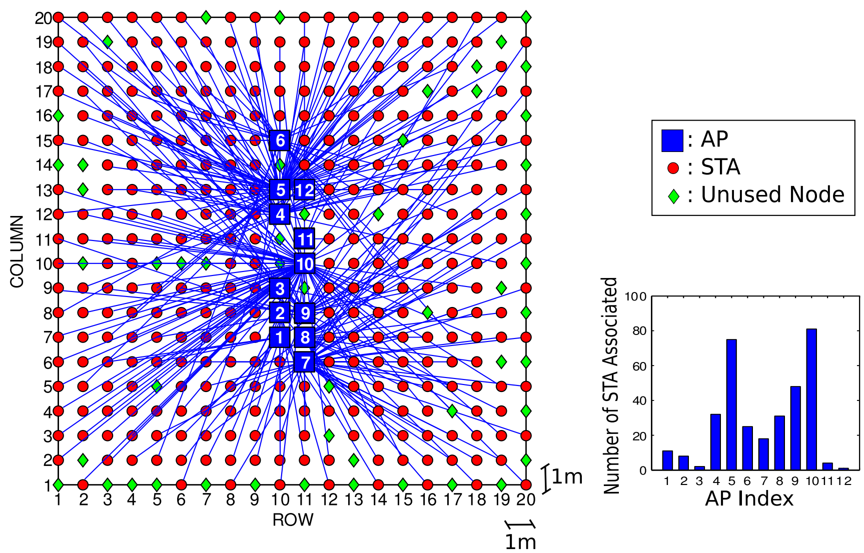
(b) ORBIT experiment for 4 APs, 75 STAs

Figure 3.4: System performance for the most general case: *multi-cell with many clients using realistic traffic and arrival patterns*.

user model for web browsing traffic (i.e., thinking/perceiving phase vs. fetching/downloading phase). As more STAs arrive, communication demands increase, and an averaging effect is observed in the overall system throughput. Our observations from this experiment are:

- The steady state average system throughput, **8.71 Mbits/s** (calculated over the last three minutes of the experiment after all 75 STAs arrived) is below the capacity of a 54Mbits/s IEEE 802.11a network [51]. In fact, a one-AP/one-STA baseline experiment using TCP bulk data transfer, we conducted, yielded **24.02 Mbits/s** average steady state throughput in the same experiment environment.

- Distribution of clients across access points is uneven. One of the APs in the experiment serves more than twice the number of STAs of another AP. To investigate this further, we conducted an experiment with 320 stations (STA) and 12 co-located (to the extend ORBIT testbed's fixed node placement allows) APs. We observe a similar uneven distribution as shown in Figure 3.5. At the stations, we have logged RSSI measurements for



(a) STA and AP Distributions

Figure 3.5: Experiment with 320 stations and 12 APs showing association patterns and distribution to access points.

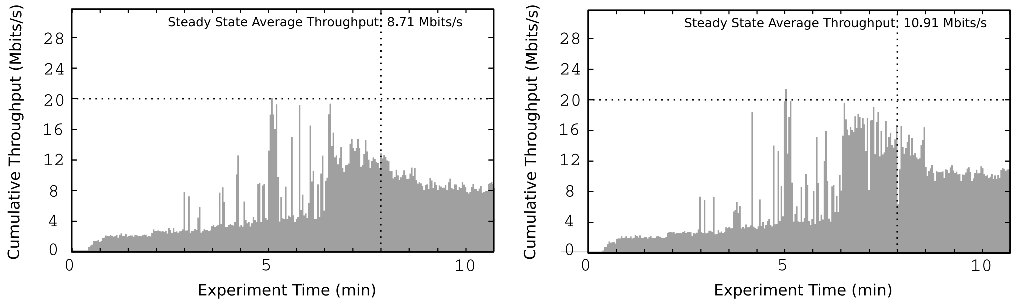
the co-located APs and found out that they differed likely due to a combination of multipath, antenna gains, RF frontend dynamics, connectors, and cabling. In summary, even a high-density distribution of client positions with line-of-sight propagation does not necessarily lead to an even distribution of associations over co-located access points. This observation motivates the need for association control techniques in multi-channel/multi-AP WLAN installations (e.g., [21]) aiming to create even load distribution within the network.

To understand the root cause of threefold throughput change, we will study the effect of access point density, traffic workload, dynamic user arrivals, and station density on this result.

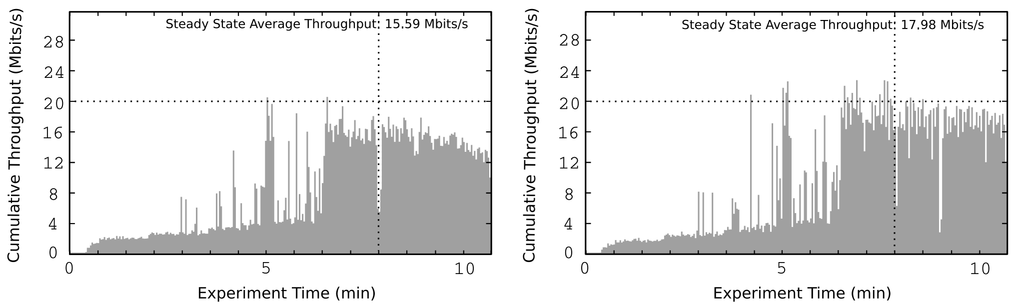
3.2.2 The Effect of the Number of APs

To measure the effect of *the number of APs*, we repeat the previous experiment with one, two, and three access points, while keeping the total number of STAs and their offered load the same. Results from this experiment are illustrated in Figure 3.6, shown together with the prior four AP experiment result to facilitate comparisons. Our observations follow:

- All four experiments show two distinct phases. During the first phase, lasting until about 75 STAs are associated, system behavior is comparable across all four cases. We attribute



(a) ORBIT Experiment for 4 AP, 75 STAs (b) ORBIT Experiment for 3 AP, 75 STAs



(c) ORBIT Experiment for 2 AP, 75 STAs (d) ORBIT Experiment for 1 AP, 75 STAs

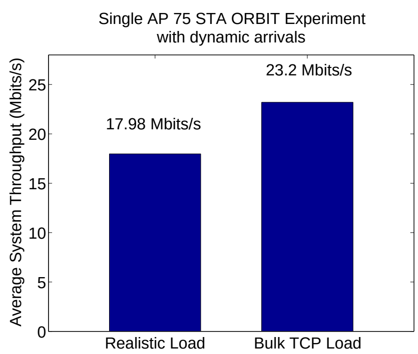
Figure 3.6: Investigating the effect of the number of APs to system performance. Everything else is kept the same as the previous experiment in Section 3.2.1.

this to the very spiky offered load staying mostly below the capacity provided by the system—throughput is *offered-load limited*.

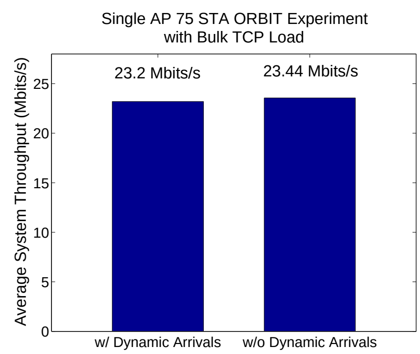
- In the following *capacity-limited* steady state phase (after all 75 STAs join), configurations with fewer APs result in a significant increase in throughput. In particular, using three APs increases the average throughput by 25.2% compared to four APs. Similarly, reduction to two APs increases the average throughput an additional 53.8% compared to three APs. Finally, a single AP network achieves an average system throughput of **17.98 Mbits/s**, which is about a **106.4% performance improvement** compared to the four AP network. Despite this improvement, the average system performance is still below our 24.02 Mbits/s baseline result.

3.2.3 The Effect of Traffic Workload, Dynamic Station Arrival, and Number of Stations

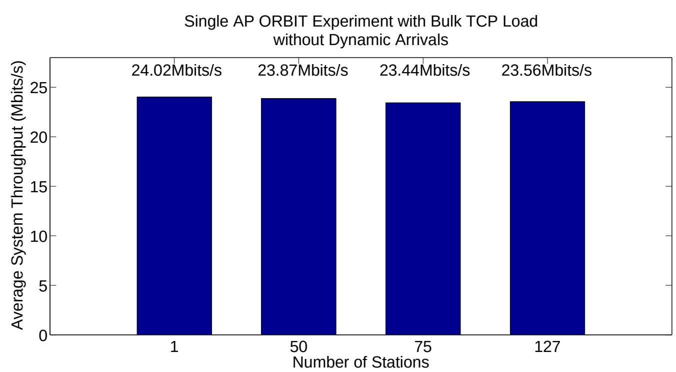
To analyze the remaining difference, we first repeat the previous single access point experiment (using short-lived TCP sessions generated according to a web-dominated workload) with a bulk



(a) Effect of the Traffic Workload



(b) Effect of the Arrival Pattern



(c) Effect of the number of STAs

Figure 3.7: Results from the ORBIT experiments investigating the effects of three different factors on system performance

TCP transfer workload. The bulk TCP workload is constructed by initiating a TCP session from each STA upon arrival and downloading a large file (large enough to not complete during the experiment duration) from a server on the wired network. Results of this experiment are illustrated in Figure 3.7(a). We observe that the bulk TCP workload results in an average system throughput **23.2 Mbits/s**, an increase of 29% compared with the web-dominated workload. We believe that the main cause for this throughput difference is TCP's rate control mechanism not adjusting quickly enough to the optimal TCP congestion window size when short lived TCP sessions are dominant in the network.

While the throughput with bulk TCP is close to the baseline experiment, we will also show the effect of dynamic station arrivals for the purpose of completeness. Figure 3.7(b) presents results from the 75 STA experiment with bulk TCP workload and all stations joining the network simultaneously (i.e., without the dynamic arrival pattern). In steady state, only a negligible difference of 0.2 Mbits/s can be observed between the two experiments. We also observed high throughput fairness at steady-state for both arrival patterns. Jain's fairness index [48] is 0.96 with realistic arrivals and 0.94 otherwise.

We have also investigated the STA association performance, since it is the other major system functionality likely to be affected by the type of the arrival pattern being used. In the dynamic arrival case, new clients come into an already loaded WLAN, and associations are frequently delayed. We observed that with dynamic arrivals, about 40% of the STA associations take e 270ms, about 50% of them take 5.5s and a further 5% take up to 17s. In the case where all STAs appear at the same time, we observed that 80% of the associations complete within 270ms while the remaining 20% takes up to 5.5 seconds. This stepping behavior can be explained by association disruptions, whereupon the driver enters an active scan cycle of all 23 IEEE 802.11a/g channels. Scanning takes approximately five seconds in the Madwifi [44] driver implementation for Linux.

The final minor difference in throughput is due to the number of stations. Figure 3.7(c) shows the same experiment repeated with 1, 50, and 127 stations[4]. Results show little dependence on the number of stations.

From the experiments we have conducted so far, it is empirically observed that system performance has the strongest dependence on the **number of interfering APs**. We will continue with the investigation of this dependence in the following sections.

3.2.4 Discussion

The foregoing experiments show that (i) a single-AP network performs efficiently under TCP workload irrespective of the number of stations it serves (up to 127 stations in our setup) and that (ii) a multi-AP network serving the same number of clients on the same channel leads to significant throughput degradation.

As a byproduct, secondary effects that have been observed in practice such as inefficient bit-rate adaptation [20] do not manifest themselves in *single-cell deployments* with such traffic characteristics no matter how many users actively use the system.

These results show that congested WLAN systems can not be fully understood through traditional MAC-layer analysis. Models from neither Bianchi [15] nor Kumar et.al. [16] explain these results. According to Bianchi's model the efficiency of a WLAN depends primarily on

[4]127 STA limitation comes from the particular Madwifi driver version we have used on our APs.

the number of active stations, regardless of their role as access points or clients.

The single-AP result confirms more recent theoretical and experimental work with a smaller network setup [17, 25] which have suggested that TCP flow control, when used in a single cell network, operates the network efficiently and maintains robust system throughput regardless of the number of STAs. The multi-AP result, to our knowledge, has not yet been reported and we analyze it further in the following section.

3.3 TCP Analysis in Multi-Cell Networks

The reduction in cumulative throughput for multi-cell networks raises the following inter-related questions:

- why is TCP Reno over WLAN robust against intra-cell congestion but not against inter-cell congestion and interference?
- does TCP Reno adjust the flow rate to minimize collisions?

In this section, we will answer the above questions by identifying the applicability of Gong and Marbach's TCP [26] model to multi-cell networks, validate it through experiments and simulation, and finalize by discussing TCP flow control's ability to identify the optimal operating point.

3.3.1 Gong and Marbach's Model for Multi-cell Networks

According to Gong and Marbach's model [26] of TCP, in a single BSS case, on average, two stations will be backlogged, irrespective of the number of clients. Also, if n additional flows, in the form of Independent Basic Service Set (IBSS, a.k.a. an ad hoc network) are added as interferers, the expected number of active (i.e., backlogged) nodes in the network would be $2(1+n)$.

While Gong and Marbach do not comment on the multi-AP case, all assumptions made for IBSS flows also hold for BSS networks. Following the same steps, one can therefore also derive the following proposition:

Consider multiple IEEE 802.11 infrastructure networks (BSS) with the following two characteristics: (1) *Each BSS consists of at least one station and a single AP. All BSS are within* *transmission range of each other.* (2) *There is a single TCP connection per client and applications using TCP connections always have data to send. For a network topology consisting of* i *BSSs, in steady state, the expected number of backlogged nodes then equals* $2i$.

3.3.2 Verification via Experiments and Simulations

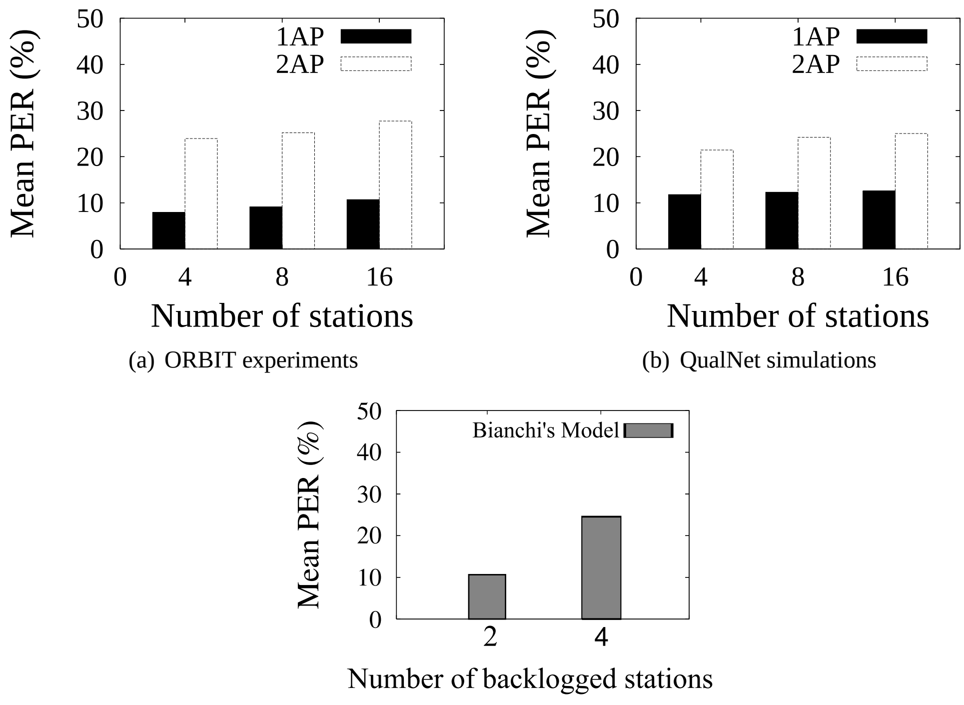
To verify the proposition in Section 3.3.1, we study the collision rate observed in the ORBIT experiments[5], where we use bulk TCP workload on six different network configurations (one and two APs, four, eight and sixteen STAs). Using Bianchi's model, we can then calculate the collision rates for the number of backlogged stations predicted by Gong and Marbach's model and compare it with the measurement result. We use this indirect approach, since MAC queues are maintained in hardware and we cannot directly determine whether a station is backlogged.

We further approximate the collision rate with the overall packet error rate (PER) from these experiments, since WLAN devices cannot distinguish collisions from other transmission errors and thus do not allow direct collision measurements. This approximation is accurate, since we operate in a high SNR environment where frame-errors due to pure channel bit-errors are negligible ($<$1% in our tests). As additional validation, we also simulate the same configuration in QualNet where we can extract the exact collision rate.

Figures 3.8(a) and 3.8(b) show the mean collision rate as the number of STAs is increased for one and two AP networks in the ORBIT testbed experiments and simulations, respectively. In both cases, we observe that PER due to collisions is marginally affected by an increase in the number of stations but is significantly affected by an increase in the number of APs— an increase from 1 to 2 APs more than doubles the average PER from 11% to nearly 28%. Near identical results from the simulations and the ORBIT experiments also indicate that the collision rate approximation we used was reasonable.

The empirical collision rate with a single AP (for 4, 8, and 16 STAs) matches the PER value predicted by Bianchi's saturation model [15] with two backlogged stations. Similarly, the empirical collision rate with two APs (for 4, 8, and 16 STAs) matches the PER value predicted by Bianchi's model with four backlogged stations. Predictions of Bianchi's model, obtained from MATLAB simulations, for the same experiment configuration are provided in Figure 3.8(c). To further verify the accuracy of this model, we conduct additional simulations with four and

[5] In these experiments, we do not consider dynamic STA arrivals and bit-rate adaptation in order to focus on MAC contention in a baseline scenario



(a) ORBIT experiments

(b) QualNet simulations

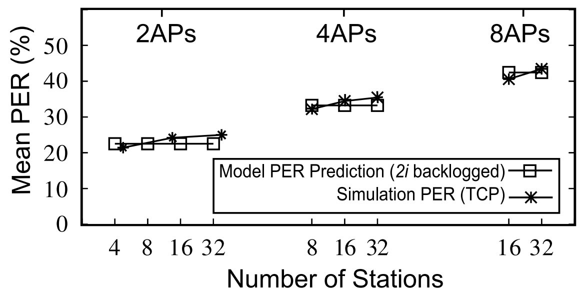
(c) Bianchi’s Model

Figure 3.8: (a) Empirical and (b) simulated collision-caused packet error rate (PER) with up to 16 clients and 2 APs. (c) Collision-PER predicted by Bianchi’s Model.

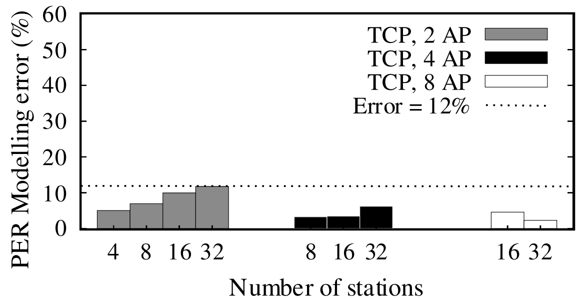
eight APs connecting up to 32 STAs and report the modeling error. Observed PER from model predictions and simulations closely follow each other. The percentage modeling error is illustrated in Figure 3.9(b)—the worst case error is only 11.76%. Moreover, modeling error reduces with an increasing number of interfering APs. We also observed that these simulation results exhibited negligible variance (a maximum PER variance of 0.12) across ten runs with different random seeds. These results verify that each active cell increases the total number of backlogged stations in the network by *two*, as stated by the proposition in Section 3.3.1. The network efficiency in a TCP-dominated network, for this reason, is primarily a function of the number of interfering APs.

3.3.3 Discussion

Incidentally, an IEEE 802.11a network running with two backlogged stations maximizes throughput according to Bianchi’s model. Fewer backlogged stations lead to too much idle time, while more backlogged stations lead to too much collision overhead. Thus, one may ask whether



(a) PER (multiple APs)



(b) PER modeling error

Figure 3.9: Accuracy of the model relative to simulation results. All simulation results exhibited very small variance characteristics, hence errorbars are not shown here.

TCP's flow control algorithm, designed for managing congestion in the Internet, can also control the flow rate to maximize throughput in congested wireless networks? And if so, why does it not achieve this under co-channel interference from other access points?

To address these questions, we conducted simulations in which we change the MAC CW_{min} parameter, while keeping the number of stations constant (32) and using bulk TCP transfers to saturate the channel. If TCP can identify the optimal operating point to maximize throughput in single AP networks, it should respond to the changed MAC CW_{min} settings with a corresponding change in the number of backlogged stations. Recall that to maximize 802.11 network throughput, the number of backlogged stations has to increase with an increase in MAC CW_{min}, to balance collisions against idle overhead [15].

Figure 3.10 shows the MAC PER under TCP for the single AP case (with 32 STA) with increasing CW_{min}. It also shows the expected packet error rate when only two stations are backlogged (i.e., no TCP adjustment) labeled as "analytical PER prediction" and an optimal PER curve, which corresponds to the PER at which the cumulative MAC throughput is maximized. Note that the observed (measured) MAC PER curve approaches the optimal value for

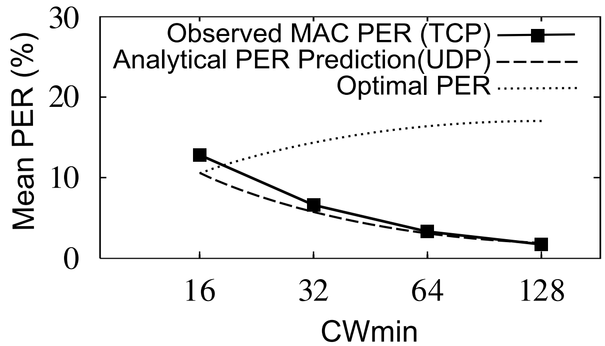


Figure 3.10: Collision-rate behavior of TCP under varying CW$_{min}$ values.

$CW_{min} = 16$, but does not follow the curve for higher CW_{min} settings. Instead it tracks the PER for two backlogged stations, indicating that *in general, TCP's flow control cannot identify the MAC operating point at which throughput is maximized.*

Instead the measured results can be explained through an interaction between flow control and MAC layer channel access. A TCP workload leads to two backlogged stations for each access point because TCP requires an equal packet flow rate both up- and downlink directions (i.e., DATA and ACK). Regardless of the direction of the data traffic, the flow is limited by the AP's MAC layer channel access probability. Two backlogged stations means that on average the AP itself and one associate station are backlogged. If more associated stations were backlogged, they together would have a higher channel access probability than the access points, and the backlogged queues would empty. Similarly, adding more APs increases the cumulative channel access probability for APs, hence they can excite more client stations (one per AP).

Note that this result is not a function of the downlink dominance of the traffic [25]. We have also independently carried out simulations to compare a TCP-uplink dominated scenario and observed that direction of data traffic did not change the throughput or collision rates significantly ($<$1%).

3.4 Effect of AP Density on Multimedia over WLAN

In this section, we evaluate the effect of inter-cell interference on multimedia traffic, with and without Wireless Multimedia Extensions (WME).[6] We place a special emphasis on voice-over-IP (VoIP) application performance since most current video streaming traffic is not interactive,

[6] WME is an interoperability standard from the WiFi Alliance based on the IEEE 802.11e standard [24].

Table 3.4: Empirical VoIP Performance

No of APs	**VoIP Call #1**			**VoIP Call #2**		
	Packet Drop Rate (%)	*Avg. Jitter* (ms)	*Avg. Latency* (ms)	*Avg. Drop Rate* (%)	*Avg. Jitter* (ms)	*Avg. Latency* (ms)
1	0.14	2.11	54	0.78	2.16	78
2	1.55	4.16	77	1.12	2.50	65
3	2.97	4.82	101	1.84	5.76	138
4	11.40	8.63	304	10.82	7.85	191

i.e. it can be buffered and transported over HTTP/TCP (e.g., YouTube™ video streaming). The recent emergence of IEEE 802.11 in VoIP handsets and cell phones, however, can be expected to lead to increased VoIP usage on access points.

3.4.1 Performance over Legacy IEEE 802.11a

We first characterize VoIP performance in environments with legacy IEEE 802.11a stations using our realistic workload mix. For this purpose, we conduct ORBIT experiments configured as in Section 3.2.1 (IEEE 802.11a, 75 STAs with realistic arrivals, and up to 4 APs). We then designate two of the stations randomly as VoIP devices and allow them to each initiate a VoIP call towards our sink on the wired network of the testbed. VoIP sessions start at uniformly distributed random times during the experiment and last for 270 seconds. A VoIP call runs over an RTP/UDP session using G.711 with one sample per packet as the codec (Voice Activity Detection disabled). For each VoIP call, we measure packet error rate, mean jitter, and mean latency. Table 3.4 summarizes results from these experiments.

As per ITU-T Recommendation G.114 [52], we consider 150ms as the upper latency limit for acceptable VoIP communications. Similarly, we consider 75ms and 3% the upper limits on jitter and packet loss, respectively [53]. Our key observations follow:

- A single congested access point can support two VoIP calls with acceptable performance, even without the use of WME quality of service differentiation.
- The two VoIP calls can be supported with adequate performance only in a congested environment with no more than two interfering APs. With the addition of the third AP, packet loss rate and latency values for both VoIP calls approach the limits described above. For four APs, packet error rates and latency reach an unacceptable 11% and 300ms, respectively.

Overall, the addition of two VoIP flows has a negligible effect on the throughput of the other

TCP flows, since each VoIP flow only generates an application layer load of 64 Kbits/s. We do not observe unacceptable jitter in any repetition of the experiment, likely because the VoIP receiver was placed only one hop away from the VoIP sources on our wired testbed network.

These results indicate that for WLAN-enabled hybrid phones to work well in interference-limited multi-AP deployments (e.g., apartments with many residents each using their own WLAN) changes to legacy IEEE 802.11a/b/g are inevitable. Next, we investigate whether the recent changes provided by WiFi Alliance's WME address this problem.

3.4.2 Performance over IEEE 802.11e (WME)

For WME experiments, we relied on QualNet simulations, because MadWifi WME implementation [44] on Atheros WLAN hardware was in its early stages of development while we conducted our experiments. We first conduct WME-enabled version of the VoIP experiments we carried out in the ORBIT testbed with two VoIP devices, for the purposes of facilitating direct comparisons. Then, we also experiment with a higher number of VoIP sessions as well as scenarios involving video streaming sessions. For the latter, we use the Mean Opinion Score (MOS) metric to characterize user perceived audio quality of VoIP calls. For the video streaming traffic, we configured the throughput mean and variance of the offered load to those captured from a five minute long movie segment encoded from DVD using DivX 5.1. The mean data rate of the stream used in the experiments presented here was 382.5 Kbit/s.

We assign application traffic to the WME MAC access queues as follows: video streaming to the *Video* queue (*WME_AC_VI*), and VoIP calls to the *Voice* queue (*WME_AC_VO*) and all other TCP data flows to the *Best Effort* queue (*WME_AC_BE*). EDCA parameter-set values for all four access queues conform to the default values suggested by the IEEE 802.11e standard [24].

Results from the WME-enabled simulations of one and four AP networks are given in Table 3.5, and they can be compared to the ones reported in Table 3.4, obtained from the experiments on the ORBIT testbed. Figure 3.11 presents the results from the second set of

Table 3.5: Simulated VoIP Performance

No of APs	**VoIP Call #1**			**VoIP Call #2**		
	Packet Drop Rate (%)	*Avg. Jitter* (ms)	*Avg. Latency* (ms)	*Avg. Drop Rate* (%)	*Avg. Jitter* (ms)	*Avg. Latency* (ms)
1	0	1.01	30	0	1.21	34
4	4.11	6.86	117	6.20	7.24	126

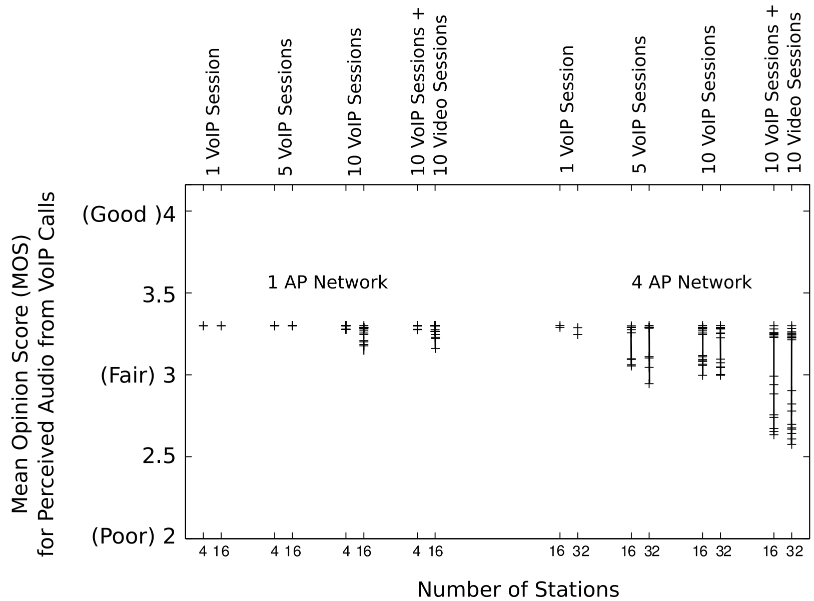


Figure 3.11: Simulated VoIP MOS results for varying intensities of multimedia traffic carried over one and four AP networks using IEEE 802.11e (WME).

experiments where we report the subjective quality for each of the VoIP calls in terms of MOS. These experiments vary the number of VoIP and video streaming sessions and the number of interfering access points. We observe that:

- Use of WME improves media quality in the multi-cell case. Average latency values of the two VoIP calls in the four AP network remains acceptable, in contrast to the non-WME case. With WME, up to 10 concurrent VoIP sessions can now be supported with a MOS at the fair level, compared to 2 VoIP sessions with 2 interfering access points without WME.

- With increasing amounts of media traffic, the MOS for VoIP sessions still indicate significant quality degradation. At 10 video and 10 concurrent VoIP sessions, average quality approaches the poor rating.

- While not shown in the figure, we also observed a tenfold increase in video streaming jitter for the four AP scenario. For non-interactive videos this can likely be addressed through application buffering. For video conferencing applications, however, this jitter may be unacceptable.

As expected, we observed a throughput reduction for best effort traffic when more media

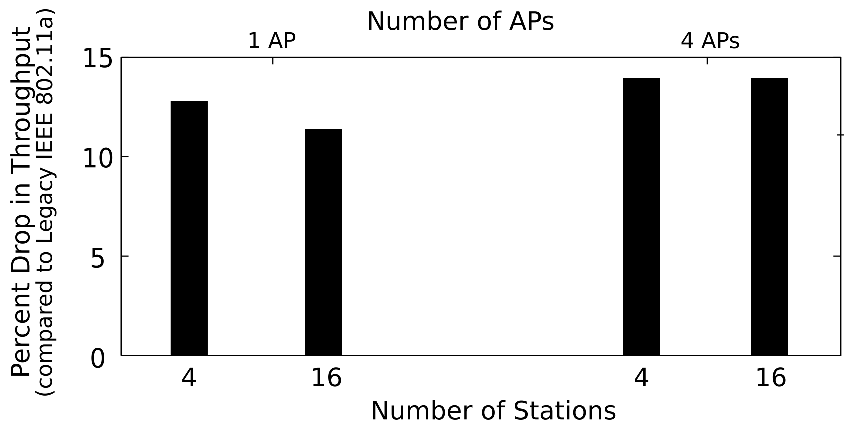


Figure 3.12: Default value for AIFSN parameter of IEEE 802.11e *Best Effort* queue results in decreasing throughput for bulk data traffic, even in the absence of any multimedia traffic.

streams are added, demonstrating effective MAC layer prioritization of media traffic. Also note that switching to WME reduces capacity even without media traffic, because the best effort queue to which all regular traffic is assigned uses a larger interframe space than default IEEE 802.11a/b/g (i.e., an AIFSN of 3). This increase in the interframe space by one slot (9μs for IEEE 802.11a, and 20μs for IEEE 802.11b) results in an 14% reduction of best effort throughput with 32 stations and 4 APs in IEEE 802.11a. We observed that up to 14% drop in throughput was likely for 32 station 4 AP WME simulations, when compared to the results from legacy IEEE 802.11a (see Figure 3.12).

Overall, WME provides an improvement in interference tolerance but not a complete solution. Additional measures will be needed to make media traffic resilient against inter-cell interference. We investigate such additional techniques next.

Chapter 4

$WiPhi$: Improving Performance with CW Adaptation

We have seen that inter-cell interference reduces cumulative throughput in WLAN systems with TCP-dominant workloads much more significantly than intra-cell contention. Also, the detrimental effects of inter-cell interference are more severe on multimedia traffic.

We propose a contention window adaptation solution that addresses this challenge at the MAC layer, named $WiPhi$. We propose a MAC layer solution because the root cause of the problems we observed was increasing contention due to added interference, and it could be best addressed with a local solution at the wireless last hop. End-to-end approaches (e.g., TCP tweaks) or network-wide solutions (e.g., IP tweaks) are undesirable due to changes that will be required at millions of hosts all over the world. The other PHY/MAC layer techniques are complimentary to our proposal, they can increase overall capacity, but in chaotic dense deployments several interfering access points might still remain after applying these techniques. Most relevant ones are visited in Chapter 2.

We first describe and evaluate the contention window adaptation approach, and then end with a discussion that highlights other challenges that should be addressed for a complete system solution.

4.1 CW_{min} Adaptation Using Active AP Count

It is well-known that IEEE 802.11 MAC performance can be improved by selecting a CW_{min} appropriate for the current number of active clients in the network [28–30]. We propose instead that *the selection of CW_{min} for a typical TCP dominated multi-cell WLAN system should be based on the number of interfering access points, not the number of clients.* This method provides the following advantages:

- Knowing the number of access points allows more accurate selection of CW_{min} under a TCP workload than knowing the number of transmitting clients. As our earlier analysis has shown the collision rate under TCP is determined by the number of access points, since TCP flow control regulates client activity.

- The number of active APs is easier to obtain than the number of clients, since APs announce their presence through beacon packets and tend to remain active and stationary over longer durations. Clients might also help determining the number of access points by listening for AP beacon packets.

Given the number of active APs N_b, the optimal contention window can be derived by combining our insight regarding the number of backlogged stations under TCP workloads with earlier contention window adaptation work. According to [30] bandwidth can be maximized with a contention window $CW_j^* = \frac{\sqrt{2\beta T'_D}}{r_j} + 1$, where $\beta = (\Sigma_{j=1}^{M} N_j r_j)^2 - \Sigma_{j=1}^{M} N_j r_j^2$ and N_j is the number of active users for each priority queue. Since we do not consider multiple priority queues here, β reduces to $N(N-1)$. Also substituting $T'_D = \frac{T}{T_s}$ yields

$$CW_{min} = \sqrt{2N(N-1)\frac{T}{T_s}} + 1, \tag{4.1}$$

where T is the time required for the transmission of a MAC frame (excluding DIFS and backoff, including ACK reception), and T_s is the duration of a MAC time-slot. Using the insight from Section 3.3 that the number of backlogged stations equals twice the number of active access points, we can substitute $N = 2N_b$, which formulates CW_{min} in terms of the number of active access points as

$$CW_{min} = 2\sqrt{N_b(2N_b - 1)\frac{T}{T_s}} + 1. \tag{4.2}$$

4.2 Implementation of CW_{min} Adaptation

This subsection describes an AP-centric algorithm for determining the number of active access points N_b and distributing the CW_{min} setting. While a client-centric or hybrid approaches are also possible, we have chosen this approach because it minimizes the number of devices that need to be modified to the access points.

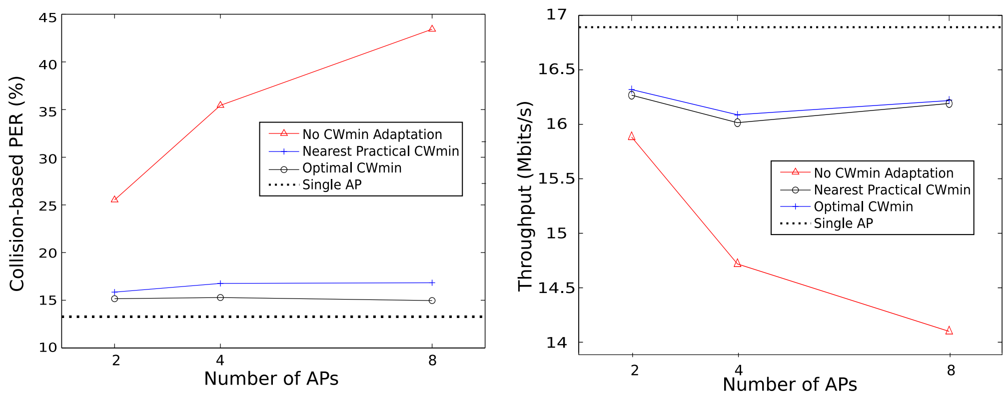
The number of active access points N_b should ideally only include access points that actively communicate with at least one client. Since APs transmit beacons even when none of their clients are active, however, the proposed algorithm determines the expected $E[N_b]$, by considering the percentage of the time neighboring APs consume channel resources. The IEEE

```
// Accepts:  CurrentChannel, t_p, T, T_s
// Updates:  (System Parameter) CW_min
1  for every t_p seconds do
2      NeighborAPList[] = doBackgroundScan(CurrentChannel);
3      for each AP_i in NeighborAPList[] do
4          sendFrame(80211H_MEASR_REQUEST, AP_i);
5          recvFrame = readFrame(80211H_MEASR_REPORT);
6          %_Backlg_i=parseFrame(recvFrame, BACKLOG_FRACTION);
7      end
8      %_Busy_self = measure(CHANNEL_BUSY_FRACTION, self);
9      E[N_b] = 1/100 · [%_Busy_self + Σ_∀i %_Backlg_i];
10     CW_min = ⌈2 sqrt(E[N_b](2E[N_b] − 1) T/T_s) + 1⌉;
11     update_EDCA_ParamSet(CW_min);
12 end
```

Algorithm 1: CW_{min} update algorithm APs execute. It finds the effective number of active APs in the vicinity and uses this to update CW_{min} and propagate it further down to its STAs with the next beacon.

802.11h standard [54], which defines spectrum and transmit power management extensions, allows for channel-related measurement-exchange mechanisms that can be extended to support $E[N_b]$ calculation in a standards-compliant way.

The AP-centric approach for this purpose is outlined in Algorithm 1. The algorithm periodically calculates CW_{min} every t_p $seconds$ by requesting neighboring APs (visible through their beacons) to report the percentage of the time their frame queues were not empty (i.e. $BACKLOG\_FRACTION$) during the last t_p seconds. This message exchange can be implemented within the IEEE 802.11h measurement request/report framework by using one of the *reserved* measurement type definitions. Querying AP sends an IEEE 802.11h measurement request frame ($80211H\_MEASR\_REQUEST$) to the AP of interest, and the receiving AP responds to this with a measurement report frame ($80211H\_MEASR\_REPORT$) including its $BACKLOG\_FRACTION$ measurement (see lines 2-7 of the algorithm). These reports from neighboring APs, when combined with the measuring access point's own wireless channel-busy percentage measurement ($CHANNEL\_BUSY\_FRACTION$), allow determining $E[N_b]$ more precisely; a neighboring AP which is backlogged 50% of the time for example, increases $E[N_b]$ by $\frac{1}{2}$. $E[N_b]$ calculated this way is used to update CW_{min} (see line 10 of the algorithm), which in turn is included in the next AP beacon to be announced to the STAs of this AP through IEEE 802.11e EDCA Parameter Set information element [24].



(a) Collision-based PER (b) Throughput

Figure 4.1: Simulations with 32 STAs showing potential gains from incorporating AP-Count-based CW adaptation into WLAN with inter-cell interference (in (b), throughput curves for "Optimal CW_{min}" and "Nearest Practical CW_{min}" almost overlap).

4.3 Simulation Results

We conduct fixed-bitrate simulations with 32 STAs to observe the potential improvement that our CW_{min} selection can provide. The results assume that all stations accurately estimate the number of active access points. Figure 4.1 shows that for both exact CW_{min} values suggested by the Equation 4.2 above, and for the nearest power of two (which is a practical restriction in today's WLAN hardware), CW_{min} adaptation based on the number of APs reduced the collision-based losses significantly, keeping them close to the residual collision losses of a single AP scenario. Also, the granularity of CW_{min} adaptation in powers of two does not have a significant negative effect on performance. Note that reduction in PER not only improves throughput but also reduces delay that affected multimedia streams as described in Section 3.4.

Note that the optimality of this CW_{min} tuning strategy, in terms of achieving proportionally fair bandwidth allocation and maximizing utilization has already been shown in [30]. Note also that, in the trivial case of a single cell, $N_b = 1$ and CW_{min} is reduced to a constant, consistent with the result that, in the case of a single cell, the number of backlogged stations is constant. Finally, the proposed CW_{min} adaptation strategy is optimized for TCP-dominated workloads (as compared to the earlier work which provides a solution for UDP-dominated workloads [30]). In mixed traffic environments, where UDP accounts for a significant share of network traffic and is used on many stations, the proposed solution would have to take into account the increase in the number of active nodes due to this additional UDP traffic. Estimating

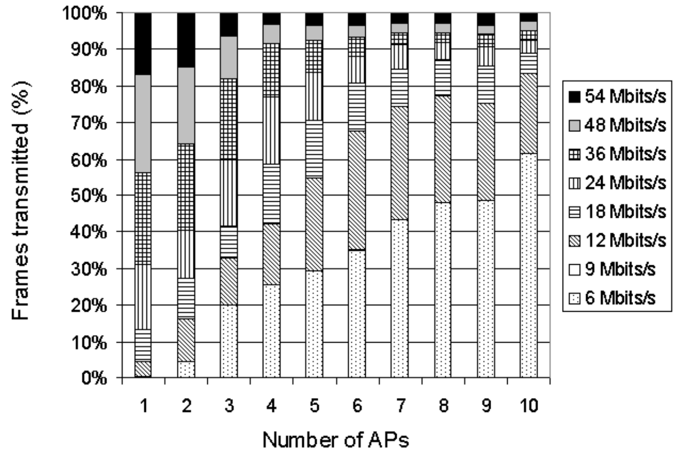


Figure 4.2: Empirical IEEE 802.11 bitrate distribution with the Samplerate algorithm.

the exact number of active nodes in this case remains an interesting open topic for future work.

4.4 Discussion: Relationship to Collision-resilient Bit-rate Adaptation

Collision-aware rate adaptation is a further MAC layer technique to improve performance in congested IEEE 802.11 networks. Prior work [20, 55, 56] has shown that many bitrate algorithms choose low bit-rates in congested environments—short-term collision errors are misinterpreted as longer-term changes in path loss. The bitrate reduction further decreases available capacity, leading to more collisions.

One might believe that CW_{min} adaptation eliminates the need for collision-aware rate adaptation techniques because it substantially reduces collisions. From the CW_{min} adaptation results in Figure 4.1(a), we observe, however, that even at the optimal operating point a residual collision rate remains. We also observe from our multi-cell experiments that bitrate adaptation is responsible for a substantial fraction of throughput loss and that even with the residual collision rate bitrate adaptation in the MadWifi driver does not choose optimal rates. Figure 4.2 shows the bit-rate distribution of SampleRate [57], the default bit-rate adaptation scheme in the MadWifi driver, for up to 10 access points and 50 clients. With more APs, an increasing percentage of frames are transmitted at the lower bitrates, even though Signal-to-Noise Ratio (SNR) at most receivers in our experiment environment is high enough for communication at 54 Mbits/s and the total number of clients remains constant. For example, 60% of frames use 6 Mbits/s for the 10 AP experiment. Even with the residual collision rate in the one AP case, however, less than 20% of frames are transmitted at the optimal bitrate of 54 Mbit/s.

Hence, we believe that improving the collision-resiliency of bit-rate adaptation mechanisms must be an integral part of the solution space. Based on the observations in [55, 56, 58], a potential solution to achieving collision resiliency could be through the adaptive use of the RTS/CTS mechanism. Rate adaptation algorithms could leverage this backwards-compatible mechanism such that they ignore RTS losses, which could be due to collision, and react only to data packet losses that occur after the channel is reserved. Packet losses that occur when the channel is reserved are most likely because SNR at the receiver is not high enough to support the current bit-rate. Thus, this mechanism could be used to distinguish between losses due to collision and those due to poor channel conditions.

4.5 Conclusions

In this work, we have investigated the effect of inter-cell interference on unplanned WLAN performance. While inter-cell interference should ideally be avoided through careful access point placement, frequency selection, and transceiver parameter control, current chaotic wireless deployments and the limited number of available channels make inter-cell interference a reality. Therefore, we have measured the effect of such interference both in a testbed with more than one hundred nodes and through simulations. We have

- found that cumulative throughput degrades significantly, by 50% with four interfering access points, while it remains remarkably robust with over one hundred clients in the single cell case.
- verified that TCP's flow control leads to an average of $2i$ nodes concurrently backlogged in the network, where i is the number of actively interfering access points. Thus, the collision rate increases with the number of access points. TCP does not adjust its flow rate to optimize throughput for different CW_{min} choices.
- showed that increased collision rate with inter-cell interference also affects media streaming. With only two congested interfering access points, VoIP mean opinion score (MOS) is unsatisfactory.

These findings underline the need to consider system performance in addition to studying the MAC layer in isolation. They also lead to a practical recommendation that largely recovers cumulative throughput loss from inter-cell interference. TCP's control over the number of active nodes allows for a novel approach to selecting the appropriate CW_{min} based on the number of active access points, not the number of active clients. We also point out the importance of collision-resilient rate adaptation algorithms. Even with improved CW_{min} selection, a 20% cumulative throughput gain is possible through collision-resilient bit-rate adaptation.

Chapter 5

$HeedNet$: Improving Performance with ISP Participation

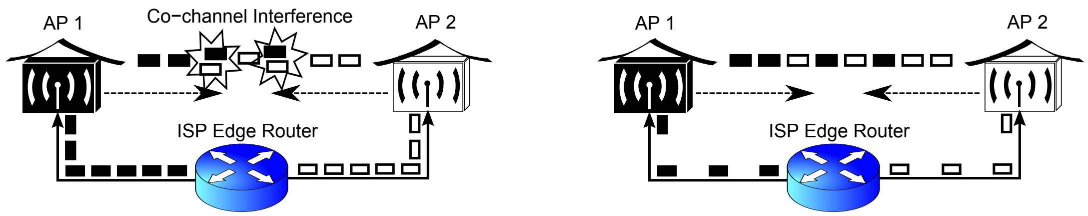
This section describes the design and implementation of $HeedNet$, an orthogonal solution for unplanned high-density WLAN interference management, where an ISP actively participates in Layer-2 interference management by scheduling Layer-3 routing to those APs having the potential to interfere with others.

5.1 Overview

In previous chapters, we have analyzed how high-density WLANs performed under typical workloads and showed that, if left unmanaged, the increasing collisions due to interfering cells can bring the cumulative system performance to its knees. As a solution, we have proposed $WiPhi$ to adapt IEEE 802.11 channel access mechanism to the number of active WLANs. This solution, however, has an important drawback. If the large-scale adoption of a modified MAC protocol is not feasible in the domain of application, then the utility of $WiPhi$ would be severely limited. For example, it is very costly for an Internet Service Provider (ISP) to ship new wireless access points (APs) to its subscribers with new MAC protocols installed. Similarly, issuing firmware updates for all installed APs and requesting users to install these updates are prohibitively costly[1].

In this chapter, we introduce $HeedNet$ – a backhaul network driven approach to interference management. $HeedNet$ lets the edge routers of the ISP network participate in residential high-density WLAN interference management by scheduling outgoing bulk IP traffic towards those APs that are determined to interfere with each other. Since $HeedNet$ only requires a utility software on some of the WLAN clients, and no changes for the AP and the underlying MAC protocol, it could be easily deployed, as most of the functionality of $HeedNet$ is designed to

[1]Although automatic remote installation of firmware updates are possible and used by some ISPs (e.g. BT in United Kingdom), most ISPs avoid them due to reliability and support complications.



(a) High-density Residential WLAN Interference (b) Reduced Interference with HeedNet Scheduling

Figure 5.1: (a) Interfering residential APs (b) Less interference after ISP router schedules outgoing packets.

be managed by a software agent on the ISP edge router.

$HeedNet$'s objective is to improve high-density WLAN system utility by reducing the amount of collision-induced interference by controlling the flow of IP packets due to bulk traffic at the ISP edge router towards those access points determined to cause interference. The information on the APs that have the potential to interfere are collected by a subset of the clients in the domain of deployment (i.e., some of the client devices of the subscribers) with the help of an ISP-distributed utility software. These client surveys are reported back to the $HeedNet$ agent that runs on top of the ISP edge router. Using the information, $HeedNet$ agent creates a TDM-like schedule, and buffers the IP packets to delay routing to all but one AP. With the right time-slot duration and AP schedule ordering, most of the overlapping transmissions could be serialized, thus reducing the amount of destructive collisions in the air. Figure 5.1 illustrates this fundamental concept behind $HeedNet$'s design. Such a management capability provides an ISP a great opportunity to improve service for the subscribers having to deploy APs in close proximities and to reuse limited spectrum due to the density of the neighboring APs. $HeedNet$ schedules packets at Layer-3, hence can work with virtually all wireless APs and all IEEE 802.11 variants using DCF channel access mechanism.

Specifically, $HeedNet$ aims to i) increase overall system capacity that is hampered by interfering access points in a high-density setting, and ii) improve fairness by preventing some of the WLANs from experiencing a very low throughput share. The analysis in Chapter 3 provides an assessment of the significant degree of performance penalty even with a small number of access points.

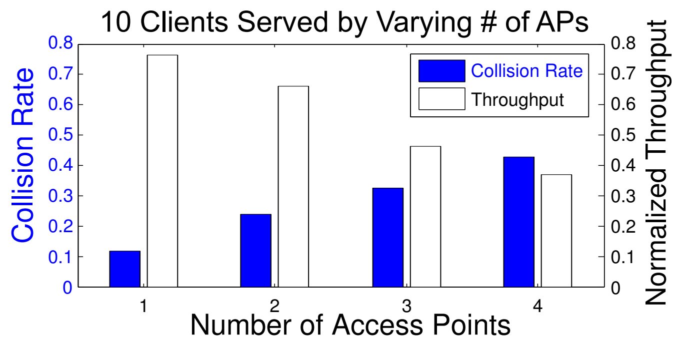


Figure 5.2: System capacity with increasing APs. The same workload causes more collisions thus reduced system throughput.

5.2 Severity of Interference

We start with an experimental demonstration of the detrimental inter-cell interference effects on the system capacity using a WLAN deployment, emulating a small multi-family apartment building. In our experimental setup, a fixed number of IEEE 802.11g clients can be served by a varying number of access points (AP), all placed randomly in a building of 400 square meters space and all operating on the same 2.4GHz ISM-band Channel 1. For each run, we increase the number of APs that can serve the clients, while the traffic workload from the clients are kept constant. We make use of the same realistic traffic mix described in [12], with web browsing, file transfers and VoIP traffic. Summary results from these experiments with 10 clients and up to four APs are outlined in Figure 5.2. Throughput values are normalized to maximum achievable TCP throughput by a single flow in a one-AP/one-client high SNR baseline configuration.

We observe that the system capacity shrinks about 40% when four APs are used instead of one. This is because the number of collisions increase with more APs as TCP flow control can not regulate the workload to keep the collisions down at the optimal level needed by the underlying IEEE 802.11 MAC protocol, in agreement with Ergin et.al. [12]. Also, default rate control algorithms (e.g., ARF or SampleRate) on WLAN devices worsen the performance as they mistake increasing collisions as packet errors due to bit-errors on the channel [59].

5.2.1 Practical Densities and Interference

Having observed the significant reduction of system capacity due to the interference from only four WLANs, we would like to quantify the number of strongly interfering WLANs typically

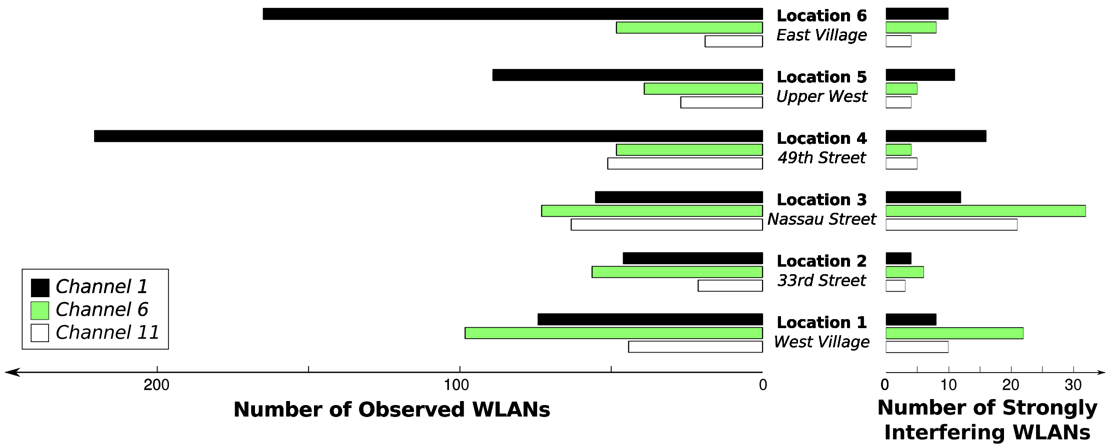


Figure 5.3: Residential WLAN observation experiment we conducted in Manhattan, NY. Hundreds of WLANs seen from a single location, with tens of them having the potential to strongly interfere with each other.

observed in current residential environments. We have conducted passive data collection experiments in six arbitrarily selected residential locations in Manhattan, NY, with many apartment buildings. We have used three off-the-shelf IEEE 802.11g clients on three PCs, each observing one of the three orthogonal channels on 2.4GHz ISM-band (Channels 1, 6, and 11). Each wireless client is equipped with a regular 3dBi rubber duck antenna, placed inside a passenger vehicle, which is parked on the street to record channel activity for 20 minutes. Our data post-processing marks unique WLANs as *observed*, if 30% or more of its AP beacons were successfully overheard during the experiment session. Also, WLANs whose APs overheard with over 20dB SNR are classified under *strong interferers* category, indicating their potential to create detrimental interference should there be enough traffic load on their respective networks. Results from our experiment are summarized in Figure 5.3. We have observed that hundreds of WLANs operate on the same channel in typical residential areas, and having tens of those with the potential of strong interference should be considered today's norm. Therefore, we continue by outlining our high-density residential WLAN scenarios that are likely in near future, based on our observations from existing deployments.

5.2.2 Design of the High-Density Residential Scenarios

Typical urban high-density residential deployments take place in buildings that are used in apartment settings, thus we first survey apartment buildings in a metropolitan city and characterize them based on the number and distribution of floors/units. Our public dataset, obtained

Table 5.1: Statistics from the apartment dataset

	Number of Units		**Number of Floors**	
	Mean	**Variance**	**Mean**	**Variance**
Low-Rise	6	1	5	1
Mid-Rise	15	2	9	3
High-Rise	136	26	34	11

from [60] in August 2009, is composed of 1379 apartment buildings with rental units, located in Manhattan, New York City. We have clustered the apartment data into three groups, similar to the ones used in real-estate classification, as low-rise, mid-rise, and high-rise. The normal fit statistics derived from the dataset for these groups, forming the basis for our scenario design, are summarized in Table 5.1. The statistics are rounded to the closest integer values.

We create three representative apartment buildings for our scenarios, using the statistics derived. The *small-sized* apartment building is 5-storeys high with one-unit per floor, the *medium-sized* one is 9-storeys high with two-units per floor, and the *large-sized* apartment building is 34-storeys high with four-units per floor, as illustrated in Figure 5.4. Each unit is assumed to be approximately 700 sq.ft. with 8 ft. ceilings and 3 feet corridors between the units. We have written a small utility software to lay these hypothetical floor-plans down and randomly place one access point and two clients per unit's 3-D space. Coordinate outputs from this utility is written to a file, later to be used by our simulation software. Each WLAN (consisting of one AP and two STAs) is assigned a channel according to the WiFi channel statistics published by WiGLE.net [61], obtained from over 19 million real networks, as of August 2009. Based on these data, about 35% of the APs in our scenarios are observed on 2.4GHz ISM Band Channel 6 and constitute the largest group of the WLANs that have the potential to interfere with each other [2]. Therefore, in each of the apartment scenarios, we randomly pick 35% of the units and consider the WLANs in them for our simulations of interference evaluation. As the number of interfering WLANs (35% of five units) in the small-sized apartment scenario (SSA) is not significant, we will report only on the experiments for the medium-sized apartment scenario (MSA – seven interfering WLANs) and the large-sized apartment scenario (LSA – fifty interfering WLANs).

[2] There is another 12% of WLANs that operate on adjacent channels partially overlapping with Channel 6. However, we only consider the stronger co-channel interference case here.

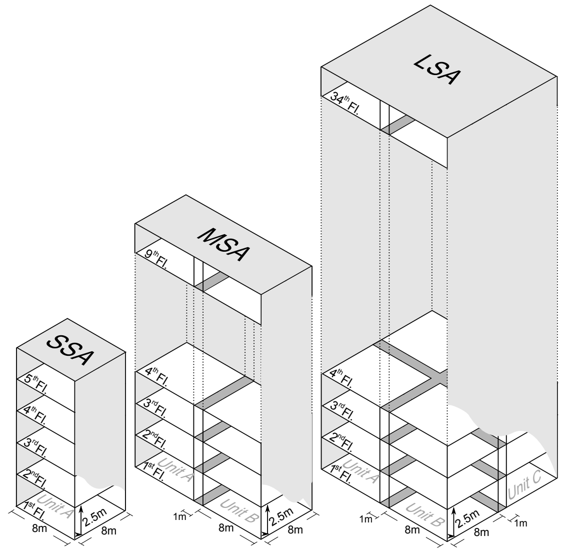


Figure 5.4: Residential setup used in simulation scenarios with 5-units in the small (SSA), 18-units in the medium (MSA), and 136-units in the large-sized apartment (LSA). One AP and two clients are placed randomly in 3D space of each unit and 35% of the units are assumed to have WLANs that are on the same channel.

5.2.3 Simulation Setup

The topologies of the described two scenarios are implemented as part of a QualNet v.4.5 [23] simulation. Simulator software places all nodes according to the 3-D coordinate file, created as described in the previous section. The network topology for the simulation is illustrated in Figure 5.5. The APs are connected to an ISP edge router through broadband links that behave similar to DSL or cable connections. We assume that the apartment building is served by a single ISP. Clients in the units try to download a large file from one of the ten different Internet servers using TCP as the transport protocol. The servers are reachable from the ISP via high-capacity low-delay backbone links. For baseline experiments, we assume an unmodified legacy IPv4 router at the ISP site. The other parameters used in the simulation experiments are described in Table 5.2.

IEEE 802.11g at 2.4GHz spectrum was used in the simulation experiments where each experiment was run for 120-seconds and repeated 10-times with different seeds. First second of all experiments is used as the warm-up period for client-to-AP association signaling, and the following second is used by all clients to initiate their downloads from the servers. All downloads are terminated with the 120^{th} second of the experiment. The results are processed to indicate 95-percentile of the variation. In addition to the two apartment scenarios (i.e., MSA

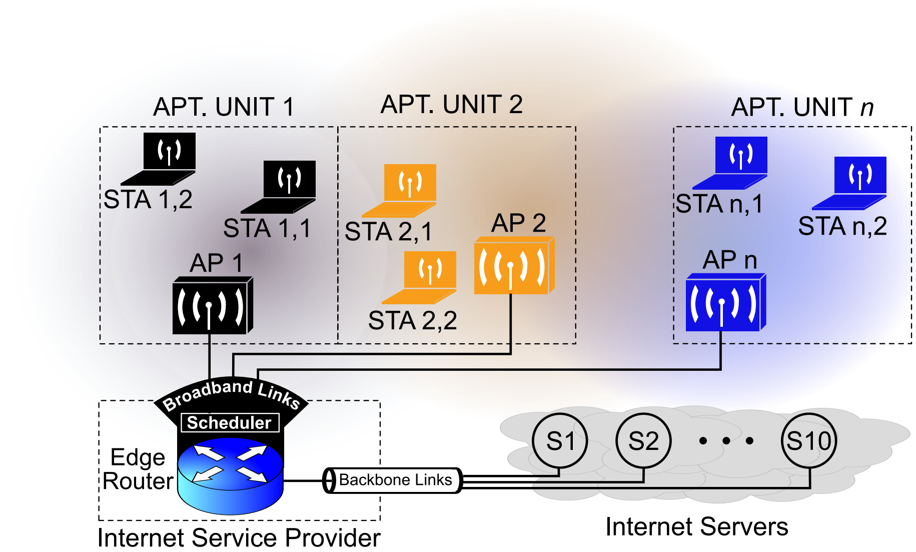


Figure 5.5: Topology used for assessing the high-density residential scenarios. All WLANs are assumed to be served by a single ISP router with packets to and from ten different Internet servers.

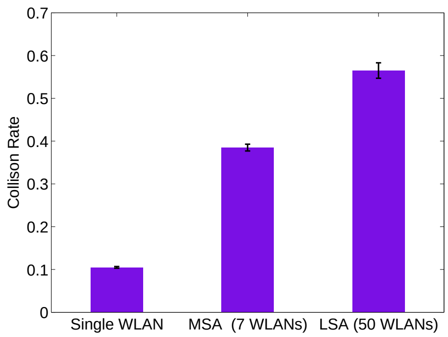
and LSA), a single WLAN scenario was also simulated to facilitate comparisons. The single WLAN experiment consists of one AP and two stations with the same workload described for the larger scenarios. The collision rate (percentage), total system capacity (bytes transferred) and per-WLAN capacity fairness (Jain's index) are analyzed from the simulation traces and presented in the next subsection.

5.2.4 Baseline Results

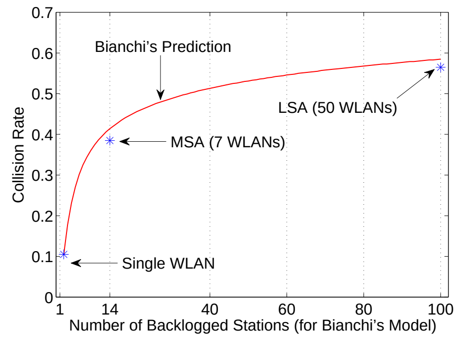
Results from our baseline simulation experiments are summarized in Figure 5.6. From Figure 5.6(a), we observe that the amount of collisions, in comparison to the single WLAN scenario, increases almost four and six-times in MSA and LSA scenarios respectively. To verify if these drastic increases in the amount of collisions are in agreement with earlier research on the

Table 5.2: Parameter Summary for Simulations

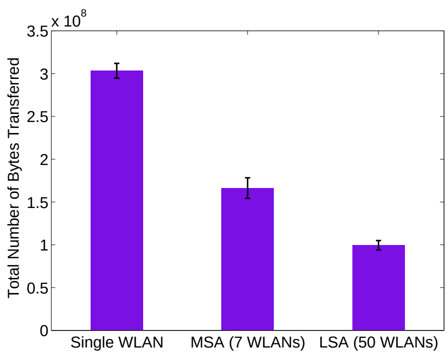
Attribute	Value
PHY & MAC (WLAN)	IEEE 802.11g @ 2.4GHz, Two Ray Ground, Tx. power of 18 dBm, DCF access in BSS mode (RTS/CTS off) with 10 retries max., Auto Rate Fallback, Rx. Sensitivity of -69dBm (at 54Mbps)
PHY & MAC (Broadband)	Abstract MAC with 1ms avg. propagation delay, 10Mbps capacity from APs to the ISP edge-router. Then, 100Mbps capacity from ISP to the Internet servers
NET	IPv4, IP Queue size of 75000 bytes
TCP	NewReno with RFC 1323, Max. segment size of 1500 bytes, Send/Rcv buffer size of 110KB, Delayed ACK disabled



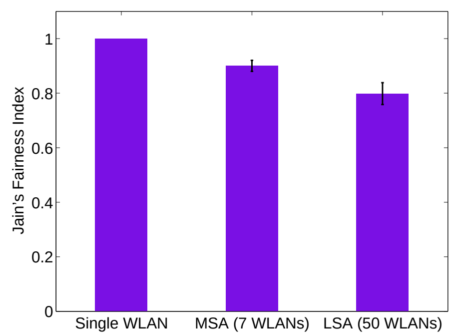
(a) Collision Results



(b) Collision Comparison with Bianchi's Model



(c) System Capacity Results



(d) Fairness Results

Figure 5.6: Severity of High-Density Residential WLAN Interference. Errorbars indicate 95-percentiles of the resultsets.

subject, we do the following comparison. It is reported in [12] that the collision-rate would be equivalent to a backlogged IEEE 802.11 network with nodes as many as two-times the number of WLANs involved. We plot, in Figure 5.6(b), the collision-rate results from the simulation scenarios (star-shaped points) together with Bianchi's model prediction (continuous line) [15] for equivalent (i.e., two-times the number of WLANs) systems with backlogged stations. We confirm that each new interfering WLAN increases the collision-rate of the high-density deployment we simulate by an equivalent amount coming from two backlogged IEEE 802.11 stations.

From Figure 5.6(c), we observe that increasing collisions result in system capacity reductions of approximately one half and two thirds of the single WLAN system capacity, for MSA and LSA scenarios respectively. Hence, for a large apartment complex, there exist a 3X throughput improvement potential for an ISP deploying a solution that can bring the collision rates down towards 10% (i.e. the collision rate observed in the single WLAN scenario).

In addition to the total capacity loss, per-WLAN capacity fairness is considerably hampered with increasing interference, as visible in Figure 5.6(d). LSA scenario, with its 50 WLANs, has a Jain's fairness index value [48] of 0.79, where an index value of one indicates a perfectly-balanced sharing (i.e., the single WLAN baseline scenario).

5.3 HeedNet Design and Algorithms

HeedNet aims to reduce the effect of destructive interference by scheduling the bulk traffic in the downlink direction towards interfering APs. We start this section by outlining three most relevant questions, answers of which shape the design of HeedNet. Then, we provide details of the HeedNet design with its system components and algorithms. We finally evaluate the algorithms on their performance.

How do we know which WLANs to schedule? Not all APs served by an ISP will interfere with each other, and once they do, the level of interference would change from one AP to the other. HeedNet incorporates a station-assisted interference assessment algorithm to determine the subset of APs to be scheduled. The more WLANs HeedNet schedules together, the less the amount each can be given for exclusive routing. For this reason, HeedNet tries to schedule WLANs in as small interfering groups as possible without sacrificing interference reduction gains significantly (§5.3.3).

What should be the scheduling interval for maximum benefit? The tradeoff between exclusively serving an AP with a longer time-slot and introducing too much scheduling delay for the rest of the APs is critical on HeedNet's performance. HeedNet dynamically adjusts its scheduling parameters based on its system performance measurements (§5.3.4).

In what order should the APs be scheduled? Due to the processing involved in forwarding a packet from the ISP edge router to its destination AP, a non-trivial jitter exists in the time interval after the scheduled packet is sent out from the ISP and before it is in the air. Therefore, HeedNet needs to carefully manage the order in which the APs are scheduled, so that the unwanted interference from back-to-back scheduled APs are minimized (§5.3.5).

5.3.1 HeedNet Components and Operation

We design HeedNet to be an efficient software agent that can run on any ISP edge router (IER). Also, a small software utility complements the HeedNet agent, and makes an observer out of

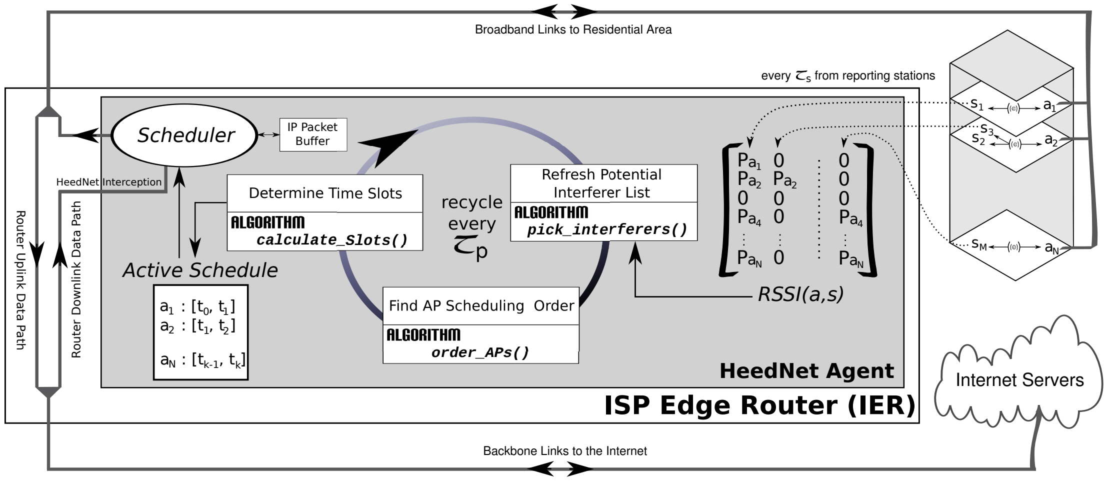


Figure 5.7: Overview of HeedNet Operation with Functional Units

the client it is installed onto. The functional units that constitute the HeedNet design and their relationships are depicted in Figure 5.7.

In the IER, HeedNet agent intercepts the IP packets that are sent towards the APs (i.e., downlink) to see if the current scheduling policy applies for the particular packet type. If so, the scheduling unit inspects the packet destination and uses the *Active Schedule* to determine whether the packet should immediately be released to the outgoing port or it should be stored in the associated IP packet buffer until the scheduled time-slot for this WLAN becomes active. HeedNet agent's control loop is periodically invoked every τ_p ms to re-assess the WLAN interference and construct an updated schedule for the WLANs served by this router. The control loop uses the AP RSSI observations reported by a subset of the stations as its main ingredient. Following our assumptions and the notation in the next subsection, we describe the details of the algorithms to select the interfering WLANs and to determine the right scheduling parameters.

5.3.2 Assumptions and Notation

We assume a network model where APs and stations are deployed in a high-density setting and each station is associated with the access point in the same dwelling unit. The network is composed of: (i) a set of APs $a_i \in \mathbb{A}, i = 1, \ldots, N$, and (ii) a set of stations $s_j \in \mathbb{S}, j = 1, \ldots, M$.

We also assume that a subset of the stations, $s_l \in \mathbb{S}_l \subseteq \mathbb{S}$, $l = 1, \ldots, M' \leq M$, provide assistance to HeedNet by executing a small client software that records a list of observed APs

Algorithm 1 $pick\_interferers()$ [Interferer Selection]

Require: Normalized $RSSI(a, s)$ to $[0, 1]$

$\aleph(Z) \triangleq$ {Cardinality of set Z}
$\mathbb{A}_k \leftarrow \emptyset$ {To Be Scheduled}
$\mathbb{A}_e \leftarrow \emptyset$ {To Be Excluded}
for each AP $a_i \in \mathbb{A}$ **do**
    $r = \mathbb{R}(i) = \aleph_l(RSSI(a_i, s_l) > 0)$
    $\mathbb{R}_{aug}(i) = r. \prod_{l, RSSI(a_i, s_l) > 0} RSSI(a_i, s_l)$
    if $\mathbb{R}_{aug}(i) < \gamma_{th}$ **then**
        $\mathbb{A}_e \leftarrow \mathbb{A}_k \cup a_i$
    else
        $\mathbb{A}_k \leftarrow \mathbb{A}_e \cup a_i$
    end if
end for
return $\mathbb{A}_k, \mathbb{A}_e$

and their average beacon signal strengths. These RSSI recordings are compiled and sent to the HeedNet agent on the ISP edge router (IER) every τ_s ms. The signaling cost of the client monitoring utility is very low and the utility can be easily bundled with the software distribution found in ISP welcome packages. The agent software on IER-side constructs and maintains a matrix of RSSI values, $RSSI(a_i, s_l)$, from these reports, where l^{th} column of the matrix has information on all of the observed APs by the station s_l. The entries corresponding to the APs that were not observed by a given station are populated with zero.

5.3.3 Selecting Potential Interferers

An ISP would maintain information on their subscriber addresses and circuit termination locations, however, we believe that this information is very course for determining a subset of WLANs subject to interference management. Unnecessarily enlarging the set of APs to schedule negatively affects the spatial re-use characteristic of the WLAN. Therefore, HeedNet uses signal observations from participating clients to determine the WLANs that do not pose significant potential threat of interference to all other WLANs and exclude them from further scheduling. Specifically, the more and the stronger a given AP is reported by the observing stations, the higher the potential of interference due to this AP not being put on “to-be-scheduled” list by HeedNet.

Selection process is given in Algorithm 1 and can be summarized as follows. We first

Algorithm 2 $pick\_interval()$ [Interval Selection]

Require: Δ, N', Y

$\delta_t = \delta_{ts} = min(\Delta, 25s)/N'$
$stepSize = \delta_{ts}/5$
for every τ_p second **do**
  repeat
    wait(Y x τ_s) {till next observer updates come}
    $collDiff = E[collRates_{cur}] - E[collRates_{prev}]$
    $\delta_t = \delta_t - collDiff$ x $stepSize$ {update schedule}
  until $collDiff < 0.05$
end for

define an augmented rank operator to be used on the rows of $RSSI(a_i, s_l)$ after its entries are normalized to $[0, 1]$ interval. Augmented rank of a row, $\mathbb{R}_{aug}$, is obtained by multiplying the rank of the row, $\mathbb{R}$ (i.e., the number of non-zero entries) with all of the non-zero elements of the row. Thus, $\mathbb{R}_{aug}(RSSI(a_i, s)) \equiv \mathbb{R}_{aug}(i)$ is a scalar indicator of total potential interference due to AP a_i as seen by the observers, and it is always upper-bounded by M'. Note that the quality of this indicator in discriminating interfering APs is dependent on the number of observing stations and their relative placements. Our algorithm eliminates those APs $a_e \in \mathbb{A}_e \subset \mathbb{A}$ with $\mathbb{R}_{aug}(e) < \gamma_{th}$. All APs with $\mathbb{R}_{aug}(e)$ below this γ_{th} threshold are considered to have insignificant potential for interference, and withdrawn from the list of APs to be scheduled, which is $a_k \in \mathbb{A}_k \subset \mathbb{A}$, $k = 1, \ldots, N' \leq N$. We empirically determine that a γ_{th} value of 2 (i.e., an AP observed by two stations with *max RSSI* or by more stations with *less-than-max RSSI*) works fine for the network configurations we tested.

The scheduling scheme outlined in the following subsections operates only on the set of APs in $\mathbb{A}_k$. The interferer selection in Algorithm 1 is repeated every τ_p seconds (i.e., refresh period) to capture the possible changes in the WLAN deployment. Section 5.3.6 outlines the evaluation of the performance of this algorithm.

5.3.4 Determining Schedule Intervals

HeedNet allocates an equal time-slot of duration δ_t ms to each of the scheduled APs in $\mathbb{A}_k$. Therefore, IER routes IP packets exclusively to $a_k \in \mathbb{A}_k$ for a maximum of δ_t ms before moving onto the next AP in $\mathbb{A}_k$. If all of its awaiting IP packets are sent to a_k in less then δ_t ms, HeedNet immediately continues with the next AP on the scheduling list. Determining the appropriate δ_t value is critical to HeedNet's performance, therefore, we use a low-duty-cycle feedback-loop driven algorithm to tune this value.

The algorithm to determine δ_t exploits the beacon packets broadcast by the APs. For every Y^{th} RSSI observation report, HeedNet client utility will piggyback the number of successfully received beacons B_s, and the number of missed beacons B_m from its own AP, tallied since the last piggybacked report. If Y is selected large enough, the ratio of $B_m/(B_m + B_s)$ provides a reasonable and very low-cost estimate of collision rate, as observed by this station. These collision estimates are then averaged across all observers by the HeedNet agent and used as the search feedback parameter to assess current selection of δ_t value.

The interval selection algorithm requires a configuration-time parameter Δ, representing the maximum scheduling tolerance of the network. This parameter allows HeedNet to determine δ_t such that whenever N' APs are to be scheduled, the time interval between two rounds of service to a given AP never exceeds Δ, implying $\delta_t < \Delta/N'$. Although not all IP traffic flowing towards APs are scheduled by HeedNet, for the scheduled part, ISPs can still place an upper-bound on the maximum amount of time an AP is left unserved by specifying Δ appropriately. The local search, as given in Algorithm 2, navigates within this $(0, \Delta/N')$ search-space, starting at the point $\delta_t = \delta_{ts}$. After experimenting with different values, we found that a δ_{ts} value of $min(\Delta, 25s)/N'$ provides a good start-up for various network topologies, shortening the search iterations as opposed to a random starting point. After average collision estimates become available for $\delta_t = \delta_{ts}$ initial value, we start a gradient descent search [62] to find a local minimum of the observed collision rate. Each step of the search requires HeedNet to operate with the new δ_t value until observing stations piggyback their current collision estimates (i.e., every Y^{th} report). Algorithm stops if no more than 5% change in collision rate could be observed, and restarts from δ_{ts} for every τ_p refresh period. We explain validation of the algorithm in Section 5.3.6.

5.3.5 Ordering APs for Scheduling

The time delay between when an IP packet is placed on the head of the hardware queue of an IER outgoing port and when the last bit of the last IEEE 802.11 frame for this IP packet is transmitted on air towards the destination station varies a lot depending on many factors. These include wired network delays, router processing delays, and medium access delays. Therefore, serializing wired transmissions out of IER with δ_t-long time-slots does not guarantee

Algorithm 3 $order\_APs()$ [Finding AP Scheduling Order]

Require: $RSSI(a, s), \alpha$

for each $s_l \in \mathbb{S}_l$ **do**
  for $\forall a_p, a_q \in \mathbb{A}_k$ **do**
    if $R_p > 0$ and $R_q > 0$ **then**
      $D_l(p, q) = |R_p - R_q|$
    else if either $R_p = 0$ or $R_q = 0$ **then**
      $D_l(p, q) = 0$
    else
      $D_l(p, q) = \emptyset$
    end if
  end for
end for
$\overline{D} = E_l[D_l]$
for each $a_p \in \mathbb{A}_k$ **do**
  $\overline{D}(a_{p-1}, a_p) = \overline{D}(a_{p-2}, a_p) = 0$
  for $i = 1 : N'$ **do**
    $j = \mathrm{argmax}_j[\alpha\overline{D}(a_j, a_{p+i-1}) + (1 - \alpha)\overline{D}(a_j, a_{p+i-2})]$
    $a_{p+i} \leftarrow a_j$
  end for
  $Q_p = \{a_p, a_{p+1}, \ldots, a_{p+N'-1}\}$
end for
$p^{max} = \mathrm{argmax}_p[distanceTraveled(Q_p)]$
return $Q_p max$

non-overlapping wireless transmissions from the scheduled APs. In order to minimize the interference due to these out-of-schedule transmissions, HeedNet opportunistically orders APs on the scheduling list based on their interference potential to each other.

Construction of the scheduling list is carried out via a search heuristic. The search process is effective but not necessarily always optimum, and is shown in Algorithm 3. HeedNet agent on IER first calculates a signal space distance matrix D_l for each of the reporting stations s_l. Elements of this $N'xN'$ symmetric matrix are absolute RSSI differentials for AP pairs a_p and a_q as observed by s_l. Specifically for $RSSI(a_p, s_l) \equiv R_p$, and $RSSI(a_q, s_l) \equiv R_q$, the distance matrix element at p^{th} row and q^{th} column is calculated as

$$D_l(p, q) = \begin{cases} |R_p - R_q| & \text{if both } R_p > 0 \text{ and } R_q > 0 \\ 0 & \text{if either } R_p = 0 \text{ or } R_q = 0 \\ \emptyset & \text{if both } R_p = 0 \text{ and } R_q = 0 \end{cases}$$

and $\emptyset$ elements are not used in further calculations. HeedNet then averages M' of these D_l

matrices, as $\overline{D} = E[D_l]$. Thus, $\overline{D}$ holds the average signal space distances between all observed AP pairs. HeedNet agent uses $\overline{D}$ to extract the sequence of APs, starting with a_p and maximizing the inter-AP signal-space distance at each step

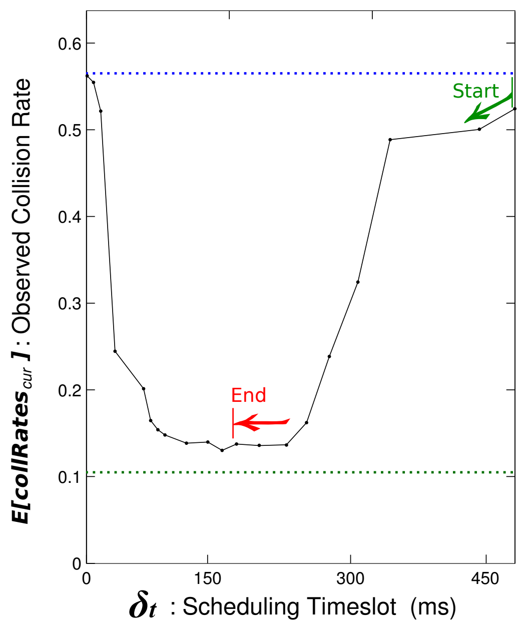
$$Q_p = \{a_p, a_{p+1}, \ldots, a_{p+N'-1}\}$$

Each Q_p sequence orders all N' APs. The objective is to schedule the most contrasting APs signal-strength wise back-to-back, in order to alleviate the effect of out-of-schedule transmissions. Depending on the schedule time-slot δ_t, transmissions may even spill-over more than one time-slot. For this reason, HeedNet uses a composite distance discriminant at each step, factoring in both the distance to the previous AP as well as the AP before the previous. Therefore, an AP a_j is selected as the $(p+i)^{th}$ AP of Q_p so that, $j = \mathrm{argmax}_j[\,\alpha\,\overline{D}(a_j, a_{p+i-1}) + (1-\alpha)\,\overline{D}(a_j, a_{p+i-2})\,]$, where initial distances are assumed to be $\overline{D}(a_{p-1}, a_p) = \overline{D}(a_{p-2}, a_p) = 0$. The history constant $0 \leq \alpha \leq 1$ determines the significance of the spill-overs in the decision process. Once all such Q_p sequences are calculated, one for each of the N' APs in $\mathbb{A}_k$ as starter, HeedNet picks the sequence that travels the maximum cumulative distance in terms of the signal-strength and adopts this as the current scheduling order of the APs.

5.3.6 Evaluation of Algorithms

Before proceeding with the presentation of the overall HeedNet system performance in recovering the performance loss in high-density deployments, we evaluate the algorithms outlined in the design section. Specifically, we ensure that the algorithms for determining the right scheduling time-slot, and finding the scheduling order, perform well according to their definition.

Evaluation of the $pick\_interval()$ algorithm uses the LSA scenario described in Section 5.2.2. First, we conduct simulation experiments with HeedNet scheduling enabled for the LSA scenario, using a number of predetermined timeslot values (δ_t) selected from $[0, 500ms]$ interval. Measured collision rates for these δ_t values are plotted in Figure 5.8(a) and exhibit a parabolic behavior. In the same figure, we also show collision rates for the single WLAN baseline scenario (lower dotted-line) and for the LSA scenario with no HeedNet scheduling (upper dotted-line). Then, experiment is repeated one more time with the $pick\_interval()$ algorithm enabled, in order to see if the algorithm actually settles on to a desired value in the timeslot search space. With the algorithm parameters given in Figure 5.8(b), we observed convergence to a δ_t value



(a) Gradient Search-space

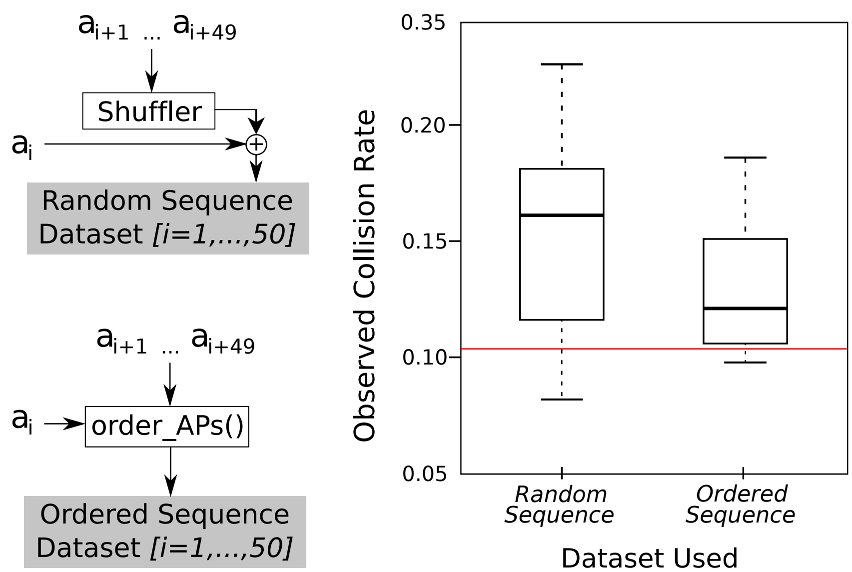
Parameter	Value
N'	50
Δ	¿ 25 s
δ_{ts}	100 ms
τ_s	100 ms
Y	50
Iterations	9
Completion	49 sec
Final δ_t	161ms

(b) Algorithm Parameters

Figure 5.8: Evaluation of $pick\_interval()$

of 161ms in 49s with 9 iterations. From Figure 5.8(a), we verify that this δ_t value falls in the desired-range of the timeslot values that reduce the observed average collision rate close to those of the single WLAN scenario. We have also tested the MSA scenario and found similar collision rate behavior and observed similar performance from the algorithm.

In order to evaluate our AP ordering scheme given in the $order\_APs()$ algorithm, we use the LSA scenario and its 50 interfering APs to construct two different AP sequence datasets: *random sequence* and *best sequence*. Both datasets have 50 different sequences in them, and each sequence starts with a distinct AP from the list of to-be-scheduled APs. In the sequences of the *random sequence* dataset, the 49 APs to follow the first one are selected randomly. In the *best sequence* dataset, the sequence to follow the first one is obtained using the $order\_APs()$ algorithm, as described in Section 5.3.5. History constant, α, is used as 0.8. Creation of these two datasets are illustrated in Figure 5.9(a). We then run one simulation experiment per sequence, thus 50 per dataset, and report our collision rate findings in Figure 5.9(b) for each dataset. The boxplot in the figure illustrates sample minimum/maximum (the handles), lower/upper quartiles (the box), and the sample median (the line in the box). We observe that HeedNet's median collision rate reduces a further 5%, when APs are scheduled according to the $order\_APs()$ algorithm, compared to a random selection of APs. Also, deviations from the



(a) Dataset Generation (b) Collision Performance

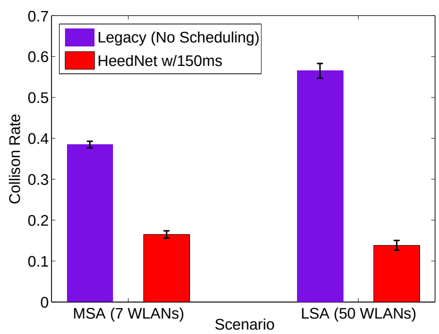
Figure 5.9: Evaluation of $order\_APs()$

mean are less significant (i.e., more consistent performance and less dependence on a particular AP to start the sequence) when APs are ordered by HeedNet's $order\_APs()$ algorithm. The straight line in the figure indicates the average collision rate observed from the experiments with the single WLAN baseline scenario.

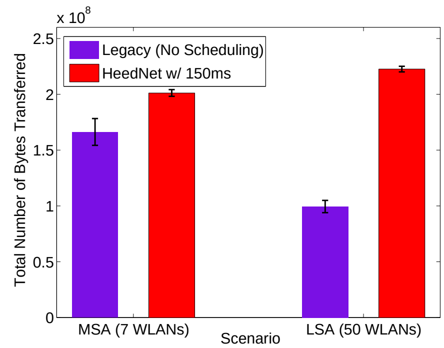
5.4 Performance Evaluation

In this section, we evaluate the performance of a HeedNet adopting network by answering the most relevant questions in the following subsections and by providing experimental data to support the answers. In particular, we are interested in whether HeedNet can substantially recover the lost throughput in high-density deployments, and how fairness and delay behaves while HeedNet tries to recover the lost throughput.

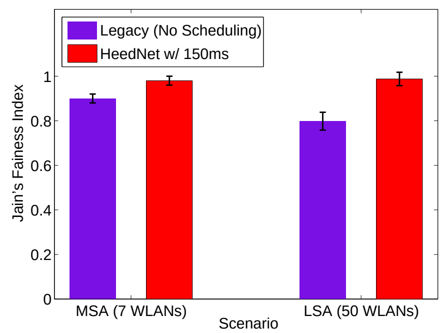
For the evaluations in this section, we have re-used the MSA and LSA scenarios described in Section 5.2.2. We have integrated the HeedNet implementation into the QualNet simulator as a software module, residing in the Network Layer (IP) processing chain. Unless otherwise noted, two-minute experiments are repeated at least ten times with different seeds and average values are reported together with 95-percentiles. For all experiments, we assume that the interferer selection algorithm in Section 5.3.3 is already run and the list of APs to schedule include all 7 and 50 APs in MSA and LSA scenarios, respectively.



(a) Collision Rate



(b) System Capacity



(c) Jain's Fairness Index

Figure 5.10: HeedNet's Performance on High-Density Residential WLAN Interference Management. Errorbars indicate 95-percentiles of the resultsets.

5.4.1 Recovering Lost Performance

We have demonstrated significant performance loss for a high-density WLAN deployment in Section 5.2.4. For the same two scenarios, we have run experiments with HeedNet deployed and summarize the results in Figure 5.10.

From Figure 5.10(a), we observe that HeedNet successfully brings the collision rates down, very close to the levels of a single WLAN baseline case. For MSA scenario, half of the collisions were avoided, and for the LSA scenario, three quarters of the collisions were avoided by using HeedNet. This reduction recovers the lost system capacity substantially, shown in Figure 5.10(b) by the total number of bytes served from the ISP to the wireless clients. Specifically, with HeedNet, the system throughput increases up to 1.3 and 2.2 times of the legacy system for MSA and LSA scenarios, respectively. This increase corresponds to 71% of the loss being recovered for the MSA scenario, when compared to the single WLAN capacity baseline reported in Section 5.2.4. We attribute the remaining 29% non-recovered capacity to imperfect

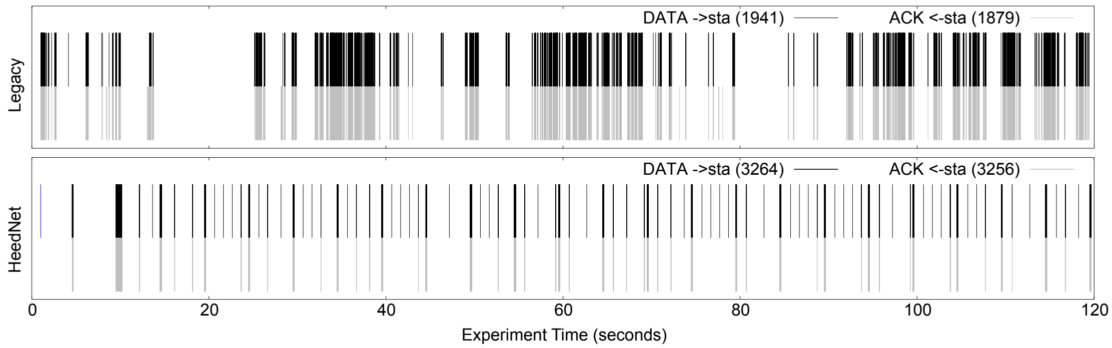


Figure 5.11: HeedNet Under Microscope: Behavior of a TCP flow in Legacy (top) and HeedNet (bottom) system.

serialization of frame transmissions out of APs due to ISP-to-AP and AP-to-air delay jitter, as well as the overheads incurred in heuristics used to find scheduling order and slot duration.

We also take a closer look at how individual TCP flows change their behavior under HeedNet. For this purpose, in Figure 5.11, we plot TCP data and ACK segment exchanges between a randomly selected station in the LSA scenario and the Internet server it receives the content from. Top plot in the figure is from the legacy network, whereas the bottom plot is observed when HeedNet is enabled. Within each plot, black bars (top portion) are drawn per data segment in the downlink direction and gray bars (bottom portion) are drawn per ACK in the uplink direction. The numbers in figure legend show the total number of exchanged segments, and agree with the system capacity gains described above. We observe that without HeedNet, there is an inconsistent rate of flow with starving periods up to several seconds. Thinner black bars in the bottom plot indicates that HeedNet scheduling regulates data flow much more consistently, and allows higher-volume periodic bursts, resulting in a larger amount of data exchange. We have observed similar behavior with other randomly selected source-destination pairs.

5.4.2 Effects on Fairness

We have analyzed the simulation experiment traces to see if HeedNet scheduling degraded fairness that was provided to each WLAN in terms of the number of bytes served to each subscriber. With HeedNet, the ISP performed better in providing fairness to its WLANs, as illustrated in Figure 5.10(c). For both scenarios, HeedNet allows the system to distribute to each AP very close to a perfectly fair share, indicated by a Jain's Fairness Index value of 1.

5.4.3 Delay Behavior with HeedNet

HeedNet improves system capacity by doing selective buffering at ISP, therefore, it is vital to check if delay characteristics of data flows are negatively affected. We analyzed the inter-packet gaps for the TCP data segments, as they arrive at the stations of the LSA scenario, and plotted CDF for the delay in Figure 5.12. Although there is no scheduling for the legacy network, the median time between two data segments are much larger (i.e., 10.19ms) compared to the HeedNet enabled network (i.e., 0.78ms). Only the segments at scheduling boundaries have larger delay in HeedNet, which is dependent on the scheduling timeslot and the number of APs in the scheduling list.

We also inspect the distribution of delay to complete transmission of various size files with and without HeedNet. Files in 10, 20, 50, and 100 KBytes sizes are transmitted back-to-back from Internet servers to the clients as we record completion times. The statistics from the experiments on the LSA scenario are given in Figure 5.13. We observe that file transfers with HeedNet take about one-half of what legacy network can do, independent of the file size chosen. Also, HeedNet reduces the variation of the transfer time (i.e., standard deviation). From CDF plots for different file sizes, the effect of the HeedNet scheduling time slot and the number of APs to schedule can be seen at around 5 seconds, for file sizes larger than 20 KBytes. Nonetheless, the legacy network exhibits heavier distribution tails for those file sizes, making HeedNet still a better alternative for high-density deployments. The total number of files completed in experiments for each case are given in CDF plot legends, which agree with earlier system capacity improvement figures.

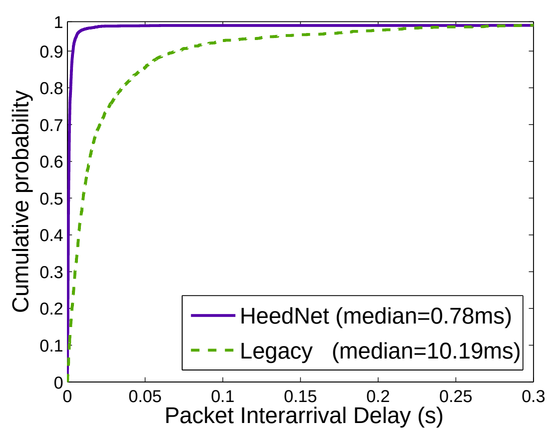


Figure 5.12: Distribution of delay between packet arrivals at the STA

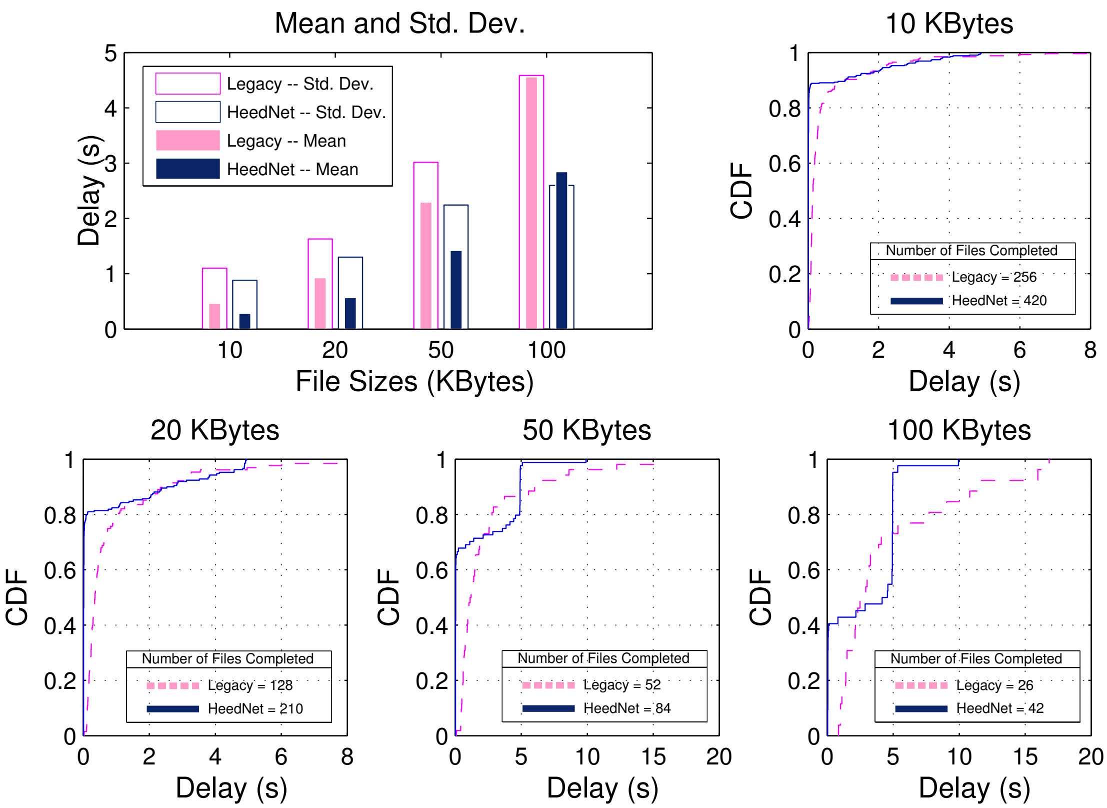


Figure 5.13: Delay behavior for different file-sizes.

5.4.4 Buffering Requirements

We have evaluated HeedNet in terms of the buffer space requirements for the IER, to be able to store IP packets temporarily during scheduling. For our LSA scenario, we have probed the occupancy of the network layer buffer of in our QualNet simulator implementation, and observed it during a minute of the simulation. We observe, from Figure 5.14, that even for a network with 50 interfering APs and saturated traffic, HeedNet's maximum buffer occupancy never exceeds 250KBytes, a RAM space of which is practically available in almost all of telco-grade routers today. We also found out that the average buffer space occupancy is around 25KBytes for the scenario tested.

5.5 Implementation and Deployment

We have demonstrated the viability of $HeedNet$ for recovering interference-related performance losses with the experiments reported in the previous section. For exploring $HeedNet$'s performance on a real deployment, we have extended the ORBIT wireless testbed [22], and

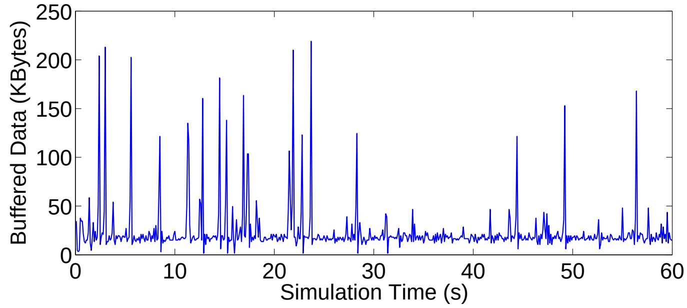


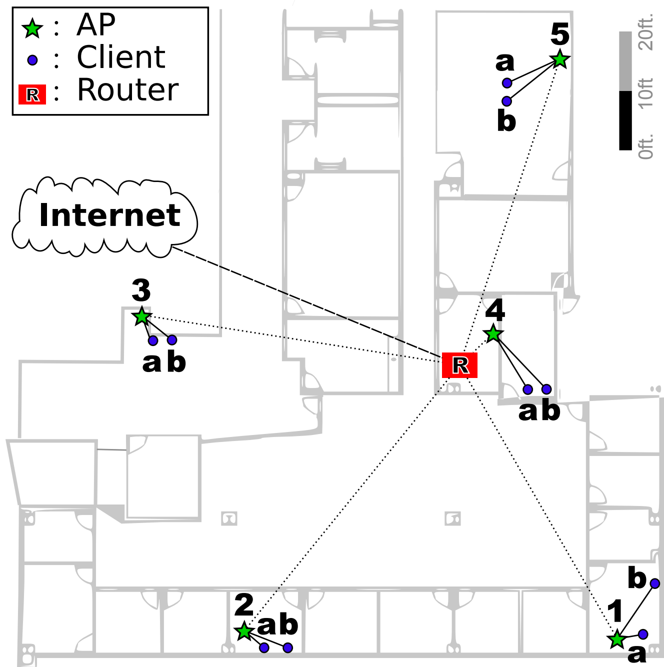
Figure 5.14: Buffering requirement on IER

implemented $HeedNet$ on the testbed using a programmable router, typical AP devices and applications seen on today's residential network clients.

5.5.1 Testbed and Design

Our testbed uses the ORBIT outdoor grid infrastructure and utilizes five IEEE 802.11g consumer APs and ten wireless clients (i.e., two for each AP), all served by a programmable Click router [63] for the Internet access. The APs are connected to the router via their 100BaseT Ethernet ports and operate on a quiet channel of 2.4GHz ISM Band. The testbed is located in Rutgers University WINLAB facility, an indoor office space of approximately 10,000 sq. feet. The placement of the nodes on the floorplan and the device specifications are given in Figure 5.15.

We have emulated the ISP edge router by a general purpose rack-server with multiple 1 Gbps Ethernet ports. The server is installed with Ubuntu Linux version 9.10 and Click modular router software version 1.7.0rc. As shown in Figure 5.16, the router is connected to the Internet on one port and to the LAN switch, with all of the APs, on another port. These two ports are exclusively managed by the Click kernel module and the packets to and from these ports are not handed up to the Linux networking stack. HeedNet is implemented as a Click script that maintain one queue per-destination AP. The script *paints* incoming packets from the Internet interface, based on the destination AP and source port (i.e. application). Painting is Click's internal tagging feature, which is used by HeedNet to allow certain APs and applications to bypass scheduling, as well as to implement the right AP-to-queue mapping. The applications to bypass scheduling are read from *configure-time parameters* file. The APs to bypass scheduling are read from *run-time parameters* list, updated by the interferer selection algorithm described



ID	Equipment
$AP1$	Cisco 1230abg
$AP2$	Dlink DIR-655bgn
$AP3$	Dlink DI-774abg
$AP4$	Linksys WRT55ag
$AP5$	Netgear DG-834
$Clients$	ORBIT Compact PC Nodes with Atheros 5212abg WLAN Card
$Router$	SuperMicro Rack Server with Dual Xeon 2.66GHz, 1GB RAM, multiple 1Gbps network devices, installed with Ubuntu 9.10 and Click 1.7.0rc (kernel module driver)

Figure 5.15: Testbed Deployment for HeedNet

in Section 5.3.3. Packets marked with *to-be-scheduled* colors are filled into respective queues according to their destination APs. Each queue is set-up to hold a maximum of 200 MTU-size IP packets, based on the buffer-size observation made in Section 5.4.4. An efficient timer-driven switch selects to serve only one queue at a time, and moves onto the next one after a timeslot amount of time. The timeslot δ_t and the order in which the queues are served are read from *run-time parameters* list, updated by the algorithms described in Sections 5.3.4 and 5.3.5. The output of the timer-driven switch feeds the packets directly onto the outgoing path of the interface on the local network. In the uplink direction, from local network to the Internet, packets are routed directly with no painting or extra queuing involved.

The clients are ORBIT testbed's compact PC nodes installed with Ubuntu Linux version 9.04. Due to the placement of APs and clients in the testbed, interfering signal strength is at considerable levels at all APs, thus the client agent software is inactive and we assume all APs

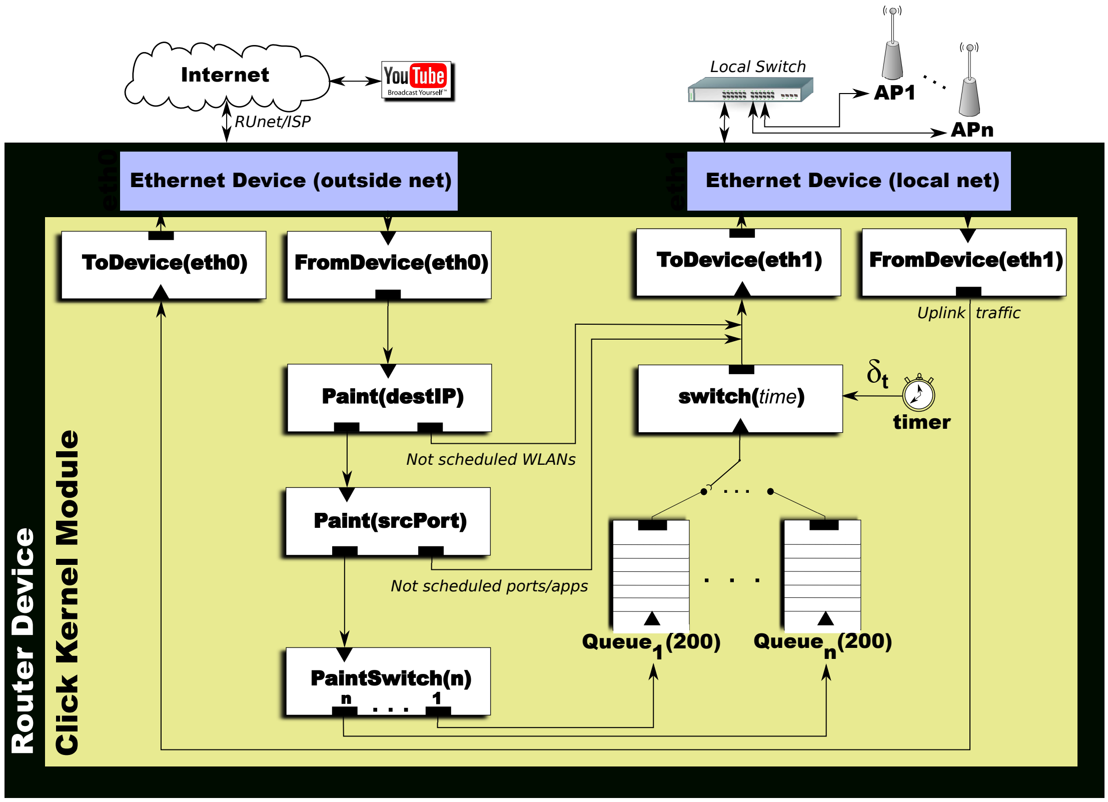
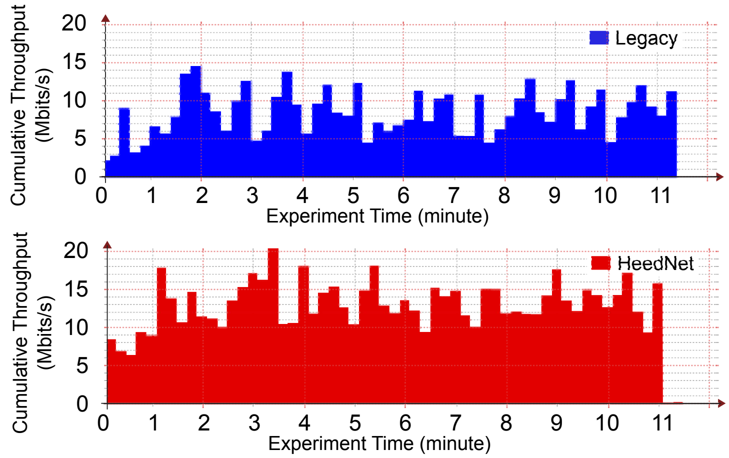


Figure 5.16: HeedNet implementation on the router using Click

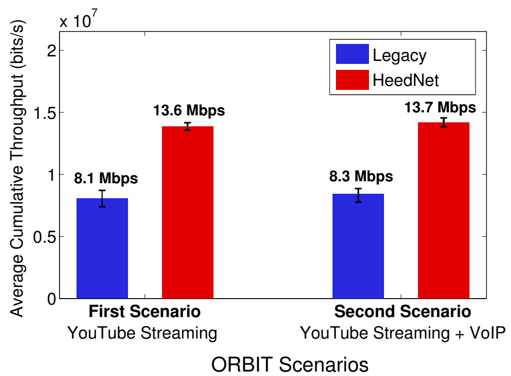
should be scheduled at all times. The AP devices are arbitrarily selected from popular consumer devices available in the market. There are no changes in the hardware or the software of the APs and they are used with the default settings except for SSID and channel.

5.5.2 Scenarios Tested and Results

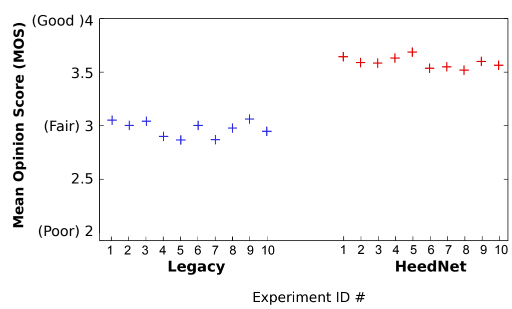
We use a very popular and communication-intensive application for testing HeedNet on our testbed: *YouTube* streaming. Having ranked third most visited website in the world with two billion views per day, we believe YouTube represents today's residential WLAN workload the best [64]. Originally, YouTube streamed clips encoded in Flash Video format with an average rate of 200Kbps, whereas today, users can receive a variety of quality levels, including 1080p HD videos with an average rate over 3500Kbps [65]. When combined with aggressive buffering for a smooth playout buffer, not more than a few users are capable of saturating a residential WLAN deployment with YouTube streaming. We are also interested in exploring how performance of VoIP communications is affected on a HeedNet deployed WLAN with multiple clients streaming YouTube video clips.



(a) Cumulative throughput recorded during the experiment (12 sec. averages)



(b) Average cumulative throughput for both scenarios on Legacy and HeedNet



(c) VoIP MOS on Legacy and HeedNet

Figure 5.17: Results from the scenarios tested on the ORBIT testbed.

In our tests, all clients fire-up their browsers with the start of the experiment, and try to watch the video clip of the *State of the Union Address 2010* at 720p encoding with an average encoding rate of 2000Kbps. We observe that YouTube's Adobe Flash-based player application would try buffering well beyond average video rate to maintain a sufficiently-filled playout buffer, whenever necessary. For the first scenario we tested, we observe cumulative system throughput for a period of 10 minutes and report averages with legacy and HeedNet enabled configurations. For the second scenario, we initiate a UDP-based G.711 VoIP call between one of the wireless clients and an external client on the Internet, while the same YouTube streaming session is taking place on all clients. The VoIP traffic is not subject to scheduling by HeedNet, and we report on how subjective mean opinion score (MOS) for the call audio quality changes between legacy and HeedNet enabled tests. The results are summarized in Figure 5.17.

We have repeated tests with both scenarios ten times and calculated the average and standard deviation before reporting. Figure 5.17(a) depicts the throughput vs. time behavior from one of the runs of the first scenario, with and without HeedNet. In the plot, cumulative system throughput measurements are averaged for every 12sec for the first 10 min period of the experiment. Communication demands from the clients are observed to be very dynamic due to YouTube's buffering behavior and HeedNet increases the amount of system capacity available to address this demand of streaming clients, from a maximum of 14.5 Mbits/s to a maximum of 21 Mbits/s at certain intervals. The average capacity improvement, as shown with the left-side bars on Figure 5.17(b), is on the order of 68%. The capacity improvements are similar for the second scenario, when VoIP call was added, as throughput demand for the VoIP session is quite small and around 64Kbits/s. However, due to reduced-collision rates with the use of HeedNet, the VoIP call quality significantly improves when HeedNet is enabled. From Figure 5.17(c), we observe that most *fair* grade calls get upgraded to *good* grade, when HeedNet is used for the rest of the high-volume streaming traffic.

5.6 Discussion and Limitations

In this section, we discuss some of the factors that might potentially limit utility of HeedNet and point towards possible solutions and future work needed to address those shortcomings.

Heterogeneity of the residential deployment environment would be one factor that could reduce the gains that HeedNet can harvest. For example, a set of interfering APs being served by more than one ISP requires will for cooperation and explicit mechanisms for coordination on those ISPs' side. Such cooperation is mutually beneficial, and if agreed upon by the ISPs, an extended version of HeedNet can be designed and deployed for multiple ISPs with the help of interfering AP-list message exchanges and scheduling space partitioning in time domain. In fact, this is an area we are planning to work on in the future. Heterogeneity within the management domain of a single ISP, i.e., interfering APs served by multiple edge routers of the same ISP, is a much easier problem to solve. Low-delay fast backbone links between edge routers can allow a single HeedNet agent to span across multiple IERs with the help of inter-router message exchanges. Finally, some APs may never be managed by HeedNet due to a variety of reasons, like some ISPs unwilling to cooperate, or an enterprise AP in the same environment etc. For those cases, HeedNet can still improve the system performance to a certain extent, dictated by the ratio of such unmanaged APs to the APs under HeedNet management.

Another factor to consider for HeedNet's success would be the amount of participation from the clients of the ISP. HeedNet relies on its client agent software for selecting the right set of interfering APs and the appropriate scheduling timeslot. Although not all clients are required to participate, too low of an adoption would cause suboptimal scheduling to be enforced on APs. Our preliminary tests indicate that HeedNet scheduling parameters change only by a small amount when participation from clients are forced to drop from 100% to 50%, and by a considerable amount when the participation falls below 25%. Nonetheless, many subscribers use ISP provided software install kits for the ease of initial set-up and HeedNet can easily be bundled with those software kits. Also, subscribers are usually open to opt in for later add-ons as long as they are free and made to improve their Internet experience.

Determining which protocols and applications should be scheduled by HeedNet would also need careful consideration. Delay intolerant applications that do not contribute to the overall traffic volume, such as VoIP calls or DNS queries, should be always allowed to bypass HeedNet scheduling. HTTP/FTP downloads, non-interactive video streaming, and torrent downloads are good candidates for HeedNet scheduling as they constitute the bulk of the traffic on the ISP network. Beyond static, and port based application identification, HeedNet needs to be augmented with stateful packet inspection type classifiers to be used correctly on proxy-based bulk traffic as well as for other types of new usage models that consume a lot of bandwidth. We have been experimenting with a HeedNet prototype that can bypass TCP control traffic (i.e., for handshaking) and observed that occasional TCP connection establishment timeouts could be prevented with this kind of a content-aware policy decision.

Chapter 6

Conclusions

6.1 Conclusion

Ever growing deployment of residential WLANs have resulted in very high levels of co-channel interference in dense settings like apartment buildings. The extent of performance degradation due to such uncontrolled interference has not been studied thoroughly. In this thesis, we have presented a systematic analysis of the effect of inter-cell interference in such unplanned, high density WLANs through detailed experiments and simulations. Our work complements previous real-world measurements through experimentation with hundreds of IEEE 802.11 enabled nodes in repeatable settings with controlled interference. Thus, it allows in-depth analysis through simulations and repeatable experiments, with precisely known configurations. The insights we have presented on the relation between the number of APs and the interference-driven collision enhances the existing IEEE 802.11 based performance models and allow for performance modeling for high density WLAN deployments. We then have proposed two alternative solutions that can recover most of the losses demonstrated in this thesis. The following paragraphs explain these contributions in more detail.

We have analyzed high-density Wi-Fi system performance using a realistic TCP dominated workload in unplanned multi-cell WLANs by conducting experiments on ORBIT testbed [22] as well as QualNet simulator [23]. Results show that a single-cell network remains remarkably robust even with 125+ clients; the collision rate remains low. We have also shown that, in an unplanned multi-cell network, however, the collision rate increases significantly.

Attributed to to TCP flow control, the number of backlogged stations equals twice the number of active access points, meaning that network efficiency is determined by *the number of interfering access points*, not *the number of clients*. In addition, we have shown that TCP can not regulate the flows in the IEEE 802.11 network for optimal system operating point (i.e. max.

throughput) across different contention window settings. Even with Wireless Multimedia Extensions based on the IEEE 802.11e standard [24], VoIP users could still experience substantial performance degradation in unplanned deployments. This deterioration starts to occur even when the number of interfering APs is *relatively small* (three). Video streaming in the network makes the system performance worse for VoIP users.

As our first solution, we have proposed $WiPhi$ as a practical distributed interference mitigation technique. $WiPhi$ is a contention window adaptation method based on the number of active access points, not the number of clients in the network. We have shown that $WiPhi$ could recover the lost throughput by its effective MAC layer contention window adaptation approach. We have also shown that an additional 20% gain would be possible with collision-resilient rate adaptation.

We have then designed realistic residential high-density WLAN scenarios for performance evaluations with medium and large-sized apartment buildings, based on recent metro-area apartment surveys. We have introduced $HeedNet$ as an alternative to high-density residential interference management. $HeedNet$ provides ISPs with a powerful and easy-to-deploy interference management tool that is effective in recovering lost system capacity due to the increasing number of APs operating and interfering on the limited number of available WLAN channels. Bulk traffic towards the interfering APs is scheduled by $HeedNet$, giving more exclusive and deterministic access to channel for each AP, reducing the amount of collisions due to overlapping frame transmissions. $HeedNet$ algorithms could be deployed on ISP edge routers, with no changes required on the APs at subscriber premises or the IEEE 802.11 MAC protocol itself. $HeedNet$ design aimed efficiency with simple algorithms considering deployment feasibility. Our simulation experiments on these scenarios have revealed that use of $HeedNet$ can improve system capacity up to 2.2X, while giving a more fair share of capacity to each WLAN subscriber. We further have implemented $HeedNet$ on actual devices and deployed a testbed to validate our findings from the simulations. Our findings proved that gains observed from the use of $HeedNet$ are practical and deployment of $HeedNet$ by ISPs is feasible. We have also found out that $HeedNet$ is beneficial in improving performance of non-scheduled traffic, such as VoIP, due to the reduced collision-rate environment it provides.

References

[1] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Proc. ACM SIGMETRICS*, July 1998, pp. 151–160.

[2] "62. IETF Meeting Traces at CRAWDAD," http://crawdad.cs.dartmouth.edu/ietf2005.

[3] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *Proc. ACM MOBICOM*, 2005, pp. 185–199.

[4] Cisco Inc., "20 Myths of Wi-Fi Interference: Dispel Myths to Gain High- Performing and Reliable Wireless (Whitepaper)," 2007.

[5] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless LANs," in *Proc. IEEE INFOCOM*, April 2006, pp. 1–12.

[6] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *Proc. ACM MOBICOM*, 2004, pp. 315–329.

[7] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for wlans," *ACM SIGMOBILE MC2R*, vol. 9, no. 3, pp. 19–31, 2005.

[8] "Aruba Networks, Inc." http://www.arubanetworks.com.

[9] "Cisco Systems, Inc." http://www.cisco.com.

[10] "Trapeze Networks," http://www.trapezenetworks.com.

[11] A. Mishra, V. Shrivastava, D. Agarwal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. ACM MOBICOM*, September 2006, pp. 170–181.

[12] M. A. Ergin, K. Ramachandran, and M. Gruteser, "An Experimental Study of Inter-cell Interference Effects on System Performance in Unplanned Wireless LAN Deployments," *Computer Networks (Elsevier)*, vol. 52, pp. 2728–2744, October 2008.

[13] J. Zhu, B. Metzler, X. Guo, and Y. Liu, "Adaptive csma for scalable network capacity in high-density WLAN: A hardware prototyping approach," in *Proc. IEEE INFOCOM*, April 2006, pp. 1–10.

[14] T.-S. Kim, J. C. Hou, and H. Lim, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *Proc. ACM MOBICOM*, September 2006, pp. 366–377.

[15] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE JSAC*, vol. 18, no. 3, pp. 535–547, May 2000.

[16] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. on Info. Theory*, vol. 46, pp. 388–404, 2000.

[17] S. Choi, K. Park, and C. Kim, "On the performance characteristics of WLANs: revisited," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 97–108, 2005.

[18] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc of ACM MOBICOM*, September 2004, pp. 187–201.

[19] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," in *Proc. SIGMETRICS*, 2002, pp. 195–205. [Online]. Available: http://doi.acm.org/10.1145/511334.511359

[20] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in IEEE 802.11b wireless networks," in *Proc. USENIX IMC*, October 2005, pp. 279–292.

[21] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth, "IQU: Practical queue-based user association management for WLANs," in *Proc. ACM MOBICOM*, September 2006, pp. 158–169. [Online]. Available: http://doi.acm.org/10.1145/1161089.1161108

[22] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. IEEE WCNC*, vol. 3, March 2005, pp. 1664–1669.

[23] Scalable Network Technologies, "Qualnet v.3.9.5 user's guide," http://www.qualnet.com, May 2006.

[24] IEEE, "Standard Document," *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, 2005.

[25] R. Bruno, M. Conti, and E. Gregori, "Performance modelling and measurements of tcp transfer throughput in 802.11-based WLAN," in *Proc. ACM MSWiM*, 2006, pp. 4–11.

[26] Y. Gong and P. Marbach, "Interaction of rate and medium access control in wireless networks:: the single cell case," in *Proc. ACM MOBIHOC*, 2006, pp. 178–189.

[27] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over Wireless LAN," in *Proc. IEEE INFOCOM*, 2003, pp. 863–872.

[28] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, 2000.

[29] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs," in *Proc. of ACM SIGCOMM*, 2005, pp. 121–132.

[30] C. Hu and J. C. Hou, "A novel approach to contention control in IEEE 802.11e-operated WLANs," Univ. of Illinois at Urbana-Champaign, Tech. Rep. UIUCDCS-R-2005-2600, August 2006. [Online]. Available: http://tinyurl.com/yvkl6z

[31] A. Vasan, R. Ramjee, and T. Y. C. Woo, “ECHOS - enhanced capacity 802.11 hotspots.” in *Proc. IEEE INFOCOM*, 2005, pp. 1562–1572.

[32] F. Guo and T. cker Chiueh, “Scalable and robust WLAN connectivity using access point array,” in *Proc. DSN 2005*, June 2005, pp. 288–297.

[33] A. Kamerman and L. Monteban, “WaveLAN-II: A high-performance wireless LAN for the unlicensed band,” *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[34] M. Lacage, M. H. Manshaei, and T. Turletti, “IEEE 802.11 rate adaptation: a practical approach,” in *Proc. of ACM MSWiM*, 2004, pp. 126–134.

[35] MADWiFI, “RSSI in MadWifi,” http://madwifi.org/wiki/UserDocs/RSSI.

[36] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-hop wireless networks,” in *Proc. of ACM MOBICOM*, Rome, Italy, 2001.

[37] J. C. Bicket, “Bit-rate selection in wireless networks,” Master’s thesis, M.I.T., Cambridge, MA, February 2005.

[38] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “Opportunistic media sccess for multirate ad hoc networks,” Proc. of ACM MOBICOM, 2002, pp. 24–35.

[39] M. Andrews, “A survey of scheduling theory in wireless data networks,” *IMA summer workshop on wireless comm.*, 2005.

[40] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless lan,” in *Proc. of the 6th annual international conference on Mobile computing and networking*, USA, August 2000.

[41] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, “Centaur: realizing the full potential of centralized wlans through a hybrid data path,” in *Proc. International Conference on Mobile Computing and Networking*, 2009.

[42] White-paper from Meru Networks, “Microcell deployments: Making a bad problem worse for pervasive wireless lan deployments,” http://www.merunetworks.com/pdf/whitepapers.

[43] E. P. Evaluation, S. Choi, J. Prado, S. Shankar, and N. S. Mangold, “Ieee 802.11e contention-based channel access,” pp. 1151–1156, 2003.

[44] MADWiFi, *Multiband Atheros Driver for WiFi*. http://madwifi.org, 2007.

[45] “Acess Point Density Traces rutgers/ap_density at CRAWDAD,” http://crawdad.cs.dartmouth.edu/rutgers/ap_density, 2007.

[46] M. Takai, J. Martin, and R. Bagrodia, “Effects of wireless physical layer modeling in mobile ad hoc networks,” in *Proc. ACM MOBIHOC*, 2001, pp. 87–94.

[47] H. Chang, V. Misra, and D. Rubenstein, “A general model and analysis of physical layer capture in 802.11 networks,” in *Proc. IEEE INFOCOM*, April 2006, pp. 1–12.

[48] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” DEC TR-301, Tech. Rep., September 1984.

[49] "GNU wget file retrieval tool," http://www.gnu.org/software/wget.

[50] A. Botta, A. Dainotti, and A. Pescape, "Multi-protocol and multi-platform traffic generation and measurement," in *IEEE INFOCOM 2007 Demo Session*, May 2007.

[51] Y. Xiao and J. Rosdahl, "Throughput and delay limits of ieee 802.11," *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, August 2002.

[52] "ITU-T Rec. G.114 One-way Transmission Time," http://tinyurl.com/2z4co5.

[53] "Cisco Systems Inc, White Paper – A Clear Understanding: What Makes Voice Over IP Work?" http://tinyurl.com/3jko5, 2002.

[54] IEEE , "Std. 802.11h: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 5: Spectrum and transmit power management extensions in the 5 GHz band in europe," 2003.

[55] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006, pp. 1–11.

[56] K. Ramachandran, H. Kremo, M. Gruteser, P. Spasojevic, and I. Seskar, "Scalability analysis of rate adaptation techniques in congested ieee 802.11 networks: An orbit testbed comparative study," in *Proc. of IEEE WoWMoM*, 2007, pp. 1–12.

[57] J. C. Bicket, "Bit-rate selection in wireless networks," Master's thesis, M.I.T., Cambridge, MA, February 2005.

[58] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. ACM MOBICOM*, Sept 2006, pp. 146–157.

[59] K. Ramachandran, H. Kremo, M. Gruteser, P. Spasojevic, and I. Seskar, "Scalability analysis of rate adaptation techniques in congested ieee 802.11 networks: An orbit testbed comparative study," in *Proc. of IEEE WoWMoM*, 2007, pp. 1–12.

[60] "NY Bits - An online rental search tool by Gromco Inc." http://www.nybits.com.

[61] "WiGLE.net - Wireless Geographic Logging Engine," http://wigle.net.

[62] J. A. Snyman, *Practical Mathematical Optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms*. Cambridge, MA, USA: Springer, 2005.

[63] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.

[64] "The Official YouTube Blog: At five years, two billion views per day and counting," http://tinyurl.com/25jkucg.

[65] "YouTube. In Wikipedia, the free encyclopedia." http://tinyurl.com/la4cm.

Curriculum Vita

MESUT ALI ERGIN

EDUCATION

- Ph.D., Electrical and Computer Engineering, October 2010
  Rutgers University, New Brunswick, NJ, USA
- M.S., Computer Engineering, January 2003
  Bogazici University, Istanbul, Turkey
- B.S., Control and Computer Engineering, June 1999
  Istanbul Technical University, Istanbul, Turkey

WORK EXPERIENCE

- Intel Corp., Circuit & Systems Research Lab, Hillsboro, OR, 02/2010 - current
  Research Scientist
- WINLAB, Rutgers University, North Brunswick, NJ, 09/2004 - 02/2010
  Graduate Research Assistant
- Intel Corp., Radio Comm. Lab / CTL, Hillsboro, OR, 05/2006 - 09/2006
  Graduate Intern
- Intel Corp., Radio Comm. Lab / CTL, Santa Clara, CA, 06/2005 - 09/2005
  Graduate Intern
- Yeditepe University, CSE Dept., Istanbul, Turkey, 09/1999 - 08/2004
  Teaching and Research Assistant
- Alcatel Inc., R&D, Istanbul, Turkey, 06/1998 - 06/1999
  Contract Engineer
- A & A Computers Ltd., Istanbul, Turkey, 11/1997 - 09/2002
  Technical Manager

PUBLICATIONS

- C. Dwertmann, M. A. Ergin, G. Jourjon, M. Ott, T. Rakotoarivelo, I. Seskar, and M. Gruteser, “DEMO: Mobile Experiments Made Easy with OMF/Orbit”, *In Proceedings of the ACM SIGCOMM 2009 (Demo Session)*, Spain, August 2009.

- G. Chandrasekaran, M. A. Ergin, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. "DECODE: Exploiting Shadow Fading to DEtect CO-Moving Wireless DEvices", IEEE Transactions on Mobile Computing, Vol.8, Issue 12, pp. 1663–1675, December 2009.

- G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Empirical Evaluation of the Limits on Localization Using Signal Strength", *In Proceedings of the 6th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Network (IEEE SECON 2009)*, pp. 1–9, Rome, Italy, June 2009.

- X. Jing, S. Anandaraman, M. A. Ergin, I. Seskar, and D. Raychaudhuri, "Distributed Coordination Schemes for Multi-Radio Co-existence in Dense Spectrum Environments: An Experimental Study on the ORBIT Testbed", *In Proceedings of the 3rd IEEE International Symposia on Dynamic Spectrum Access Networks (IEEE DySPAN 2008)*, pp. 156–166, Chicago, IL, USA, October 2008.

- M. A. Ergin, K. Ramachandran, M. Gruteser, "An Experimental Study of Inter-cell Interference Effects on System Performance in Unplanned Wireless LAN Deployments", *Computer Networks (Elsevier)*, Vol.52, Issue 14, pp. 2728–2744, October 2008.

- G. Chandrasekaran, M. A. Ergin, R. P. Martin, M. Gruteser, J. Yang, Y. Chen, "DECODE: Detecting Co-Moving Wireless Devices", *In Proceedings of the Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2008)*, pp. 315–320, Atlanta, GA, USA, September 2008.

- M. A. Ergin, M. Gruteser, L. Luo, D. Raychaudhuri, H. Liu, "Available Bandwidth Estimation and Admission Control for QoS Routing in Wireless Mesh Networks", *Computer Communications (Elsevier)*, Vol.31, Issue 7, pp. 1301–1317, May 2008.

- G. Chandrasekaran, M. A. Ergin, M. Gruteser, R. P. Martin, "Bootstrapping a Location Service Through Geocoded Postal Addresses", *In Springer LNCS, Proceedings of the 3rd Intl. Symposium on Location and Context-Awareness (LoCA, held with UbiComp)*, volume 4718, pp. 1–16, Germany, September 2007.

- M. A. Ergin, K. Ramachandran, M. Gruteser, "Extended Abstract: Understanding the Effect of Access Point Density on Wireless LAN Performance", *In Proceedings of the ACM International Conference on Mobile Computing and Networking (ACM MobiCom 2007)*, pp. 350–353, Montreal, Canada, September, 2007.

- M. A. Ergin, M. Gruteser, "Using Packet Probes for Available Bandwidth Estimation: A Wireless Testbed Experience (Poster)", *In Proceedings of First ACM SIGMOBILE Int. Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization (WiNTECH 2006, held with Mobicom)*, Los Angeles, CA, USA, September 2006.

- S. Baydere, M. A. Ergin, and O. Durmaz, "Constructing Wireless Sensor Networks via Effective Topology Maintenance and Querying", *Proc. of Med-Hoc-Net 2004, 3^{rd} Annual Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, June 2004

- M. A. Ergin, "A Cross Layer Protocol for Service Access in Mobile Ad Hoc Networks", M. S. Thesis, *Bogazici University*, 2003

- E. Ozcan, A. Alkan, S. Demir, M. A. Ergin, H. Kul, and S. E. Seker, “STARS - Student Transcript, Administration and Registration System, an Open Source Internet Application”, *Fifth National Academic Information Technologies Conference*, e-paper ref. 87, Turkey, 2003

- S. Baydere and M. A. Ergin, “An Architecture for Service Access in Mobile Ad Hoc Networks”, *Proceedings of the IASTED Wireless and Optical Communications*, pp. 392–397, Banff, Canada, July 2002

- S. Baydere and M. A. Ergin, “A Model for Dynamic Service Discovery in Wireless Ad Hoc Networks”, *Proceedings of the Sixth Symposium on Computer Networks*, pp. 120–128, Gazimagusa, Cyprus, June 2001