Understanding and Improving Performance in Next-Generation WLANs and Cellular Networks

A Dissertation Presented

by

Fatima Zarinni

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

December 2015

Stony Brook University

The Graduate School

Fatima Zarinni

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dr. Michael Ferdman - Dissertation Advisor
Assistant Professor, Department of Computer Science

Dr. I.V. Ramakrishnan - Chairperson of Defense
Professor and Graduate Program Director, Department of Computer Science

Dr. Jie Gao - Committee Member
Associate Professor, Department of Computer Science

Dr. Yanni Ellen Liu - Committee Member
Research Assistant Professor, Department of Computer Science

Dr. Ritesh Maheshwari - External Committee Member
Lead Network Performance Engineer and Researcher, LinkedIn

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Understanding and Improving Performance in Next-Generation WLANs and Cellular Networks

by

Fatima Zarinni

Doctor of Philosophy

in

Computer Science

Stony Brook University

2015

Over the past decade wireless devices, such as smart phones, tablets and laptops have gained tremendous popularity and proliferation worldwide. Additionally, both the WiFi technology and Cellular networks have observed significant advancements from different aspects, with the aim of achieving better network throughput and user experience. Hence, it becomes vital to conduct new, principled studies in such state-of-the-art networks, quantify the true benefits attained, identify previously unknown problems, and propose, develop and evaluate solutions to these problems, that could further improve performance.

To this end, in the first part of this dissertation, we focus on Next-Generation WLANs. The key feature of these networks is that now they are capable of supporting very high physical layer (PHY) data rates (i.e., 1 Gbps and up). While the PHY developments are meant to improve network performance, we find that MAC schemes face new challenges in such settings, and thus, fall short in utilizing the underlying high capacity channel well.

We verify the problem of severe throughput degradation in such networks, when the entire channel is used as a single resource by MAC protocols similar to the standard IEEE 802.11 CSMA/CA. Additionally, we find that some related promising MAC solutions have not been extensively studied before in this context. To address this gap, we thoroughly study these MAC schemes, under a wide range of settings, via both analytical modeling and simulations. We identify new issues that can cause drastic under-utilization of the high speed channel, and/or unfairness, in realistic settings, with these schemes. The insights obtained by our studies motivate the need for developing better, new MAC protocols for high data rate WLANs.

To achieve this goal, we propose and promote the idea of adaptively channelizing the spectrum, and we develop new MAC techniques to enhance performance in emerging high data rate WLANs. We show via extensive simulations, that our MAC schemes can significantly outperform existing schemes, in different network topologies and traffic scenarios, in terms of channel utilization, per-user-throughput and fairness.

In the second part of this dissertation, we focus on an emerging and important, yet unexplored, area in Cellular Networks, Mobile Virtual Network Operators or MVNOs, that operate on top of existing cellular infrastructures. While MVNOs have shown significant growth in the US and elsewhere in the past two years and have been successful in attracting customers, there is no prior systematic study to understand performance in such networks. To the best of our knowledge, we present the first systematic measurement study to shed light on this emerging phenomenon. We design and develop measurement techniques and testbeds with Android smartphones. We take approaches to avoid measurement bias, and we study in detail the performance of 3 key applications: web access, video streaming and voice, in 2 popular MVNO families (a total of 8 carriers) in the US. Interestingly, we find that some MVNOs do indeed exhibit significant performance degradation and that there are key differences between the two MVNO families. Our measurements, analysis and new insights are beneficial to not only end-users, but also to MVNOs, underlying carriers and application developers.

Dedicated to my very kind and caring Parents

Table of Contents

List of Figures **xiii**

Acknowledgements **xviii**

1 Introduction **1**

1.1 Research Problems 3

1.1.1 Channel Utilization Problem in High Data Rate WLANs 3

1.1.2 Understanding Performance in MVNOs 6

1.2 Contributions 7

1.2.1 Exploring Channelization as a Solution for High Data Rate WLANs 7

1.2.2 Studying Channelization in Multiple Collision Domain Settings for High Data Rate WLANs 8

1.2.3 Understanding the FICA PHY/MAC protocol in High Data Rate WLANs 9

1.2.4 btFICA MAC protocol for High Data Rate WLANs . . 10

1.2.5 Conducting Measurement Study in Mobile Virtual Network Operators (MVNOs). 11

1.3 Outline 13

2 Understanding the FICA PHY/MAC Protocol in Next-Generation WLANs and Our btFICA PHY/MAC Solution 15

2.1 Introduction 15

2.2 Description of the FICA Scheme 20

2.2.1 The FICA Physical Layer Architecture 21

2.2.2 The FICA MAC Protocol 22

2.2.2.1 Frequency-domain Contention 22

2.2.2.2 DATA/ACK Phase 23

2.2.2.3 Frequency-Domain Backoff 24

2.2.2.4 Network Allocation Vector (NAV) Band . . . 25

2.3 Performance Issues with the FICA MAC scheme 25

2.3.1 The Deafness Problem 28

2.3.2 The Muteness Problem 31

2.3.3 The Hidden Terminal Problem 33

2.4 btFICA: Busy Tone Assisted Fine-Grained Channel Access Scheme 39

2.4.1 btFICA - Design 39

2.4.1.1 Solving the Deafness and the Muteness Problems 39

2.4.1.2 Solving the Hidden Terminal Problem and Preserving M-RTS Alignment Amongst Contenders 41

2.4.1.3 Additional Changes from the FICA Scheme . 42

2.4.2 btFICA - Complete MAC Protocol Description 43

2.5 Points of Discussion 45

2.6 Performance Evaluation 52

2.6.1 Simulation Methodology 52

2.6.2 Simulation Results - Sample Networks 53

2.6.2.1 Scenario 1 54

2.6.2.2 Scenario 2 56

2.6.2.3 Scenario 3 59

2.6.3 Simulation Results - Single Cell Random Networks . . 60

2.6.4 Simulation Results - Multi-Cell Random Networks . . . 66

2.7 Related Works 69

2.8 Conclusion 70

2.9 Appendix 71

2.9.1 Stage Synchronization Protocol 71

2.9.1.1 Synchronization at every ACK Stage 73

2.9.1.2 Network-wide Synchronization 74

3 Adaptive Channelization for Efficiency and Fairness in Future WLANs 78

3.1 Introduction 78

3.2 Adaptive Channelization in High Data Rate Wireless Networks 83

3.2.1 Models for the Adaptive Multichannel protocol (AMC) 84

3.2.1.1 AMC without ACKs 85

3.2.1.2 AMC with ACKs 86

3.2.2 Why and how is AMC fair? 87

3.3 Extended-Reservation Protocol in High Data Rate Wireless Networks 90

3.3.1 Analytical Modeling of the Extended-Reservation Protocol 94

3.3.2 Extended-Reservation Protocol Vs. AMC and 802.11-like DCF 97

3.3.2.1 **Throughput Comparision** 98

3.3.2.2 **Fairness Comparision** 103

3.4 The Pipelining Protocol in High Data Rate Wireless Networks 109

3.4.1 Description of the Pipelining Protocol 110

3.4.2 Pipelining protocol Vs. Standard 802.11-like DCF With Static Contention Window 113

3.4.3 Original Pipelining Vs. 802.11-like DCF With Optimal Contention Window 117

3.4.4 Original Pipelining Vs. Tuned Pipelining 120

3.4.5 Tuned Pipelining Vs AMC protocol 121

3.4.6 Fairness Comparision of Tuned-pipelining protocol, 802.11-like DCF and AMC . . . 125

3.5 Related Works . . . 128

3.6 Conclusion . . . 135

4 Adaptive Spectrum Distribution in Future WLANs with Multiple Collision Domains 137

4.1 Introduction . . . 137

4.2 Definitions and Terminologies . . . 140

4.3 Network Architecture . . . 142

4.4 Solving the Channel-Configurations Problem . . . 144

4.4.1 Step 1 - Determining Channel-Widths for Links . . . 145

4.4.2 Step 2 - Determining Channel-Locations for Links . . . 147

4.4.2.1 **Formulating the Channel-location Problem as a CSP (Constraint Satisfaction Problem)** . . . 148

4.4.2.2 **Approximation Algorithm for Solving the Channel-location problem** . . . 150

4.5 Performance Evaluation . . . 153

4.5.1 Part 1 - Evaluating Performance in Sample Network . . . 155

4.5.2 Part 2 - Evaluating Performance in Random Networks 159

4.6 Related Works . . . 160

4.7 Conclusion . . . 161

5 A First Look at Performance in Emerging Mobile Virtual Network Operators **163**

5.1 Introduction 163

5.2 Measurement Setup 168

5.3 Application Performance 171

5.3.1 Web Browsing 172

5.3.2 Video Streaming 178

5.3.3 Voice calls 181

5.4 Other Applications 184

5.5 Related Work 187

5.6 Conclusions 188

6 Conclusion **190**

Bibliography **194**

List of Figures

1.1 As we shift to higher data rates, the bandwidth-independent overheads dominate. Part (a) shows events for a slow network of 54 Mbps and Part (b) shows events for an 802.11n network of 600 Mbps. 4

2.1 Single cell setting with 3 clients. We have different packet sizes for each link. All nodes can hear each other. 26

2.2 FICA MAC operations for the single cell setting of Figure 2.1. 27

2.3 The dotted lines represent the nodes that can carrier sense each other. Solid lines represent client-AP associations. 36

2.4 A simple setting that shows two isolated groups of nodes (i.e., AP_1 and its client and AP_2 and its two clients), that are not synchronized in terms of protocol stage. A new incoming client, c_4, arriving in the vicinity of AP_1 and AP_2 will always sense the channel busy due to no time-domain contention. . . 49

2.5 Results for the scenario described in Section 2.6.2.1. The effectiveness of btFICA in addressing the deafness problem is shown. 55

2.6 Results for the scenario in Section 2.6.2.2. The effectiveness of btFICA in addressing the muteness problem is shown. . . . 58

2.7 Results for the scenario described in Section 2.6.2.3. The effectiveness of btFICA in addressing the hidden terminal problem is shown. 61

2.8 Results for one AP and randomly placed clients. For every flow we randomly choose a *fixed* packet size. 63

2.9 Results for one AP and randomly placed clients. For each flow we have *variable* packet sizes that are randomly chosen. 65

2.10 Single cell setting with arrival rates for flows chosen randomly. 67

2.11 Results for WLAN with 6 APs deployed in a 200m x 200m area. 68

3.1 The events happening in a network that uses the Extended-Reservation MAC protcol is demonstrated on a timeline. Here $L = 3$. The data packets within the Reservation Period belong to the same transmitter. 93

3.2 Normalized throughput versus Reservation Limit for packets of different sizes. The number of nodes is 25 and optimal contention window is assumed. 95

3.3 Throughput Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot. 99

3.4 Fairness Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot. 105

3.5 Sequence of events happening in a standard 802.11 network. . 111

3.6 Sequence of events happening in an pipelined network. 112

3.7 Saturated Normalized Throughput Vs. Varying Number of Nodes for the pipelining protocol and 802.11-like DCF with static contention window. 114

3.8 Percentages of channel time spent in no transmissions, collisions and successful transmissions with the pipelining protocol for different packet times. Number of nodes is assumed to be 16. 116

3.9 Saturated Normalized Throughput Vs. Varying Number of Nodes for the pipelining protocol and 802.11-like DCF with *optimal* contention window 119

3.10 Saturated Normalized Throughput Vs. Varying Number of Nodes for the *tuned-pipelining* protocol and 802.11-like DCF with *optimal* contention window. 122

3.11 Saturated Normalized Throughput Vs. Varying Number of Nodes for the AMC protocol with varying guardband widths, Pipelining protocols and the 802.11-like DCF 123

3.12 Jain's Fairness index for tuned-pipelining protocol and the AMC protocol with different guardband widths. 64 nodes is assumed. 127

4.1 A conflict graph where max-min fair share of each link is $B/2$. Here, we cannot get collision-free channels of width $B/2$ for all links. B is the total spectrum bandwidth. 146

4.2 Solving the CC-Problem in an Infrastructured WLAN. We get conflict-free channels for all the links. B is the total bandwidth of the wide spectrum. 152

4.3 Normalized throughput and max-min fairness measure achieved with different techniques. *Fixed Channels (y)*, means that the wide spectrum is statically divided into y channels of fixed and equal widths. 156

4.4 Total Network throughput and max-min fairness measure achieved with different techniques in random networks. 158

5.1 We conduct measurements at 11 different locations spanning across a 3000 km^2 wide area. The annotations show the names of the measurement locations along with the type of location. 170

5.2 Distribution of page load times (median, 25th and 75th percentiles): We see that (a) MVNO family `A` usually performs better than MVNO family `B`; (b) within each MVNO family one or more MVNOs is worse than the base carrier; and (c) some MVNOs (e.g., `B2`, `B3`) suffer more than others). 173

5.3 Focusing on the key observed factors shows that generally speaking the MVNOs in family `A` with higher page load times have higher RTT and the MVNOs in family `B` with higher page load times tend to have high retransmission rates. 175

5.4 Higher page load times for `B1` relative to `B` are not due to retransmissions but rather due to high radio dormancy periods. The red line in (a) shows intervals when data activity is dormant. 176

5.5 Video quality-of-experience metrics for the MVNOs and base carriers. Note that MVNO family `B` always plays the high-quality resolution and suffer significant buffering, startup delay, and video load failures. 179

5.6 TCP throughput influences video quality 181

5.7 Call quality in terms of setup time and the audio quality. While there is no significant difference in the call quality, we do observe that some of the MVNOs in MVNO family `B` have a higher call setup time. 183

5.8 TCP Uplink Throughputs. 184

Acknowledgements

I would like to give my deepest gratitude to my advisor Professor Michael Ferdman, and to my Graduate Program Director Professor I.V. Ramakrishnan.

Professor I.V. Ramakrishnan helped me significantly during difficult times in my career. He did his best to keep me progressing in my research and career. I really like his open-door policy for students and the way he listened to concerns and tried to find ways of helping students. I thank him very much for all the time, support, help and encouragement that he has given me, and for being an outstanding and valuable graduate student mentor.

I also profoundly thank Professor Michael Ferdman, who proved to be an excellent graduate advisor. I am amazed by his competence in various fields of Computer Science, his management and research skills, and helpfulness. I thank him for the time that he spent in discussing my research work with me, and giving me valuable advice and directions to make my work better and stronger. He helped me significantly with my dissertation and helped me towards a successful defense. He also provided me with support when I needed it the most and helped me at very difficult times during my Ph.D. career. He provided me with resources and membership to his hospitable COMPAS lab. He truly cares about students' progress and without doubt, he is one of the best and vital Professors for the Computer Science department. My experience at Stony Brook would have been different without him. I highly, highly appreciate all that he has done for my success.

I would also like to thank my other defense committee members, Professor Jie Gao, Professor Ellen Liu and Dr. Ritesh Maheshwari for their valuable feedback on my dissertation.

I would like to thank Professor Vyas Sekar, Professor Samir Das, Professor Phillipa Gill, Ayon Chakraborty, Gaurav Dugar, Jihoon Ryoo, Dr. Su'aad Zaman, and Naila Kabiraj for their help with the MVNO project [124].

Also many thanks to Professor Vyas Sekar. I learnt several technical and research skills from him.

I would also like to give special thanks to Professor Anita Wasilewska, Professor Steven Skiena and Professor Annie Liu for providing a positive environment and a wonderful learning experience.

I also would like to thank Brian Tria for sometimes giving me equipment for conducting my experiments, and for maintaining the compserv and sbrocks machines. These were helpful.

I would also like to thank all my friends in the Computer Science Department, in COMPAS, WINGS, Networking Research Group, OSCAR, Knowledge Systems Lab, and other Labs, that have graduated, moved or are still present, for enriching my experience here, and for their support and interesting discussions.

Very importantly, I am immensely indebted to my Parents. It is impossible to use words to describe everything that they have done for me. Without them, my Ph.D. would have been impossible. I give my sincere and deepest gratitude to them. They took good care of me and it was their presence that helped me endure the hardships that came on my path. They shared my happiness and felt my problems. My Father's encouragements, and my Mother's patience and graceful care, kept me going. I would never be able to repay them back for everything that they have done for me. Additionally, I would also like to give many thanks to Uncle Asef for his contineous support. This thesis is dedicated to them.

Chapter 1

Introduction

Over the past decade, wireless devices, such as, smartphones, tablets and laptops have gained tremendous popularity and proliferation worldwide. Cisco has forecasted that by 2019, more than 67% of the total IP traffic will be due to data related to WiFi and Cellular networks.

Additionally, over the past years, both WLAN technologies and cellular networks have undergone significant changes from different aspects, in order to improve network performance and user experience.

One of the fundamental changes that has progressively occured for WLANs is development of radios that can now transmit and receive data at very high data rates. For example, with IEEE 802.11ac standard [1] (released in 2014), now wireless radios can communicate with speeds of > 1 Gbps at the physical (PHY) layer. This is about an 18 times improvement over the past 802.11a/g standards. Furthermore, a newer standard, IEEE 802.11ax, is in the works and is expected to be released in 2019, in which the wireless PHY capacity will further be increased to 10 Gbps. These developments have occurred with the aim of improving network throughput and thus, supporting a growing number of WiFi users, while providing them good quality of experience.

However, as we explain later in this dissertation such networks are fraught with new challenges (that can severly impact network performance), for which trivial solutions are not practical. In the this dissertation, we tackle important research problems that arise in this space. We conduct principled studies for understanding performance in next-generation High Data Rate WLANs, and develop and extensively evaluate effective MAC layer solutions, for improving network performance and fairness in such networks.

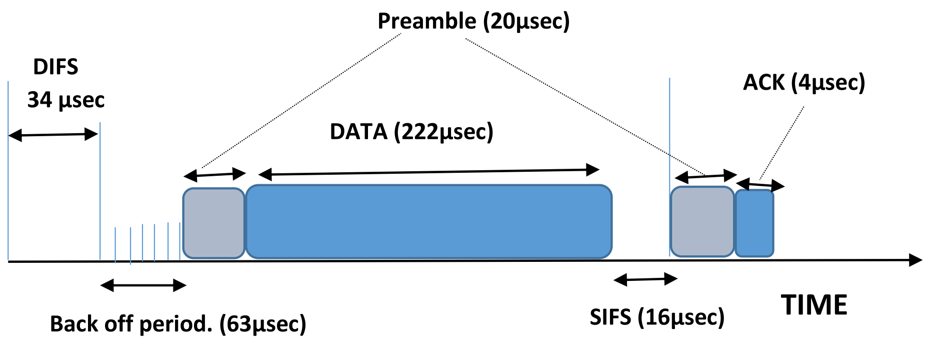
Like WLANs, cellular networks have also undergone a variety of changes over the past years. In addition to improved MAC/PHY performance with the help of improved 3GPP standards, such as 4G (HSPA+/LTE), we also have a different phenomenon emerging in the mobile market: Mobile Virtual Network Operators (MVNOs). This is an important and growing area in cellular networks, and there is significant interest in understanding performance in such networks, however, no prior work addresses this gap. In this dissertation, we develop our own measurement techniques and tools, conduct extensive and systematic measurements, as well as, perform factor analysis to better understand performance in this area.

Next, we describe in further detail the research problems that we tackle in this dissertation:

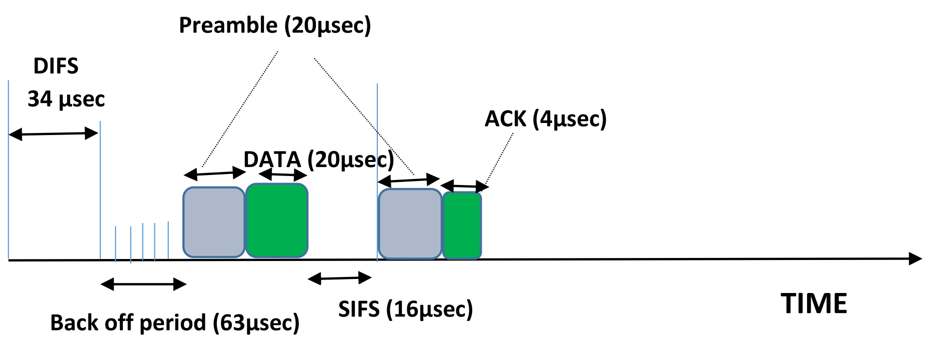
1.1 Research Problems

1.1.1 Channel Utilization Problem in High Data Rate WLANs

As WLANs are shifted from low physical data rates to higher physical data rates, the standard MAC schemes now face new challenges. The IEEE 802.11 DCF (MAC layer protocol) that uses the wide channel as a single resource casues a drastic reduction in channel utilization. While now packets will be transmitted faster than before, 802.11 DCF's throughput does not scale linearly with the physical layer datarate. The reason for this is that the 802.11 DCF protocol introduces per-packet bandwidth-independent overheads (e.g., DIFS, SIFS, backoff periods) that remain the same regardless of the channel's bandwidth. Therefore, the effect of the bandwidth-independent overhead introduced by the MAC protocol becomes more noticeable as one shifts to *higher* data rate networks, where packets will now take *smaller* transmission



(a) Illustration of overheads with 54Mbps PHY data rate. The timing values for different components are realistic.



(b) Illustration of overheads with 600Mbps PHY data rate. The timing values for different components are realistic.

Figure 1.1: As we shift to higher data rates, the bandwidth-independent overheads dominate. Part (a) shows events for a slow network of 54 Mbps and Part (b) shows events for an 802.11n network of 600 Mbps.

times. With low data rate networks, the same packet's transmission time takes longer, and hence masks the amount of time wasted by the bandwidth-indpendent overheads.

We can demonstrate this concept via a simple example shown in figure 1.1. All the timing numbers presented here are realistic. There is a 1500 Byte packet that is transmitted on both the 600 Mbps and the 54 Mbps network. Also, for the sake of understanding, we can assume that for each of Figure 1.1a and Figure 1.1b, the sequence events starting from the DIFS period to the end of ACK repeat as time goes on. Now, if we look at figure 1.1a, we can see that the total bandwidth-independent overhead, from the starting of the DIFS period to the end of ACK, sums up tp 157 microseconds. The data packet transmission takes 222 microseconds when the PHY data rate is 54 Mbps. Hence, the total channel utilization in this case comes up to $222/(222 + 157) \approx 58\%$. However, a similar calculation, for Figure 1.1b will give us a channel utilization of $20/(20+157) \approx 11\%$, which is drastically lower than what we achieved with the 54 Mbps case. In fact we can see that for 600Mbps case, 89% of the channel is wasted due to bandwidth-independent overheads.

Addressing this challenge in high data rate wireless networks is not a trivial task. As we will discuss later in the dissertation, simple approaches, such as, reducing slot time to reduce the time taken by DIFS, SIFS and backoffs is impractical. Similarly, sending very long frames or multiple packets back-to-back can also be impractical, with realistic traffic.

In this dissertation, we try to tackle this research problem, and we extensively investigate MAC solutions that appear promising, for high speed wireless networks, but were not extensively studied before in this context. Based on our insights, we develop new ideas and MAC solutions that can improve the performance in high speed wireless networks significantly.

1.1.2 Understanding Performance in MVNOs

We have spotted a new, important, interesting, but uninvestigated problem in the space of mobile networks. To the best of our knowledge, no prior work has investigated performance in *Mobile Virtual Network Operators (MVNOs).* To address this gap, we conduct a systematic measurement study, to understand performance in MVNOs, and how they compare to their base carriers. We face the challenges of finding the best metrics to study; how to design and implement our systematic measurement tests, techniques, and tools, while avoiding any form of measurement bias; where and when to conduct our experiments; finding all the appropriate possible lower-layer metrics to collect for further factor analysis; collecting the actual data; and analyzing our data in different ways, to finally reveal interesting patterns in regards to MVNO behaviors, and how they compare with their base carriers.

We hope that our techniques and insights would be beneficial to researchers, end-users, MVNOs, underlying carriers and application developers.

1.2 Contributions

We make the following 5 major contributions to solve the above mentioned research problems. (Note that each subsection below has multiple further contributions contained):

1.2.1 Exploring Channelization as a Solution for High Data Rate WLANs

In [69] we proposed the idea of adaptive channelization for improving efficiency in high data rate WLANS. However, there are also other potential MAC solutions that appear to improve efficiency in high speed settings. These two schemes are the *Extended-Reservation protocol* and the *Pipelining protocol* [117, 118]. These protocols have not been evaluated and compared against a channelization approach in high speed settings. Additionally, the level of fairness that we can attain with all these three schemes, and how they compare with each other, is also unknown, despite that fairness is a very essential metric in any shared network. We address these gaps in chapter 3, and we conduct an in-depth study of throughput and fairness in high speed settings for (1) an adaptive channelization technique (2) The Extended-Reservation Protocol and (3) The Pipelining protocol, by analytical modeling and simulations. We show that Adaptive Channelization significantly outperforms both the Extended-Reservation Protocol and the Pipelining Protocol in high data rate wireless networks.

1.2.2 Studying Channelization in Multiple Collision Domain Settings for High Data Rate WLANs

We further study the problem of adaptive channelization in multiple collision domain settings from a theoretical and algorithmic perspective. It is well-known that the IEEE 802.11 standard statically channelizes the wide spectrum into smaller channels of equal width, and classically, each access point is assigned one of these channels to operate on. While an intelligent channel assignment for interfering APs can provide better network performance than just any arbitrary channel assignment, still recent work shows that even far better performance can be achieved if the fixed 802.11 channels are not used and instead the channel width and central frequency for each AP is adapted, based upon the traffic load at each AP in the network. However, the existing work for dynamic distribution of spectrum amongst APs, provides non-overlapping channels to interfering APs, which can cause loss of spectral reuse opportunities. In chapter 4, we propose a new dynamic spectrum distribution technique in infrastructured wireless LANs, that exploits spectrum reuse opportunities and allows overlap between channels provided to interfering APs. Our technique achieves two goals: 1) max-min fairness amongst flows in the network and 2) high network throughput. Finally, we evaluate the performance of our technique, via simulations, and present its superiority when compared to single-channel 802.11-like DCF, classic fixed channelization and the state-of-the-art dynamic spectrum distribution technique for WLANs.

1.2.3 Understanding the FICA PHY/MAC protocol in High Data Rate WLANs

Drawing upon the insights that we attained from our above work, we come to the conclusion that adaptive channelization is an effective approach to take to improve performance in high data rate WLANs. While the above studies are more theoretical in nature, we also find it important to study practical MAC schemes, that build upon the concept of channelization. We in particular find it important to study the FICA PHY/MAC protocol in High Data Rate WLANs, to see how much improvement in channel utilization we can attain. FICA also channelizes the wide spectrum into multiple smaller channels, and the senders adapt the number of channels that they use based on demand. While the FICA approach appears more promising than the other proposed schemes for high data rate WLANs, it has not been studied extensively before. Hence, in chapter 2, we focus on the FICA MAC protocol, and we study this protocol thoroughly in different traffic scenarios and network topologies. We identify, for the first time, some of the serious problems that can arise with the FICA MAC protocol. We call these problems deafness, muteness and a form of hidden terminal problem. We quantify the impact of these problems on the performance of FICA via extensive simulations. Our results show that under some typical scenarios, FICA can perform even worse than the conventional 802.11 DCF, in terms of channel utilization, per-user-throughput or fairness. The insights obtained in chapter 2 motivates the need for addressing FICA's problems and paves the path for future development of better new practical MAC protocols for high data rate WLANs.

1.2.4 btFICA MAC protocol for High Data Rate WLANs

In chapter 2 we develop a new practical MAC protocol for improving network throughput and hence, channel utilization in wireless LANs that can support very high PHY layer data rates (> 1 Gbps). We call our new MAC protocol Busy Tone Assisted Fine-Grained Channel Access (btFICA). btFICA is based upon the framework of the previously discussed, prominent state-of-the-art PHY/MAC scheme for high data rate WLANs, called Fine-Grained Channel Access (FICA). While the rationale behind the FICA scheme appears effective for enhancing channel utilization in high data rate WLANs, we showed that problems, such as deafness, muteness and a form of hidden terminal problem, can easily arise with the FICA MAC protocol. These problems can degrade the network performance, if left unaddressed. This motivates us to develop our btFICA MAC protocol that uses an additional busy tone antenna. btFICA comprehensively solves all of the three problems faced by the FICA MAC protocol, while maintaining the beneficial aspects of the original FICA scheme. Finally, we show via extensive simulations, that btFICA significantly outperforms the original FICA scheme and 802.11 DCF, in different network topologies and traffic scenarios, in terms of channel utilization, per-user-throughput and fairness.

1.2.5 Conducting Measurement Study in Mobile Virtual Network Operators (MVNOs).

To the best of our knowledge, we present the first systematic measurement study to shed light on this emerging phenomenon. We design and develop measurement techniques and testbeds with Android smartphones. We take approaches to avoid measurement bias, and we study in detail the performance of 3 key applications: web access, video streaming and voice, in 2 popular MVNO families (a total of 8 carriers) in the US. In our study, each family includes the base carrier and three popular MVNOs running on top of the base carrier. While this sample study does not cover all base carriers in US or all MVNOs atop any base carrier, the carrier/MVNO choices have been done systematically, based on popularity, as we discuss in chapter 5. To simplify presentation, we refer to the two base carriers as carrier `A` and `B`. We refer to the MVNOs within the carrier `A` as `A1`, `A2` and `A3`, and within the carrier `B` as `B1`, `B2` and `B3`. The base carrier along with its MVNOs (e.g., `A`, `A1`, `A2`, `A3`) are refered to as 'MVNO family' or just 'family.'(e.g., MVNO family `A`).

Our key findings are:

- The base carrier often performs better than the MVNOs and sometimes significantly so. For instance, some MVNOs over base carrier `B` fail to load a non-trivial ($\geq$10%) fraction of YouTube video requests and can have up to 6$\times$ worse page load time.
- There is significant diversity across MVNOs within the same MVNO family, for both the `A` and `B` MVNO families. For instance, often `B2` performs considerably worse than `B1` and `B3` in MVNO family `B`.
- There are non-trivial differences between the two MVNO families; overall the MVNOs running atop `A` have better performance w.r.t the base carrier compared to their `B` counterparts.
- Finally, we see key differences across applications as well. While voice quality is largely similar across all MVNOs and base carriers, there is huge discrepancy in data performance both for web access as well as video streaming.

We hope that chapter 5 serves as a motivation for future large-scale measurement studies in this direction, that would span wider areas, larger number of MVNOs and wider variety of data plans. Additionally, we believe that our measurements, analysis and new insights are beneficial to researchers, end-users, MVNOs, underlying carriers and application developers.

1.3 Outline

The rest of this dissertation is organized as follows: In the first part, comprising of Chapters 2, 3 and 4, we thoroughly investigate the research problems that arise in high data rate WLANs and develop and evaluate solutions for improving performance and fairness in these networks. Note that, these 3 chapters are organized in a top-down fashion. This means that we begin with our most recent and practical work in this space, and then we look at our prior work containing analytical and theoretical analysis, in order to solidify the reason for why channelization is a better approach to take for improving efficiency in high data rate wireless networks.

In chapter 2, we study the FICA MAC protocol in high data rate WLANs, and propose our btFICA approach for improving performance in such networks.

In chapter 3, we investigate two other plausible MAC protocols for high data rate WLANs (i.e., the Extended-Reservation protocol and the pipelining protocol) and compare against a channelization approach. We use both simulations and analytical modeling for our study. In chapter 4, we tackle the problem of how to adaptively distribute spectrum in Infrastructured WLANs to improve performance, from an algorithmic perspective and we evaluate our schemes via both analysis and simulations.

In the second part of this dissertation (chapter 5), we develop our measurement techniques and tools to study user performance in MVNOs and perform factor analysis to find underlying reasons for MVNO performance. We present interesting insights with respect to MVNO characteristics. Finally, we conclude our dissertation in chapter 6.

Chapter 2

Understanding the FICA PHY/MAC Protocol in Next-Generation WLANs and Our btFICA PHY/MAC Solution

2.1 Introduction

The recent advancements in physical layer (PHY) technologies, (such as MIMO, high-density modulations (up to 256-QAM), and the development of tranceivers that can operate on very wide channels), have turned the once dream of having high speed wireless links, into a reality. For example, the successor of the current 802.11n standard, the 802.11ac standard, is intended to provide physical layer data rates even higher than 1Gbps at even long distances, by using 8 MIMO antennas and channels as wide as 160 MHz [10], [1], [102], [68].

However, unfortunately, the conventional 802.11 DCF[1] running at the MAC layer, causes the channel utilization [2] to drop drastically at such high PHY data rates. This is proven analytically, experimentally and via simulations both by our prior work, as well as, succeeding work [68], [69], [102], [95], [92]. For example, in [102] it is shown that when we shift from 11Mbps to 1Gbps PHY data rate, the 802.11 DCF causes the channel utilization to drop from 75% to as low as 6%.

This undesired phenomenon arises, because as we shift to higher PHY data rates, the same packets now take a proportionately *smaller* transmission time. However, the channel idle time incurred due to interframe spacings, (such as DIFS and SIFS), and due to time-domain channel contention (backoff periods), remains *unchanged*, before every packet transmission. Additionally, the channel time spent in preamble transmissions also remain unchanged, before every packet transmission. At higher data rates, the same

[1]We assume that the readers are familiar with the 802.11 DCF MAC protocol, and RTS/CTS and NAV concept in the 802.11 standard.

[2]*Channel Utilization* is defined as the ratio of the network throughput achieved to the physical layer data rate. We use the terms *Channel Utilization* and *Efficiency* interchangeably.

packet now takes a significantly smaller transmission time, thus, making the impact of the above overheads substantial at high PHY data rates. This leads to poor channel utilization.

Hence, it becomes important to develop new random access protocols for high data rate WLANs, that will provide a better usage of the underlying channel.

For this purpose, one work that is recently proposed, and that appears promising, is the *Fine-Grained Channel Access (FICA)* technique [102]. FICA attempts to improve the channel utilization by using two main ideas: (1) performing contention and backoff on the *frequency-domain*[3] instead of the time-domain and (2) dividing the wide channel into smaller subchannels of equal and fixed width, and allowing packet transmissions by different nodes on different subchannels, simultaneously. (Hence, the term *Fine-grained* channel access.)

On each of the subchannels we will have a proportionately slower data rate, than the data rate supported on the entire wide channel. Hence, the same packet will have a proportionately longer transmission time on a subchannel, than when it is transmitted on the entire wide channel. Every node can contend for and access any number of subchannels. Hence, in essence, FICA causes relatively short frequency-domain contention periods to be fol-

[3]In frequency-domain contention, nodes compete for the medium by sending signals on randomly chosen OFDMA subcarriers.

lowed by long periods of data transmissions on the subchannels. Clearly, this approach should be effective in reducing the impact of contention overheads and improving channel utilization in high data rate WLANs.

In [102], it is also argued that FICA is more promising and practical than even 802.11 with frame aggregation [4] [11], [10] enabled. The effectiveness of 802.11 frame aggregation reduces, as we shift to higher data rates. This is because it is usually not possible for each sender to individually have enough packets of suitable sizes to aggregate, in order to enhance the overall efficiency. Also, the presence of delay-sensitive packets (e.g., VoIP packets) makes the frame aggregation scheme unsuitable [102], [92], [68]. In contrast, FICA allows in a sense, an "aggregation" of packets across different senders, after the shared frequency-domain contention period, and hence, is much more practical in enhancing channel utilization.

While the FICA approach appears appealing for improving efficiency in high data rate WLANs, the scenarios used in previous literatures [102], for evaluating the performance of FICA, are very limited. Hence, our **first goal** is to study the FICA MAC protocol thoroughly, with realistic traffic scenarios and network topologies. Since in real-world settings, senders typically have packets of different sizes at the MAC layer [92] [16], we find it important to also analyze FICA's performance under such settings.

[4] This option of 802.11 allows the same sender to send multiple MAC packets back-to-back, after winning the channel, in order to improve efficiency.

To the best of our knowledge, this is the first work that identifies the problems that can arise with the FICA MAC protocol, when packets of *different* sizes are present in the network. We refer to these problems as *deafness*[5], *muteness* and a certain form of *hidden terminal problem*. Our careful analysis and simulation results show that these problems can be sufficiently serious to cancel out the advantages of fine-grained channel access. In fact, we show that, these problems can cause FICA to perform worse than even the plain 802.11 DCF without packet aggregation, in terms of channel utilization, per-user throughput or fairness. We also show that in some occasions, where FICA provides high efficiency, it does so, at the cost of starving several nodes in the network.

Our above findings motivate the need to address the problems that arise with the FICA MAC protocol, in order to truly benefit from frequency-domain contention and fine-grained channel access.

Motivated by the above insights that we achieved, our **second goal** is to develop a new MAC protocol for improving efficiency in high data rate WLANs. We call our MAC protocol, *Busy Tone Assisted Fine-Grained Channel Access (btFICA)*. btFICA is based upon the FICA framework and uses an additional busy tone antenna. btFICA *comprehensively* solves all three of the problems faced by the FICA MAC protocol, while *preserving*

[5]We have identified a new form of deafness which is different from the type of deafness that is previously discovered in the contexts of directional antenna networks [40] and multichannel networks described in [70].

the positive features of the FICA scheme.

Note that the extra busy tone antenna used by btFICA, should not be viewed as an extra costly radio meant for data packet reception (transmission). A busy tone antenna is a lot simpler. It is used to just detect (emit) energy on different busy tone channels and there are no modulation/demodulation overheads involved [40], [70], [91], [92].

The rest of this chapter is organized as follows. In Section 2.2 and Section 2.3, we briefly describe the FICA PHY/MAC scheme, and we revisit the problems faced by the FICA MAC protocol, respectively. In Section 2.4 we develop our btFICA MAC protocol. In Section 2.5 we discuss why we did not consider some other potential solutions for addressing FICA's problems. In Section 2.6 we present our simulation results for realistic network topologies and traffic scenarios. Related work is discussed in Section 2.7, and we conclude this chapter in Section 2.8.

2.2 Description of the FICA Scheme

FICA defines both a new PHY and MAC scheme for high data rate WLANs. The FICA MAC protocol is a carrier sensing (CSMA/CA) based, random access scheme. The FICA MAC protocol is designed for WLANs where each AP and each client is equipped with a single *half-duplex* radio [102].

The radios are capable of operating on very wide channels, and can support data transmission/reception at very high rates. If a sender has packets to send, the sender has to pass through two phases: (1) *contention phase* and (2) *DATA/ACK phase*. We further briefly describe the FICA PHY/MAC scheme below:

2.2.1 The FICA Physical Layer Architecture

At the PHY layer, FICA makes use of the OFDMA [119] technology. In OFDMA, the wide channel is divided into many narrow-band orthogonal smaller channels, called *subcarriers*. The FICA PHY layer is designed to provide the following capabilities: (1) Allow the frequency-domain contention to take place, while preserving orthogonality amongst subcarriers.(2) Allow the subcarriers to be grouped into subchannels, and allow different nodes to transmit frames on different subchannels, simultaneously, while preserving orthogonality amongst subchannels. The receiver can listen on the entire wide channel and can receive frames arriving on different subchannels, simultaneously.

Now, during the *DATA/ACK phase*, the nodes involved divide the entire wide channel into M subcarriers. After accounting for guardband on the two ends of the wide channel, the remaining set of subcarriers is partitioned into smaller groups of K subcarriers each, called *subchannels*. The nodes will treat each of these subchannels as a unit, i.e., senders will be transmitting each frame on a subchannel.

Now, during the *contention phase* at a node, the FICA PHY divides the entire wide channel into 2M subcarriers. After accounting for the same amount of guardband, half of the remaining subcarriers here are used to represent all the subcarriers of all of the subchannels of the the DATA/ACK phase. The other half is used to send control information, such as, NAV, etc.

2.2.2 The FICA MAC Protocol

The FICA MAC protocol allows different senders to share the medium. We describe the main parts of the MAC protocol below:

2.2.2.1 Frequency-domain Contention

Briefly, a sender, first, begins to carrier sense on the entire wide channel for a certain specified period of time [6]. If it finds the entire wide channel idle during this entire time, it contends for randomly chosen subchannels, by immediately sending a special symbol called M-RTS. After an SIFS period, the sender expects to hear back another symbol called M-CTS, that contains the winner information for each subchannel. This M-RTS/M-CTS handshake

[6] This period is DIFS for clients and Long-DIFS or Short-DIFS for APs [102].

uses the entire channel.

Note that an *M-RTS symbol* consists of a set of tones sent on subcarriers. In order to contend for a subchannel, each sender randomly selects one of the K subcarriers that represent this subchannel, and sends a tone on it. Now, if we have multiple contenders, then all of them will simultaneously send out their M-RTS symbols.

Hence, the nodes in the vicinity, that hear an M-RTS symbol, can in reality be hearing a combination of individually transmitted M-RTS symbols. From these surrounding nodes, only the nodes that are indicated as *potential receivers*[7] will be involved in resolving contention. Each potential receiver selects the highest active subcarrier that it sees on each subchannel as the *winning subcarrier* on that subchannel. All potential receivers will simultaneously broadcast the information about the winning subcarriers in their M-CTS symbols. Upon hearing the arriving M-CTS symbol, if the sender finds its randomly chosen subcarrier selected as the winner for that subchannel, then the sender will consider itself a winner on that subchannel.

2.2.2.2 DATA/ACK Phase

Until this point we have the contention phase taking place. Now, if the sender wins on any of the subchannels for which it contended, then after

[7] *Potential receivers* are those nodes for whom the senders are contending.

an SIFS period, the sender will begin its DATA/ACK phase, and will send its data packets on all the wun subchannels. After it finishes sending all its data packets, it waits to receive ACKs on the corresponding subchannels. If a sender does not receive an ACK back on any of its subchannels after an SIFS duration, then it will assume that a collision has occured on that subchannel. A receiver will generate an ACK packet for each subchannel on which it received an intended data frame successfully. The ACKs are sent by the receiver, after an SIFS period, after receiving all intended data frames arriving on different subchannels.

2.2.2.3 Frequency-Domain Backoff

Every node, i, also maintains a local *Contention Window* variable, CW_i. CW_i represents the maximum number of subchannels for which the node i is allowed to contend. Initially, CW_i is set to the total number of subchannels in the network. In [102], the authors provide two different algorithms, for updating CW_i, namely, $RMAX$ and $AIMD$[8]. $AIMD$ stands for the Additive Increase and Multiplicative Decrease strategy. Here, if a sender i receives ACKs for all the packets that it had transmitted, then, it increases CW_i by 1 (Additive Increase). However, if the sender concludes a collision on $p\%$ of the subchannels on which it had transmitted data packets, then the sender

[8]In this paper we focus on FICA's $AIMD$ scheme , which is shown to be the better of the two schemes for FICA [102].

will reduce CW_i by $p\%$ (Multiplicative Decrease). The AIMD algorithm is said to perform *frequency-domain backoff*.

2.2.2.4 Network Allocation Vector (NAV) Band

In the M-RTS and M-CTS symbols, there exists a reserved set of subcarriers, called the NAV Band. Similar to the purpose of the NAV field in the standard 802.11 DCF, this NAV band can be used in order allow Virtual Carrier Sensing to take place, and thus, reduce the hidden terminal problems that can cause collisions at receivers. Each subcarrier in the NAV band stands for a specific data packet transmission time. Senders when sending an M-RTS, will also specify the longest data transmission time that they might require by encoding this information in the NAV band. Each potential receiver will then echo back the longest transmission time it hears, in the M-CTS symbol which it sends. The nodes that overhear an M-CTS, will use the information in the NAV band to defer contention for the longest time needed.

For more details in regards to the FICA PHY and MAC schemes, we refer interested readers to [102].

2.3 Performance Issues with the FICA MAC scheme

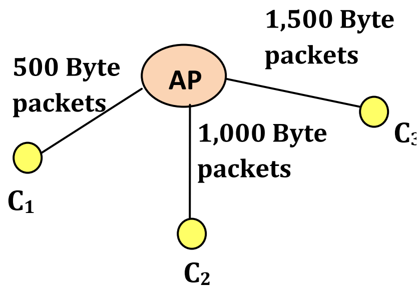
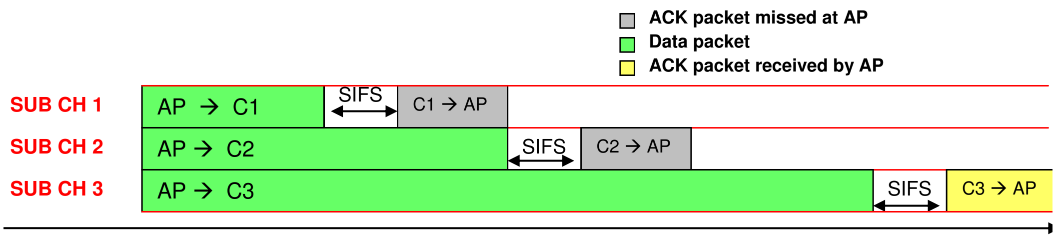


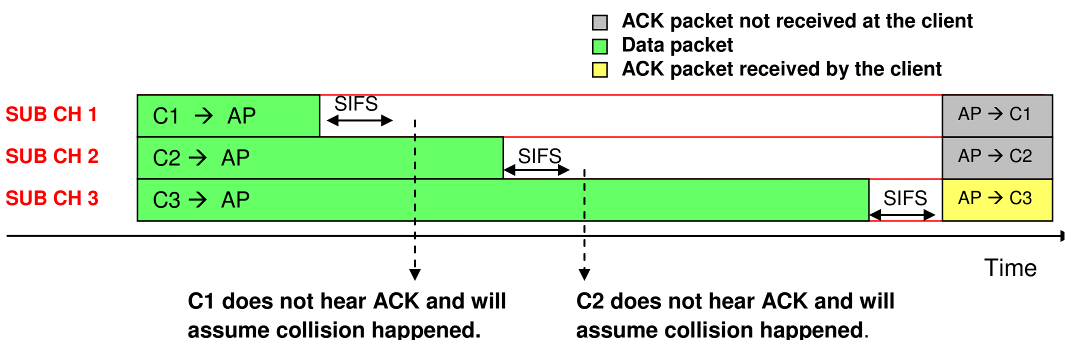
Figure 2.1: Single cell setting with 3 clients. We have different packet sizes for each link. All nodes can hear each other.

Our studies show that the FICA MAC protocol performs well, in terms of channel utilization and fairness, when all nodes in the network have packets of the same size to transmit. However, in real-world settings, senders usually have packets of different sizes at the MAC layer, depending upon upper layer protocols and applications in operation. For example, in [92], the authors collected packet traces in a real-world WLAN deployment at Duke University. Their study shows that the frame sizes generated by Skype, Web browsing and HD streaming sessions were on average, 511, 1063 and 1424 bytes, respectively. Also, we can see from the packet traces collected in the Sigcomm 2008 conference [16], that senders in WLANS can usually have frames of different sizes at the MAC layer. Thus, having a MAC protocol that provides high channel utilization and a good level of fairness amongst flows, even when frames of different sizes are present in the network, becomes important.

However, we find that three new problems can arise with the FICA MAC



(a) Illustration of the deafness problem. Activities on only 3 of the subchannels are shown.



(b) Illustration of the muteness problem. Activities on only 3 of the subchannels are shown.

Figure 2.2: FICA MAC operations for the single cell setting of Figure 2.1.

scheme, when packets of different sizes are present in the network. We call these problems (1) *Deafness* (2) *Muteness* and (3) a certain form of *hidden terminal problem*. As we show in Section 4.5, these problems can degrade FICA's performance drastically. We explain each of these issues in detail below:

2.3.1 The Deafness Problem

We say that *deafness* occurs when a sender finishes successful transmissions to some of its receivers on some of the subchannels, but *cannot hear the ACKs* intended back for it, because it is still busy transmitting to receivers on other subchannels. Here, we say that the *sender* is *deaf* to the ACKs coming from its receivers.

The sender upon not hearing the ACKs related to its successful packets, will incorrectly conclude that these packets were not successful due to collisions, which can lead to needless retransmissions and unnecessary reduction of the sender's CW. The deafness problem can *reduce channel utilization* significantly, as shown by the following example:

Let us consider the scenario of a single cell with one AP and 3 clients as shown in figure 2.1. For simplicity, let us assume only downlink traffic. For clients c_1, c_2 and c_3, the AP always has packets of sizes 500, 1000 and 1500 bytes, respectively, to send. It is clear that after any M-RTS/M-CTS handshake, the AP always wins on all the subchannels that it had chosen randomly. To give a fair channel access opportunity for all flows, the AP serves the data packets for each client in a round robin fashion. Also, it is clear that we will not have any collisions occuring on any of the subchannels.

Let us assume that the AP has started transmitting data to all its clients, as shown in figure 2.2a. Now, c_1 will be the first to finish receiving all its intended data packets, arriving on different subchannels, because c_1's packets are smaller than the packets of all other clients. After SIFS, c_1 transmits ACKs on its respective subchannels, however, the AP is deaf to these ACKs, because it is busy transmitting to c_2 and c_3 (figure 2.2a). Similarly, when c_2 finishes receiving its data packets and transmits its ACKs, the AP is again deaf to these ACKs also, because, it is busy transmitting to c_3 (figure 2.2a).

After the AP finishes all its transmissions, the AP switches to receive mode and waits for ACKs on all the subchannels it used. However, it will only hear the ACKs coming from c_3.Hence, despite that c_1 and c_2 received their packets successfully, the AP will incorrectly conclude that collisions occurred at these clients.

This has adverse effects, because, (1) the AP will incorrectly assume that there is high congestion in the network and thus, it will needlessly reduce its CW in the same way as described in Section 2.2.2.3. Thus, the AP will contend for a lesser number of subchannels in the next round, leaving several subchannels, (on which otherwise successful transmissions could have taken place), empty. (2) The AP will cause further channel wastage by retransmitting already successful packets for c_1 and c_2, on the already limited number of subchannels in the next round.

Now, deafness again repeats in future rounds, which will eventually shrink the CW of the AP to very small values. This will cause the AP to access only a very small number of subchannels (1 or 2 out of a 128, in our analysis), during data transmission rounds, despite that it has high demands. Additionally, we also find that the deafness problem also causes many of the already successful packets for c_1 and c_2, to be needlessly retransmitted multiple times. This example clearly shows how deafness can lead to a very inefficient usage of the channel.

One can argue that deafness would not have happened if the AP had picked and transmitted packets of the same size in each transmission round. However, this may not be an effective approach, because the AP might not have enough packets of the same size to send on all the available subchannels. Thus, again we can still have unused subchannels which will cause the efficiency to drop. Also, some packets can be delay-sensitive and hence, should be transmitted immediately.

Note that, deafness can also occur at the client, when the client has packets of different sizes to transmit to the AP. However, this case is rare. This can only happen when all the largest packets sent by the client face a collision at the AP, but AP finishes receiving all its other intended data packets, while the client is still busy transmitting its largest frames.

2.3.2 The Muteness Problem

Muteness occurs when a *client* finishes successful packet transmission(s) to the AP, and waits to receive ACK(s) from the AP after an SIFS period, however, the AP *cannot transmit back ACKs* during this time, because it is busy *receiving* from its other clients on other subchannels. We say that, here, the AP is *mute* for this client. When the client does not receive the ACK(s), after an SIFS period, the client incorrectly assumes that collision(s) have occured on its respective subchannels. The muteness problem can cause *starvation of nodes* and *unfairness*, as shown by the following example:

We now again consider the same scenario as shown in figure 2.1, however, this time with only uplink traffic. c_1, c_2 and c_3 can hear each other and they always have packets of sizes 500, 1000 and 1,500 bytes, respectively, to transmit to the AP. Initially, the CW of all the clients is set to the total number of subchannels. To avoid distracting details during explanation, let us also assume that during the M-RTS/M-CTS handshake no two clients win on the same subchannel and hence, we have no collisions on any subchannels. All the clients start their data transmissions at the same time on different subchannels.

Since there is a single half duplex radio on the AP, it has to finish receiving on all the subchannels, before it can send back ACKs on these subchannels. c_1's packets are the smallest in size (500 bytes), hence, c_1 finishes its data transmissions first, and waits to hear its ACKs. However, the AP is mute for c_1 because it is still busy receiving data from c_2 and c_3 (figure 2.2b), which are sending packets of larger sizes than that of c_1. Since, c_1 does not receive an ACK after an SIFS period, on any of its subchannels, c_1 incorrectly concludes that collisions occured on all its subchannels, and thus incorrectly suspecting high congestion, c_1 aggresively reduces its CW to 1. Similarly, c_2 will also face a mute AP, after it finishes sending its data packets, and thus, will needlessly reduce its CW to 1. Now, when the AP finishes receiving from c_3, the AP switches to transmit mode and transmits ACKs for all the packets that were successfully decoded, including c_1's and c_2's packets(figure 2.2b). However, these ACKs cannot be decoded by c_1 and c_2.[9] Only c_3 can correctly decode its ACKs on all its subchannels, and hence, unlike c_1 and c_2, c_3 will keep its CW at the maximum size.

Thus, clearly, in the next round, each of c_1 and c_2 will contend for only 1 subchannel, but c_3 will contend for all the subchannels and will win on almost all of them. Moreover, if c_1 and c_2 win the contention and start their data transmissions, then unlike c_3, they will waste the channel resources by retransmitting already successfully received packets. Furthermore, for c_1 and c_2, the muteness problem occurs repeatedly, which will not only cause the CW for the two clients to remain only one, but will also cause multiple retransmissions of the same successful packets in future rounds. On the

[9]This is because c_1 and c_2 by now have switched to an FFT size that is twice the IFFT size of the AP, in order to hear new M-RTS/M-CTSs.

otherhand, c_3 will again receive ACKs back on all its subchannels and hence, will maintain a large CW.

Hence, it is clear that the clients c_1 and c_2 will starve, until c_3 finishes its transmissions. Note that the channel utilization is enhanced in this example, however, at the cost of starving the clients with smaller packet sizes. This example clearly shows how muteness can cause unfairness in the network.

Note that, one way of solving the deafness and muteness problems, and allowing senders to be correctly informed of their successful transmissions, might be to replace the half-duplex FICA radio by a full-duplex radio. Now, the nodes can decode (transmit) ACKs arriving on some subchannels, while transmitting (receiving) data packets on other subchannels. However, full-duplex radios are not only costly, but also they cannot solve the hidden terminal problem described in Section 2.3.3, below.

2.3.3 The Hidden Terminal Problem

The hidden terminal problem that we are focusing on, in this Section, is different than the hidden terminal problem that causes a collision at a receiver when it is receiving its intended data packets. The NAV concept in FICA is sufficient to reduce this form of hidden terminal problems. In this Section we are dealing with the *hidden terminal problems that can cause collision(s) at the sender when it is busy receiving its intended ACK(s).*

Here, when a sender S is busy receiving ACK packets from its receiver R, another node X, hidden from R but in the vicinity of S, can sense the channel to be idle, and transmit its M-RTS. This will cause collisions with the ACKs arriving at S.

Note that in FICA, nodes do not undergo a random waiting time before sending an M-RTS. Thus, this form of hidden terminal problem can occur consistently, and thus if not addressed, can lead to *poor network thoughput* and *starvation*.

We further explain this via the following example. Let us consider the scenario shown in figure 2.3. Such topologies are also observed in real-world deployments [94]. For the sake of simplicity, let us assume only downlink traffic. AP_1 and AP_2 always have packets of sizes 500 bytes and 1500 bytes, respectively, to transmit to their clients. Initially, each AP has a CW of the maximum size. In the beginning, AP_1 and AP_2 listen on the entire wide channel for a period of Long-DIFS, and they both transmit their M-RTSs simultaneously. Clearly, both of them will win on all the subchannels after the M-RTS/M-CTS handshake, and they will both begin their transmissions.

All of these transmissions arrive at the respective clients successfully. AP_1 finishes its transmissions first because it has packets of smaller sizes than AP_2. However, the ACKs arriving back to AP_1 will collide with AP_2's on going transmissions. Hence, AP_1 will incorrectly conclude collisions on all subchannels, and will reduce its CW to 1, and will needlessly retransmit all of these successful packets in the future rounds. Now, AP_1 will find the channel busy for the entire time that AP_2 is transmitting. AP_1 starts seeing a clear channel after AP_2 finishes its transmissions. When c_2 begins sending its ACKs after an SIFS period, these ACKs cannot be heard at AP_1 because AP_1 and c_2 are hidden terminals. Hence, AP_1 continues to sense the spectrum idle for a Long-DIFS period and *starts its M-RTS transmission, which causes collisions with the ACK(s) arriving at* AP_2.This will cause AP_2 to also incorrectly conclude collisions on all its subchannels. Thus, AP_2 will also needlessly reduce its CW to 1 and will wastefully retransmit already successful packets in future rounds.

We can see that the above hidden terminal problem can easily occur, by observing the PHY parameters used in FICA [102]. In FICA, we have $SIFS = 16\mu$sec; $slottime = 9\mu$sec; $DIFS = SIFS + 2 * slottime = 34\mu$sec; Long-DIFS $= DIFS + slottime = 43\mu$sec; and the smallest preamble time is $preambletime = 46.8\mu$sec. Hence, when AP_2 finishes its data transmissions, AP_1 will start its M-RTS transmission, after 43μ seconds. This clearly causes collisions at AP_2 which is still receiving the preamble of the ACK packets from c_2.

Note that, the main purpose of having SIFS periods between entities involved in a dialogue[10], and a form of DIFS period before a new dialogue can

[10]By a *dialogue* we mean the *M-RTS/M-CTS/DATA Packets/ACKs* exchange.

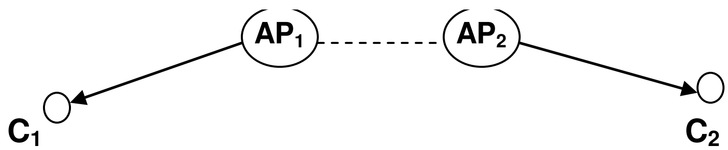


Figure 2.3: The dotted lines represent the nodes that can carrier sense each other. Solid lines represent client-AP associations.

begin is to give all the entities, including ACKs, involved with the on-going dialogue priority in transmission. While this proves effective in allowing an ongoing dialogue to complete before a new one could begin in single collision domain networks, where all nodes can hear each other, this is not the case in networks where we have hidden terminals.

Now, because AP_1 started its idle Long-DIFS interval earlier than AP_2, it so happens that for the next round of transmissions, AP_1 manages to finish its contention phase earlier than AP_2, and begins its data transmission. AP_2 upon seeing the channel busy defers from contending. Moreover, AP_1 not only accesses only 1 of the many subchannels present, but also wastes it, by retransmitting an already successful packet. Note that, the same form of hidden terminal problem occurs again, because now AP_2's M-RTS will collide with the ACKs arriving at AP_1.

This problem occurs consistently, which will lead to both APs maintaining a CW of 1, and both APs performing unneeded retransmissions.Furthermore, AP_1 and AP_2 cannot share the entire wide channel at the same time, i.e., AP_1 and AP_2 take rounds in accessing a small portion of the wide channel, which causes an inefficient usage of the channel. Clearly, this causes the overall channel utilization to drop, as well as, starvation of both the APs. Hence, it becomes important to address this problem of FICA.

Recall that, in 802.11 DCF, this form of hidden terminal problem that can cause collisions at the sender is addressed by using RTS packets and the NAV concept. Here, the nodes receiving the RTS also update their NAV accordingly. Note that, in 802.11 DCF, even if RTS/CTS is not used, then this type of hidden terminal problem may not have severe effects. This is because, unlike FICA, in 802.11 DCF, defering nodes do not immediately transmit after hearing the channel idle for a DIFS period. They go into a random time-domain backoff, which reduces the chances of such collisions at senders.

Note that FICA's NAV handling is not sufficient to prevent such hidden terminal problems. In [102] nothing is mentioned about whether, with FICA, the nodes in the vicinity of the sender that *receive the M-RTS*, use the NAV band information here to defer their contention, or not. However, even if such NAV information is not ignored in FICA, then this will still not prevent the consistent hidden terminal problem of this example and of similar scenarios. This is because, here, the *AP*s can't even receive each other's M-RTS, because at each AP, the arriving M-RTS constantly gets collided with the ACK reception here. Moreover, when two neighboring senders simultaneously transmit their M-RTS, they will miss each other's M-RTSs thus, will not be able to update their NAV accordingly in order to prevent such hidden terminal problems.

Also, it can be argued that, this form of hidden terminal issue with FICA would not have occured if, instead of giving each AP the entire wide channel to operate on, we had divided the spectrum amongst the APs, and then used the FICA scheme for each of the cells. However, we argue that one of the implicit benefits of the FICA scheme, when all nodes operate on the same entire wide channel, is that FICA can adaptively and efficiently distribute the available wide channel amongst flows, based upon the traffic demands of the flows, in a completely distributed fashion. (Note that an intelligent spectrum assignment to links can enhance network throughput [86] [77].) When a node's traffic demand changes or when interference levels change in the network, FICA can quickly adapt the spectrum distribution amongst links. This is unlike previous channel assignment schemes [86] [77], that can incur relatively large adaptation overheads when interference levels and traffic demands change in the network.

Hence, we find it important to address such hidden terminal problems that arise with FICA, while allowing all nodes in the network to be assigned the entire wide channel.

2.4 btFICA: Busy Tone Assisted Fine-Grained Channel Access Scheme

In this Section we develop our btFICA MAC scheme for high data rate WLANs, that solves the problems discussed in Section 2.3, while keeping the strengths achieved with FICA.

2.4.1 btFICA - Design

2.4.1.1 Solving the Deafness and the Muteness Problems

From sections 2.3.1 and 2.3.2, we make the key observation that *FICA's Acknowledgment scheme cannot fulfill its purpose of informing the senders of successful packet receptions.* If somehow, the senders had correct knowledge about the successful reception of their packets, then the senders would have taken correct corresponding actions, thus saving the channel from an inefficient usage. Hence, we argue that it is necessary to develop a new ACKing scheme for FICA, that prevents deafness and muteness, and that allows the senders to accurately know the state of their transmissions.

For this purpose, we equip every node with one additional, half-duplex, busy tone interface, that is capable of receiving (emitting) energy on multiple busy tone channels, simultaneously.[11] Busy tone (BT) interfaces with such capabilities are implementable [116], [92], [102].

Now, for every subchannel, we have a separate BT channel. *Upon reception of a correct packet, instead of having the receiver send an ACK packet back on the same subchannel, we make the receiver send a tone on the corresponding BT channel, for acknowledgement.* The receiver can continue receiving on the subchannels while sending tone(s) on BT channel(s), hence solving the muteness problem. The sender can also receive tones on BT channels for sent data packets, while it is transmitting on its other subchannels, hence, solving the deafness problem.

We use busy tones because, it is not only simple to implement, but, it also allows the receiver to instantaneously inform the sender of whether its transmission was successful or not. For btFICA, we allocate a portion of the wide channel for the BT channels and the guardbands needed between them. Our results in Section 4.5, show that the impact of this overhead on the performance of btFICA is not significant. We call the rest of the entire wide channel as the *data channel*. We use FICA PHY when operating on the data channel. Like FICA, we use the data channel for sending M-RTS/M-CTS symbols and DATA packets. However, unlike FICA, btFICA does not have explicit ACK packets.

---

[11] Busy tone channels are very narrow band channels (in the range of 0.1 to 10 KHz [52]).

For the correct operation of btFICA, the power level for the BT interface should also be adjusted, so that the channel gain for both the data channel and the BT channels are the same.

Also, note that, unlike previous works that deal with busy tones [52], [40], [70], we are using busy tones in a new context to solve new problems that arise with FICA. Also, in contrast to the previous works, we are making use of a BT interface that is capable of operating on multiple BT channels at the same time.

2.4.1.2 Solving the Hidden Terminal Problem and Preserving M-RTS Alignment Amongst Contenders

The solution that we proposed in Section 2.4.1.1 also solves the hidden terminal problem described in Section 2.3.3. This is because, a receiver, R, no longer sends ACK packets on subchannels, that could get collided at the sender, S, due to an M-RTS transmission from a hidden node X.

However, we do observe that there is one issue that is still occurring. S will not begin sensing the channel, in order to participate in the next contention round, before it finishes receiving its busy tones. However, X not hearing the busy tones, can begin sensing and seeing the start of an idle period before S. This can in turn lead X to transmit its M-RTS, earlier than S, and thus, an M-RTS/M-CTS handshake might complete without S even participating. Thus, S will not be able to transmit in the next round of data transmissions within the neighborhood. This can reduce channel utilization because the nodes that got a chance to transmit, might not have enough packets to utilize all the subchannels efficiently.

Hence, to address this we take the following approach. *After the sender finishes transmitting its largest data packets, the sender sends padding bits on those subchannels, for the entire time that it is receiving tone(s) from its receiver(s).* This can increase the chances for all nodes in the vicnity of both the sender and the receiver, along with the sender and the receiver themselves, to begin the next contention round at the same time. This can result in more nodes participating in contention, which can lead to more simultaneous transmissions within a neighborhood, and thus, better channel utilization.

2.4.1.3 Additional Changes from the FICA Scheme

We make use of this opportunity of having a separate BT interface, to also solve the hidden terminal problems that can corrupt data packet receptions at the receiver. Hence, in btFICA, for every intended data packet that the receiver r, started to receive correctly, r emits a tone on the corresponding BT channel for the entire time that it is receiving the packet. This will allow nodes in the vicinity of r that hear the tone(s) to defer contention. Hence, we do not need the NAV Band in the M-RTS and M-CTS symbols anymore.

This approach is better, because it is more accurate in informing contenders of whether there is an actual receiver in their vicinity that is actively involved in packet reception. In FICA, nodes receiving an M-CTS defer contention, but for the longest time that might be needed by a neighboring potential receiver, (i.e., a node that might be a receiver), to finish receiving its packets. Thus, FICA's approach for solving such hidden terminal problems is conservative, which can suppress harmless transmissions in the neighborhood.

Now, for btFICA, we also find it essential to design protocol operations that will protect the M-CTS arriving back at a sender s, from a collision with an M-RTS of a node that cannot hear the M-CTS. Hence, we allocate one more extra BT channel, which we call Q. Right after s finishes sending an M-RTS, s will start emitting a tone on Q and will continue doing so, until s finishes receiving the M-CTS. The nodes in the vicinity of s that hear this tone, will defer beginning a contention.

2.4.2 btFICA - Complete MAC Protocol Description

The btFICA MAC protocol, is for the most part similar to the FICA MAC protocol. Whenever the node is idle, it listens on both the wide data channel *and* the BT channels. In order to transmit data packets, a sender s, first starts carrier sensing on *both* the entire wide data channel and all the BT channels, for the same period of time as specified in FICA. If s finds the entire medium to be idle, it sends an M-RTS, and then begins emitting a tone on Q, and waits to receive an M-CTS.

When a node, n, receives an M-CTS, the node defers contention for a period of W_t, where $W_t = SIFS + preambletime + PLCPheaderTime + MACheaderTime$. This is inorder to give enough time for a receiver in the vicinity, to emit a busy tone. If n does not hear any tone after waiting for the W_t period then n resumes participating in contention normally.

A potential receiver, after receiving an M-RTS, will send back an M-CTS, *only* if it is not hearing any tones, and it is not waiting to hear any tones, on any of the BT channels except for Q. After sending the M-CTS, the potential receiver will switch to listening on the data subchannels, in order to receive data packets.

If s does not receive an M-CTS, s will stop the tone on Q and will will repeat the contention process again. If s correctly receives back an M-CTS, then s will stop the tone on Q and will start sending its data packets on the subchannels that it won, while listening on the corresponding BT channels.

If the receiver, r, starts to correctly receive a frame meant for it on a subchannel c, it will emit a tone on the corresponding busy tone channel b_c, for the entire time that it is receiving the frame. r knows if a frame is meant for it, if it can successfully decode the header of the frame. After r receives the entire frame, it checks for errors. If the frame is successfully decoded, then r will continue to transmit the tone on b_c for $A_t = SIFS + slottime$ more, for acknowledgment. Else, r will stop emitting the tone.

If s hears the corresponding busy tone for its frame contineously, since the beginning of its frame payload transmission until the end of its frame transmission, plus an additional A_t period, then the sender will conclude a successful packet reception. Otherwise, s will conclude a collision for this frame.

When s finishes sending its largest data packets, s will start to send *padding bits*, on these subchannels, that will take a maximum time of A_t. The padding bits are just ignored by r. s stops transmission on a subchannel as soon as it stops receiving the corresponding busy tone for that subchannel.

2.5 Points of Discussion

How can we cope with fadings on the narrow-width busy tone channels? There exists several PHY techniques for reducing the effect of fading on narrow-band channels. For example, one way is to use two or more MIMO antennas with antenna diversity schemes [18], [49], [54], instead of one antenna, when listening on the BT channels. Another technique is to allocate *two* narrow-band BT channels, b_{c1} and b_{c2}, that are spread out in the frequency spectrum, for each subchannel, c. b_{c1} and b_{c2} will face different fading characteristics, and are used as one unit. Hence, at the receiver, if a tone is detected on either b_{c1} or b_{c2}, the receiver will assume that it is receiving a tone for the subchannel c. With channels as wide as 160 MHz, and with each BT channel having a relatively very small width, the overhead of having two BT channels per subchannel should not be significant.

To avoid the muteness problem, why should we not just make the clients (senders) wait for a longer period than SIFS, in order to receive their ACKs? Before starting its data transmissions, a sender, s, usually knows of the longest possible time for which the nodes in its vicinity might be *receiving* data packets, using the NAV Band of its received M-CTS. While the sender s can wait for this entire period of time, to hear back an ACK [6], this may not be a good approach to take. This is because, after s finishes its data transmissions and waits for ACKs, there can be nodes in the vicinity of s that will begin sensing the channel idle and begin data transmissions. These data transmissions can collide with the ACKs that might later arrive at s. Another issue that can arise is that, during the time that s is waiting for its ACKs, potential receivers in the vicinity of s might send an M-CTS, which will not be received at s, since s is in its DATA/ACK phase and hence, is listening with a smaller FFT size [12] [102]. Thus, after s finishes waiting for its ACKs, s will not have a proper NAV set,

[12]With FICA PHY, a node during its contention phase will switch to FFT/IFFT size that is twice that of the DATA/ACK phase.

and thus, can begin a new transmission causing collisions at those receivers. In contrast, our busy tone approach, accurately and instantenously informs s of whether its transmission(s) were successful, without causing any of the above problems.

We can solve the deafness problem if the sender adds padding bits on its smaller sized packets, in order to make its transmissions on all the subchannels take the same time. While this fix is effective in solving the deafness problem, and is more effective than packet fragmentation discussed in [6], it still cannot solve the other two problems in Section 2.3. Our busy tone approach is a simple technique that solves all the problems comprehensively at the same time. It also provides additional benefits, such as reducing the acknowledgment overhead. With btFICA only 1 slot time is spent in signalling an "ack", which is in contrast to FICA, where the ACK packet transmission spans several time slots.

Why did we not consider the approach where nodes always sense the channel idle for at least *SIFS+entire ACK packet transmission time+defined DIFS*, before sending an M-RTS, so that the hidden terminal problem would not occur? While this approach can solve the hidden terminal problem at the sender, clearly this is inefficient. Also, the M-RTS misalignment discussed, in Section 2.4.1.2 will still occur. btFICA solves the hidden terminal problem at the sender without incurring large overheads.

How will btFICA work in networks where clients experience different SNR from the AP and thus can support different rates? In this case, even if we have one packet size in the entire network, the packets related to clients that support different rates will have different transmission times. Hence, while FICA will again face the same issues as in Section 2.3, btFICA copes well in such settings.

Both FICA and btFICA do not go through time-domian contention. Can this aspect itself cause consistent collisions or starvations, in multiple collision domain settings? In multiple collision domain settings, we can encounter the situation where two senders hidden from each other transmist their M-RTSs, however, their M-RTS misalignment is large enough to cause consistent collisions at receivers. In order to alleviate this problem, the authors in [102] propose that if a sender does not receive back M-CTSs consistently, then, the sender can go through a random time-domain back off. This is an ad-hoc solution that might not always solve the problem.

Additionally, in multiple collision domain settings, due to no time-domain contention, we can occasionally also encounter the problem where a new incoming node always observes the spectrum busy and thus, is unable to transmit, which leads to its starvation. In figure 2.4, we illustrate a simple example where this problem can occur. Here, AP_1 and its clients are completely isolated from AP_2 and its clients. Hence, these two groups of nodes

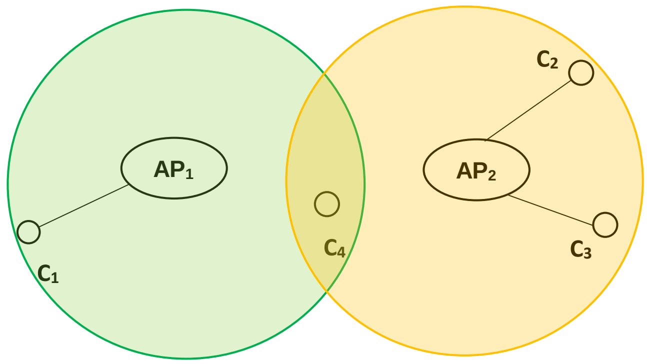


Figure 2.4: A simple setting that shows two isolated groups of nodes (i.e., AP_1 and its client and AP_2 and its two clients), that are not synchronized in terms of protocol stage. A new incoming client, c_4, arriving in the vicinity of AP_1 and AP_2 will always sense the channel busy due to no time-domain contention.

do not necessarily have to be executing the same stage of the FICA protocol (i.e., DIFS, M-RTS, M-CTS, DATA or ACK). The two groups of nodes are not necessarily *stage synchronized*. For example, AP_1 can be transmitting M-RTS and AP_2 might be transmitting data. However, if a new client c_4 arrives at a position in the vicinity of both AP_1 and AP_2, it will observe the channel to be consistently busy and will not be able to transmit an M-RTS, and participate in contention.

In order to solve the above two problems, we develop a novel *frequency-domain stage synchronization protocol*, and we describe it in the Appendix at the end of this chapter. The key idea behind this protocol is to force the entire network to synchronize back to the same initial stage of the protocol. (In the case of FICA and btFICA this is the DIFS stage.) After the protocol execution, all nodes will again observe the same stage and none of the above issues will occur. This protocol is triggered by any node that detects that it is consistently out-of-phase with its surrounding nodes.

Note that, our stage synchronization protocol not only alleviates problems in FICA, but forms a necessary component for the correct operation of other frequency-domain contention protocols, such as, Ez-channel [45] and REPICK [46]. We describe our complete general protocol in the Appendix (Section 2.9).

What happens to btFICA in the presence of *packet capture effect*? In the context of 802.11 networks, packet capture refers to a phenomenon where despite of a collision (i.e., two senders sending simultaneously to a receiver), the receiver is still able to receive and decode the frame corresponding to the stronger signal [62], [61]. This is a natural phenomenon that can cause short-term unfairness in 802.11 networks [62], [61]. In the context of btFICA and FICA, *packet capture* refers to the phenomenon where despite of a collision on a *subchannel*, the receiver is still able to receive and decode the frame corresponding to the stronger signal. However, packet capture effect is not a serious problem.

With btFICA, we not only found the probability of packet capture to be low in our experiments, (close to zero), but also, unlike 802.11 networks, btFICA *does not* necessarily cause channel access unfairness between the senders involved in this phenomenon. This is because of the following reasons.

It should be noted that, in order for packet capture to occur in the first place, we need a collision on a subchannel. With btFICA, several conditions need to be satisfied before a collision can occur. Two senders, s_1 and s_2, in the vicinity of a receiver r should not only pick the same subchannel c, but should also select the same subcarrier sc on c, *and* sc should be selected as the winner by all the potential receivers in the vicinity of *both* s_1 and s_2, in order for a collision to happen on c at r. Now, it should be noted that not all collisions will necessarily lead to packet capture. For example, for packet capture to occur, the stronger signal must satisfy the SINR requirement, and there should be a slight delay in the arrival times of the two signals at r [53].

Now, even if all the conditions for packet capture are met, with btFICA, this can lead to false positives, (i.e., a sender whose packet is not received at the intended receiver, might receive a busy tone, for the needed duration, and thus, falsely conclude that its packet was successfully received.) This will avoid channel access unfairness to occur between senders, because the sender of the weaker signal, seeing a false positive, *does not* reduce its CW. Secondly, in the next rounds of contention both senders can select and win on different subchannels, thus, packet capture and false positives will likely not occur consistently at a sender. Hence, with btFICA, packet capture is not only infrequent, and usually does not have a severe impact on network throughput or fairness.

2.6 Performance Evaluation

Now we evaluate and compare the performance of the FICA MAC protocol, btFICA MAC protocol and 802.11 DCF, in high data rate WLAN settings, and under a variety of network topologies and traffic scenarios.

2.6.1 Simulation Methodology

We have implemented a detailed event-based simulator for each of these 3 protocols. Our simulators also carefully capture the details of the FICA PHY layer, such as CP lengths, subcarrier widths, symbol time, etc. Note that the 802.11 standard has a different PHY layer than that of FICA, which will naturally cause a slight mismatch between the possible PHY data rates of 802.11 and FICA [102]. However, since our goal is to isolate and compare the benefits that can be provided by the *MAC* protocols, over the *same* high PHY data rate, we find it important to maintain the same PHY, (FICA PHY), for all the MAC schemes. Note that, for 802.11 DCF, every node treats the entire wide channel as a single entity.

We have a 160 MHz wide channel. We use the QPSK modulation with 1/2 coding rate on each subcarrier, with 8 MIMO antennas, to give us a PHY data rate of 1.05 Gbps on the entire wide channel. For FICA and btFICA, we have 16 data subcarriers per subchannel as in [102], which gives us a total of 128 subchannels. For btFICA, each BT channel and each guardband that goes between adjacent BT channels, is 4.5 KHz wide. Thus for btFICA, we allocate 0.7% of the wide channel (i.e., 1 subchannel) for busy tone operations, and use the remaining 127 subchannels for the data channel.[13]

Also in FICA PHY, we have constant power per active subcarrier across all nodes. Hence, unlike the cases in [86], [36], in our case, the interference($50m$) and transmission($45m$) ranges of nodes remain the same, even if the nodes send data on only a portion of the wide channel. Our timing parameters, such as SIFS, slot time, etc, are the same as in [102]. To make conditions favorable for the FICA MAC protocol, we have used the $AIMD$ backoff scheme.Since with 802.11 DCF, the packets will take a small transmission time, we turn off RTS/CTS in order to achieve better efficiency [33]. The maximum number of retransmissions for a packet is 7.

2.6.2 Simulation Results - Sample Networks

In this Section we consider 3 different scenarios, in each of which FICA faces only 1 of the 3 problems described in Section 2.3. This will allow us to quantify the isolated impact of each of the problems on the performance of the FICA MAC protocol. Additionally, we will be able to quantify the

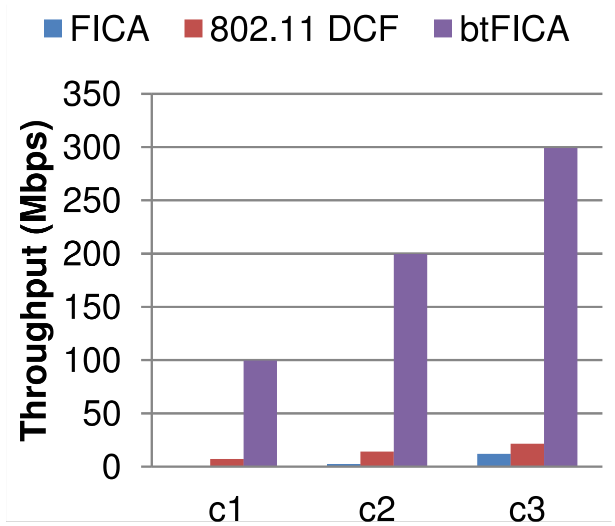
[13] Note that, we have also tried allocating upto 3% (i.e., 4 subchannels) of the wide channel for BT operations. While we observed a slight reduction in throughput for the case of all equal packet sizes in the network than FICA, our approach still yielded significantly better performance when packets of different sizes were present in the network.

improvement in performance that btFICA can provide in each of the individual cases. For completeness, we later also show results for single-cell and multi-cell *random* networks.

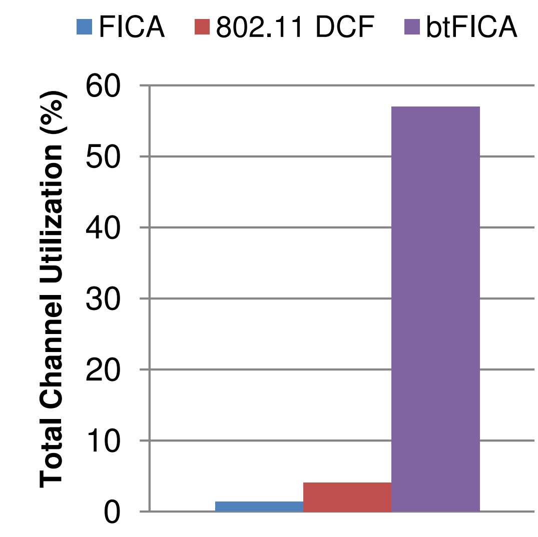
2.6.2.1 Scenario 1

We consider the scenario shown in fig. 2.1. Our choices of packet sizes are motivated by the study done in [92]. We assume only downlink traffic, and that the AP always has packets to send to all its clients. Here, it is clear that FICA only faces the deafness problem. In this scenario, no collisions are happening in the network.

From fig. 2.5b, we can see that, the deafness problem of the FICA MAC protocol is detrimental enough to cause the efficiency to drop to as low as *1.4%*. Even the worst case of 802.11 DCF, i.e., 802.11 DCF without the packet aggregation, provides 3 times higher channel utilization than FICA. This is because, unlike FICA, 802.11 DCF does not waste the channel with needless retransmissions. Moreover, with 802.11 DCF, the AP keeps its time-domain CW to the minimum value (16) and when it transmits a packet it makes good use of the entire channel. We also show that btFICA significantly improves the performance over both FICA and 802.11 DCF. btFICA provides a *40* times improvement in efficiency over FICA, showing its effectiveness in solving the deafness problem faced by FICA. btFICA also performs 9 times



(a)



(b)

Figure 2.5: Results for the scenario described in Section 2.6.2.1. The effectiveness of btFICA in addressing the deafness problem is shown.

better than 802.11 DCF, because btFICA maintains the strengths that are acheived by frequency-domain contention and fine-grained channel access.

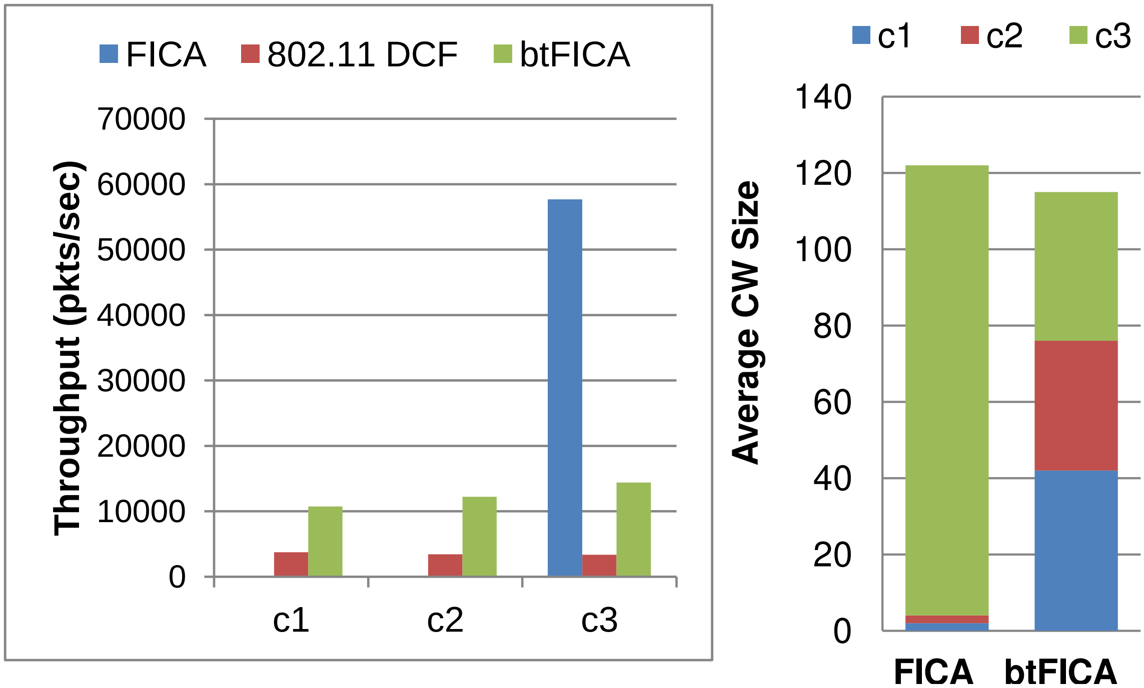
In fig. 2.5a, we show the throughput achieved per-downlink flow. As we expect, with FICA, every flow not only achieves a lower throughput that 802.11 DCF, but also, the flow towards c_1, that contains the smallest sized packets, starves. This is because for the flows towards c_1, deafness occurs more often, leading to more needless retransmissions of c_1's packets. Note that, with 802.11 DCF we are achieving better per-flow-throughput than FICA, and no starvation of flows, because, here, both DATA and ACK packets are received successfully, and thus, we do not face needless retransmissions. Additionally, again, with btFICA we can see that here, the per-flow throughput has improved significantly over both FICA and 802.11 DCF. There is no starvation of flows with btFICA and btFICA provides an equal channel access opportunity for all flows.

2.6.2.2 Scenario 2

We again consider the scenario shown in fig. 2.1, except that now we assume only uplink traffic and that all clients are backlogged, i.e., they always have packets to send. Here, FICA faces only the muteness problem. It is clear from Figure 2.6a that with FICA, all the clients with smaller packet sizes, (c_1 and c_2), starve, but, only one client c_3 that has the largest packet size, is allowed to have a very high throughput. Clearly in this scenario, FICA performs well in terms of channel utilization, but at the cost of starving all but one client in the network. An ideal MAC protocol should avoid starvations and provide a fair but as high as possible throughput to all clients. We can see that 802.11 DCF performs better than FICA in avoiding starvations, and providing fairer throughput for all the clients. However 802.11 achieves a low network throughput and hence, low channel utilization.

In contrast, btFICA alleviates starvation and provides a fairer throughput for each of the clients. We further show fine-grained results in fig. 2.6b. We can see that, with FICA the CW size for c_3 is much larger than that of c_1 and c_2, which shows that on average c_3 accesses almost all subchannels, but c_2 and c_1 access very few subchannels. In contrast with btFICA, the contention window sizes for clients are similar in size. Hence, with btFICA, all clients access almost the same number of subchannels, which leads to better fairness in the network. Thus, btFICA is effective in solving the muteness problem faced by FICA.

It is also worthy to mention that from Figure 2.6a, we can see that the combined throughput of clients, c_1, c_2 and c_3, for the case of btFICA is less than that achieved with the FICA protocol. This is a natural artifact of the AIMD scheme, which allows clients to select subchannels randomly, as permitted by their CW. Here, with btFICA, while the rate of collisions is very low, we find that most of the reduction in throughput happens, because,



(a) (b)

Figure 2.6: Results for the scenario in Section 2.6.2.2. The effectiveness of btFICA in addressing the muteness problem is shown.

the AIMD scheme results in more cases of empty subchannels that no client selected during the contention phase. Note that, the same phenomenon would have occurred with FICA as well, if the muteness problem had not occurred, (so, for example, if we had packets of same sizes for all 3 of the clients). On the other hand, in this example, FICA is achieving a higher (but unfair) throughput, because, along with low number of collisions we don't face many empty subchannels during transmission rounds. This is because, only one of the clients has a very large *CW* and it occupies most of the subchannels, while unfairly starving the other two clients. Hence, in this scenario, with btFICA, while we pay the cost of lower throughput than FICA, we achieve channel access fairness between clients instead.

2.6.2.3 Scenario 3

Now we consider the scenario shown in fig. 2.3. We assume that both the APs are backlogged. We have only downlink traffic, and both AP_1 and AP_2 have packets of sizes 500 bytes and 1500 bytes, respectively. Here, the deafness and muteness problems are not arising, but the hidden terminal problem, described in Section 2.3.3, is occurring with FICA. Both the APs begin contention simultaneously. It is clear from fig. 2.7a, that the impact of the hidden terminal problem is severe enough to cause the efficiency of FICA to drop very close to 0. We have already explained the reason for why we get such results in Section 2.3.3. We also see that 802.11 DCF, even without RTS/CTS, can still give an efficiency of 4%, which shows that this type of hidden terminal problem does not affect 802.11 DCF performance as much as it does FICA's.

In contrast, with btFICA we achieve an efficiency of approximately 120%, which is a significant improvement over the other two schemes. Note that, in this scenario if both the APs transmit simultaneously, on the same subchannels their transmissions will still be successful, because of the way the clients are positioned. btFICA is achieveing a high efficiency because, btFICA is constantly allowing both the APs to use all the subchannels simultaneously, while correctly informing both the APs of the successful packet receptions. In fig. 2.7b, we can see that with btFICA both APs maintain the maximum CW size. In fig. 2.7c, we can see that btFICA gives significantly higher per-AP throughput, while maintaining channel access fairness amongst the two flows. In contrast, FICA is starving both flows, despite that they have high demands. The two APs starve with FICA, because, the consistent occurance of the hidden terminal problem, not only causes many needless retransmissions, but also it forces only one AP to access only 1 subchannel, from the 128 free subchannels, in each transmission round. Clearly, btFICA is effective here in solving the hidden terminal problem.

2.6.3 Simulation Results - Single Cell Random Networks

We now study the 3 MAC protocols in random network topologies and random traffic settings. We have one AP and we place clients on random locations that are within the the AP's transmission range. We consider cases with one packet size, of 1500 bytes, in the entire network, as well as cases with 3 , 6 and 12 different packet sizes in the network, respectively. All packet sizes are in the range of 100 bytes to 1500 bytes.

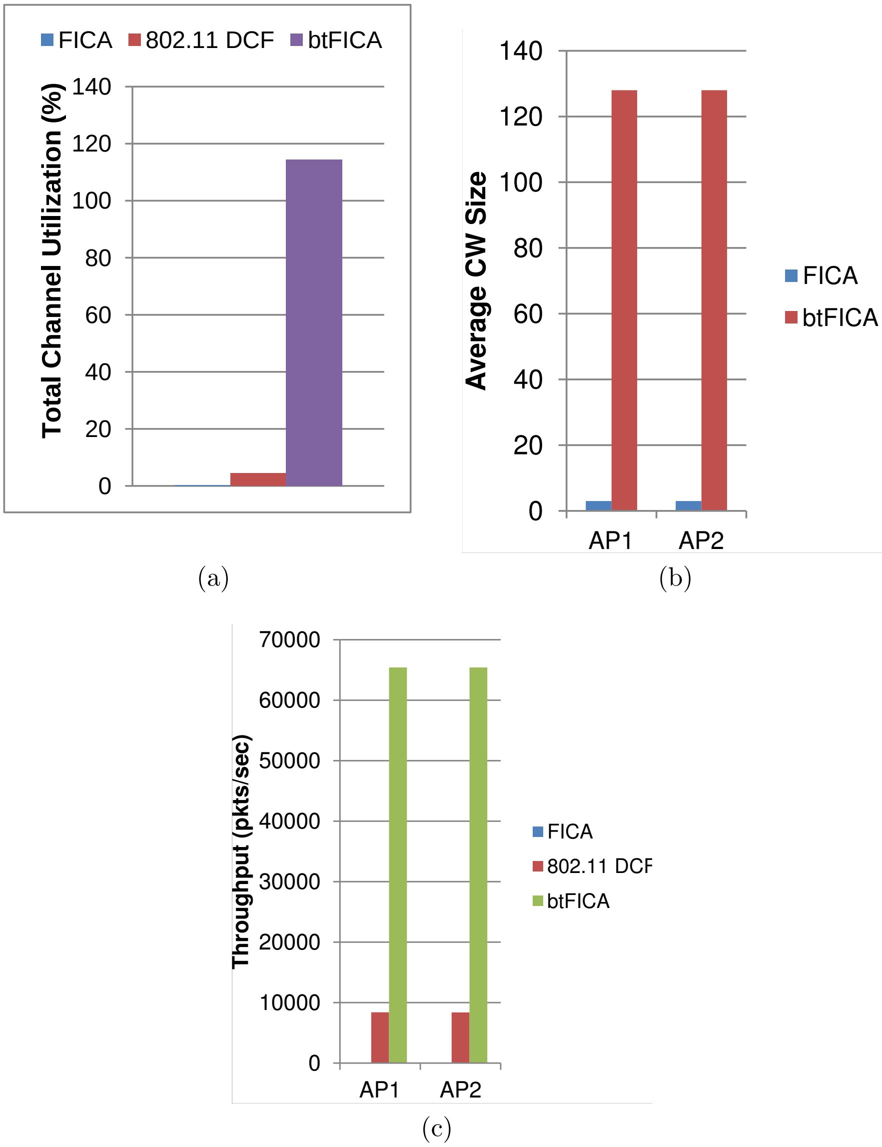
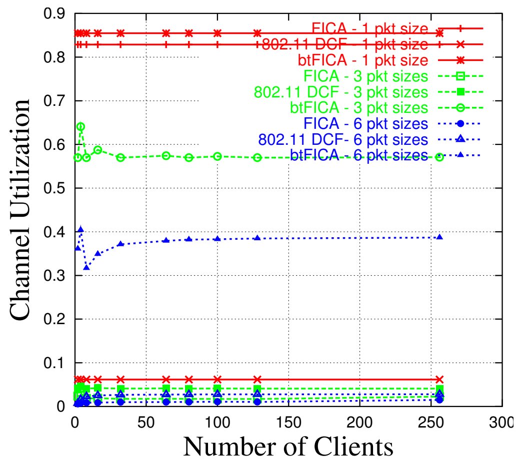


Figure 2.7: Results for the scenario described in Section 2.6.2.3. The effectiveness of btFICA in addressing the hidden terminal problem is shown.

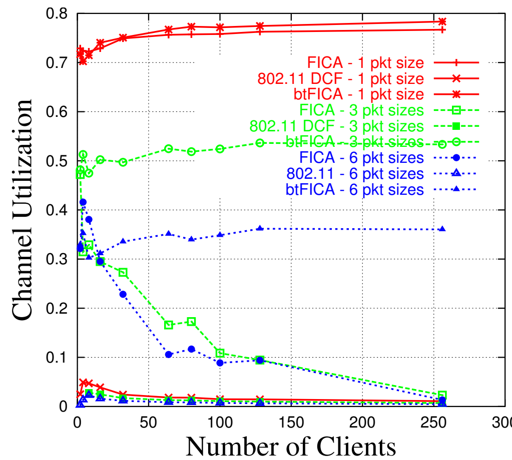
In figures 2.8 and 2.9, we assume that the AP is backlogged, which means that the AP always has packets to send to all its clients. For the scenarios where uplink traffic is present, we assume that every client is backlogged. We find it important to evaluate performance under such settings, because, it tells us how well the MAC protocols make use of the available channel, and how well do they serve all the flows, when there is a high need for the available bandwidth.

In fig. 2.8 for every link in the network, we randomly choose a *fixed* packet size, from the set of packets sizes for the network. Now, in fig. 2.8a, we only have downlink traffic in the network, and we plot the efficiency achieved with each MAC protocol, over varying number of clients. We see that when we have one packet size for all flows, FICA provides an efficiency of 83% which is a substantial improvement over 802.11 DCF. btFICA also performs very well here, showing that btFICA *maintains* the positive features of FICA. Note that, here, btFICA performs slightly better than FICA, because, btFICA does not incur overheads due to ACK preambles.

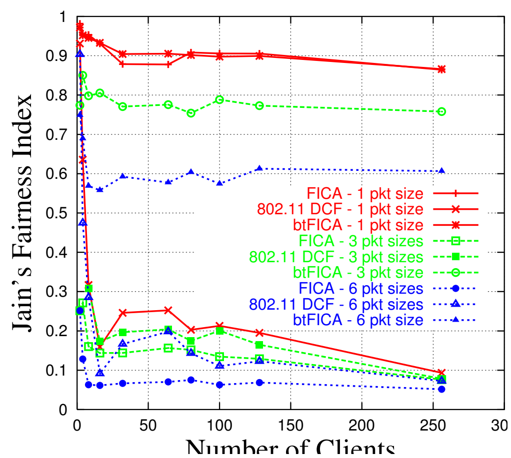
However, we show that when we have different packet sizes in the network, the efficiency for FICA drops drastically, due to the deafness problem. For example, with as little as 3 different packet sizes, FICA's efficiency drops to as low as 1%, which is worse than the 6% channel utilization that we have with 802.11 DCF.



(a) Only downlink traffic is present.



(b) Both Downlink and Uplink traffic is present



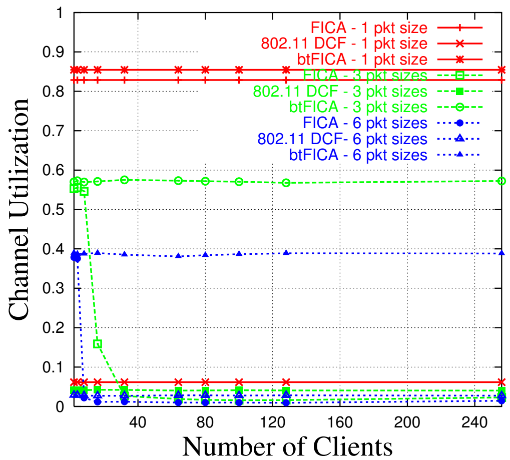
(c) Jain's fairness Index for when both Uplinks and Downlinks are active.

Figure 2.8: Results for one AP and randomly placed clients. For every flow we randomly choose a *fixed* packet size.

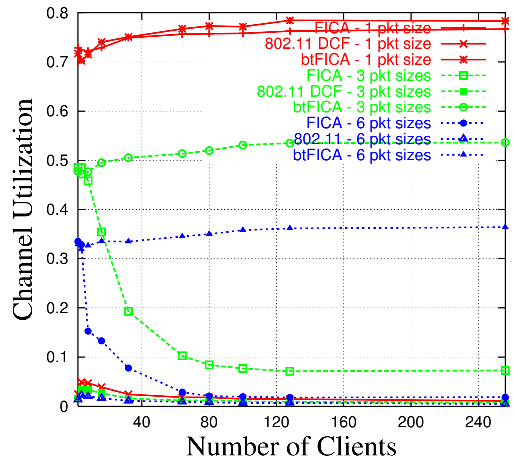
On the other hand, btFICA achieves a 57 times improvement over FICA, even with increasing number of clients. btFICA also achieves significant improvement over 802.11 DCF. We get similar results when we increase the different number of packet sizes in the network to 6 and 12, respectively.

In fig. 2.8b and 2.8c, we use the same topologies as in fig. 2.8a, however, now we have both uplink and downlink traffic. The packet size for each downlink and each uplink is randomly chosen and fixed. In fig. 2.8b we again plot the efficiency and see that when we have different packet sizes in the network, btFICA provides much better efficiency over FICA and 802.11 DCF, for different number of clients. Note that while FICA provides better efficiency than 802.11 DCF, our analysis shows that FICA gives such improvement, at the cost of starving uplink flows, that contain smaller packet sizes, and giving uplink flows with the largest packet size a larger share of the bandwidth. We can also see that for a given number of packet sizes, as we increase the number of clients, FICA's efficiency starts to drop, because now we have more clients with smaller packet sizes winning a subchannel in a round, thus causing more subchannels to be wasted due to the muteness problem, during uplink data frame transmissions. Additionally, the deafness problem also reduces the efficiency of FICA.

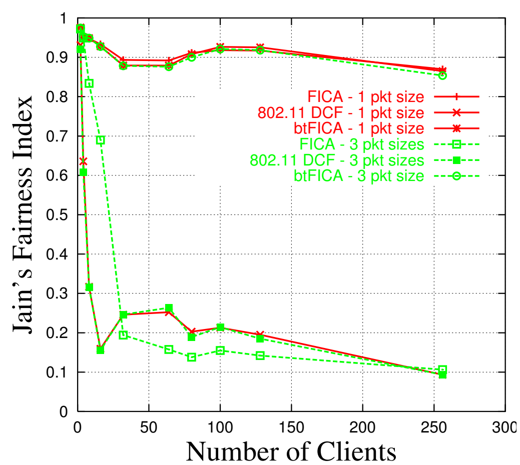
To further verify our claims, in fig. 2.8c, we use the Jain's Fairness Index [60] to compute the level of throughput-fairness amongst all flows in the network. Here, a value close to 1 indicates a high level of fairness. We see that FICA, with one packet size in the network, performs very good interms of fairness. However, as expected, FICA's throughput-fairness drops to very low values, when we have different packet sizes in the network. In contrast, btFICA maintains a much higher level of fairness amongst flows, even in



(a) Only downlink traffic is present.



(b) Both Downlink and Uplink traffic is present



(c) Jain's fairness Index for when both Uplinks and Downlinks are active.

Figure 2.9: Results for one AP and randomly placed clients. For each flow we have *variable* packet sizes that are randomly chosen.

the presence of different packet sizes and even as we increase the number of clients. For example, for the case of 16 clients and 3 different packet sizes in the network, the fairness Index for btFICA, 802.11 DCF and FICA are 0.8, 0.17 and 0.14, respectively. Clearly, btFICA provides noticeable gains in terms of both efficiency *and* fairness.

The results in fig. 2.8 were for the case where on each link we only had packets of a *fixed* size to transmit. However, in real-world settings a flow can also consist of packets of *different* sizes. Thus, in fig. 2.9, we evalute the three schemes under such settings as well. Clearly, btFICA substantially outperforms FICA and 802.11 DCF interms of both efficiency and fairness, again. We also find it important to evaluate the performance of the three schemes in settings where we do *not* necessarily have backlogged traffic. In fig. 2.10, for every uplink and downlink, we choose the data arrival rate for the sender, randomly, from the range of 800 Kbps to 200 Mbps. To every link, we randomly assign a packet size. We also vary the number of clients in the network and we can see that the efficiency of btFICA still remains better than both FICA and 802.11 DCF.

2.6.4 Simulation Results - Multi-Cell Random Networks

We have also evaluated the 3 MAC schemes in a wide variety of multiple AP settings and traffic scenarios, and we show some of our results here. In all cases where we had different packet sizes in the network, we found btFICA providing better efficiency than both FICA and 802.11 DCF. In fig. 2.11, we have 6 APs

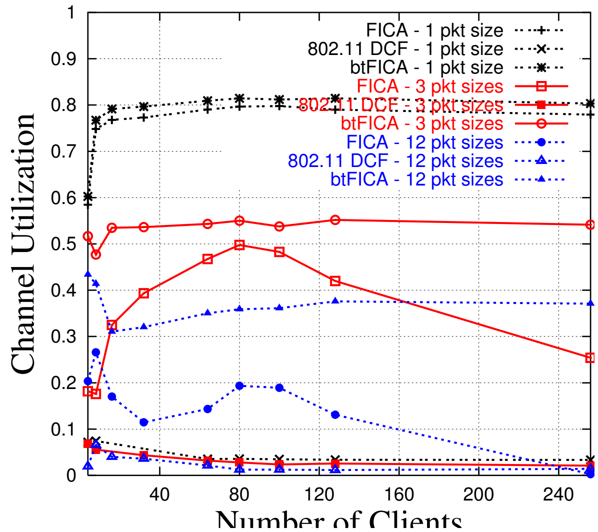


Figure 2.10: Single cell setting with arrival rates for flows chosen randomly.

randomly deployed within a 200m x 200m area, and we change the number of randomly placed clients in the network, and plot the efficiency for each case. Each client is associated with that AP from which it received the strongest signal. We assume only downlink traffic with backlogged APs. Each flow contains packets of different sizes that are randomly chosen.We can see that FICA gives very high efficiency (greater than 100%), when we have only one packet size in the network. This is a significant improvement over what can be achieved with 802.11 DCF. However, as we expect, we can see that with packets of different sizes in the network, the efficiency of FICA reduces to low values as we increase the number of clients. Note that, for very small number of clients and different packet sizes in the network, FICA still achieves a high channel utilization. This is, because in these scenarios, on average, the APs will usually

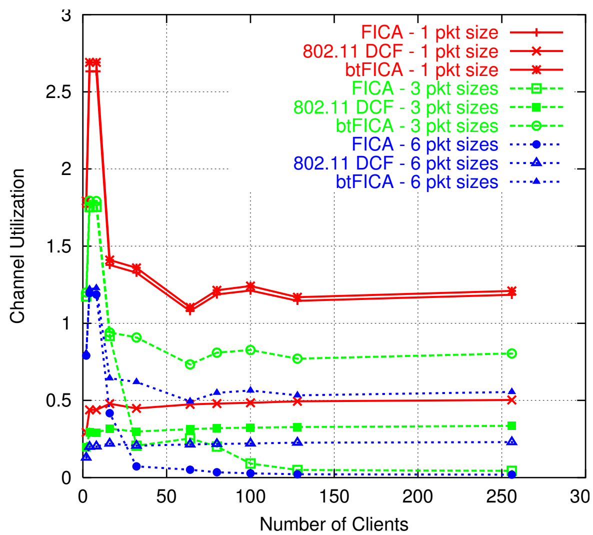


Figure 2.11: Results for WLAN with 6 APs deployed in a 200m x 200m area.

have only one client associated to it, and thus, the chances for the problems discussed in Section 2.3 to occur is low. Here again, we show that btFICA can provide better efficiency than FICA and 802.11 DCF. For example, for the case for 3 packet sizes in the network and 32 clients, for FICA we acheive an efficiency of 21%. In contrast, with btFICA we achieve an efficiency as high as 90%. Note that btFICA can in some cases give an efficiency that is greater than 100%. This is indeed possible, because in multiple collision domain networks, non-interfering links can reuse any portion of the entire wide channel.

2.7 Related Works

In [92], the B2F MAC protocol is proposed for reducing channel contention overheads. While B2F can provide a better efficiency that 802.11 DCF at high PHY data rates, B2F achieves a lower efficiency than btFICA. This is because, unlike btFICA, in B2F, the entire wide channel is treated as a single entity, which causes the DIFS period and the frequency-domain contention period to still be followed by a relatively small period of data transmission. Also, in multiple collision domain networks, with B2F we can face starvation of nodes. Moreover, the B2F scheduled transmissions option [92], for improving efficiency, can easily become ineffective in multiple collision domain networks.

In [68], a new PHY/MAC scheme, WiFi-Nano, is proposed, which reduces the slot size from the standard 9 μsec to 800ns, in order to lower overheads due to time-domain contention. However, WiFi-Nano does not increase efficiency by a large amount at high data rates, e.g., efficiency is 16.7% with 600 Mbps PHY data rate [68], whereas btFICA provides a much better efficiency under the same settings. This is because in WiFi-Nano the preamble overheads are still substantial. In contrast, btFICA masks the effect of preamble time by overlapping the preamble transmissions of multiple data packets in time, and following them by relatively long periods of data transmissions on subchannels. Also, there is no preamble overhead with btFICA's ACKing scheme. Also, WiFi-Nano depends upon IdleSense [56], which is not defined for multiple collision domain networks, thus making the performance of WiFi-Nano under such settings, unclear.

Moreover, the Contention Window tuning schemes [56], [35], proposed for 802.11 networks, cannot improve efficiency in high data rate WLANs significantly.

Finally, as discussed in [102] and [125] the 802.11 DCF's packet aggregation scheme is not as practical for improving efficiency in high data rate WLANs as btFICA. This is because, as we shift to higher data rates each individual sender becomes less likely to be able to aggregate enough packets to enhance the overall channel utilization. Also, as shown in [69], even if a sender can transmit many packets back-to-back, 802.11 DCF will enhance efficiency but at the cost of lowering fairness amongst nodes. This is in contrast to btFICA, where we can maintain both high efficiency *and* fairness in the network.

2.8 Conclusion

In this chapter, we have extensively studied the FICA MAC protocol, which is one of the leading schemes designed for the purpose of improving efficieny in emerging high data rate WLANs. We have identified, for the first time, the problems that can easily arise with the FICA MAC protocol, when packets of different sizes are present in the network. We have quantified the impact of these problems on the performance of FICA via extensive simulations. We showed that these problems can severely degrade channel utilization and fairness in the network, if left unaddressed.

The insights achieved here motivated us to the develop a new MAC protocol, btFICA, for improving efficiency in future high data rate WLANs. btFICA is based upon the FICA framework, and uses an additional busy tone antenna. btFICA effectively and comprehensively addresses all the three problems that arise with the FICA MAC protocol, while maintaining the positive features of the original FICA scheme.

We have shown, the superiority of btFICA over both FICA and 802.11 DCF, in terms of channel utilization, per-user-throughput and fairness. Our results show that btFICA can improve channel utilization in emerging WLANs by upto 40 times when compared to the original FICA scheme.

2.9 Appendix

2.9.1 Stage Synchronization Protocol

With frequency-domain contention protocols, nodes pass through a sequence of stages. Examples of stages in the FICA protocol are the DIFS, SIFS, Data transmission, ACKs, M-RTS and M-CTS stages.

Note that, we can achieve stage synchronization across nodes in the network, by using either out-of-band solutions or an in-band-solution. If we use out-of-band solutions, such as equipping each node with a GPS [87,98], then we will not incur any additional synchronization overheads. The GPS will allow us to achieve accurate time-synchronization, and thus, we will be able to easily synchronize nodes interms of stage. However, this is a costly solution and GPS suffers from poor-signal conditions in indoor settings and thus, is not suitable for indoor settings.

Hence, we propose a simple, but, effective in-band solution for achieving stage synchronization among nodes in the network. Our solution is trigerred when a node in a network detects that it is consistently out of phase with its surrounding nodes, due to which its performance is impacted. Our protocol then ensures that all nodes in the network are reset to the same initial protocol stage, (e.g., DIFS in the case of FICA and btFICA).

We describe our general frequency-domain synchronization protocol, that can also be used by other frequency-domain contention schemes such as Ez-channel [45] and REPICK [46] below. There are two main components in our general protocol, namely, *Synchronization at ACK stage* and *Network-wide Synchronization*.

Note that for the case of FICA, our synchronization scheme will only consist of the second component, and it will not be using anything of the first component. Our general frequency-domain synchronization protocol has following two main components described below:

2.9.1.1 Synchronization at every ACK Stage

We reserve a special subcarrier, called, the ACK-SYN subcarrier, that is meant to identify an ACK stage. All the nodes that are already present in the network, will always send a tone on this subcarrier, when they are passing through their ACK stage. This tone is always sent out, regardless of whether the node has been active (transmitting/receiving) in the current round. Any new incoming node, will not begin contention, until it first overhears a tone on ACK-SYN. If it hears this tone, it will synchronize its stage with that of other nodes in the network, by updating its current stage to be the ACK stage. The new node will then be able to participate. While this simple solution is sufficient in many scenarios, next we discuss and resolve its shortcomings.

In non-centralized WLANs and Adhoc networks, we can still have some rare cases, where, just sending a tone on ACK-SYN, is not enough to ensure stage-synchronization in the network. This is because, here, we can have isolated islands of nodes, such that nodes within an island are synchronized, however, nodes across islands are not necessarily synchronized. If an existing node moves to, or a new node arrives at such a position, that it can hear nodes from two isolated islands, then, this node will cause, these otherwise isolated groups to become connected but unsynchronized, as seen in figure 2.4. However, the approach presented above, will not be sufficient for achieving synchronization across all these nodes. We execute the second component of our synchronization protocol, described below, for tackling this problem.

2.9.1.2 Network-wide Synchronization

We have another reserved subcarrier called, STOP, which will be used for causing a domino effect that will stop all activity, network wide.

Initiation and Propagation of tones on STOP: For MAC contention schemes that run our general synchronization protocol, (such as the Ez-Channel and REPICK), if a node n hears (i) a tone on ACK-SYN, but (ii) its current stage is not the ACK stage, and (iii) it is not hearing anything on STOP, then, n will stop all of its activities and initiate a tone on STOP.

However, for MAC protocols, such as, FICA and btFICA, if a node consistently observes the channel to be busy and never finds a chance to transmit an M-RTS, and does not hear anything on STOP, then the node will initiate a tone on STOP. Similarly, if it finds that it persistently does not receive back and M-CTS after an M-RTS transmission, then, it again initiates a tone on STOP.

Any node that receives this tone, must immediately stop any activity, and contineously transmit a tone on STOP. In this way, this tone will reach all nodes in the entire connected region of the network. The node n, that originally initiated a tone on STOP will allow a period of t microseconds, to elapse, before attempting the next steps of the synchronization protocol. This t is a predefined value and should be large enough to ensure that the Stop signal reached the end of the network.

Stopping tones on STOP and Other Protocol Steps: The next protocol operations, allow nodes to infer when they should stop the signal emitted on STOP, and resume their normal rounds. Now the nodes will be synchronized in terms of stage, and they will begin a fresh round starting from DIFS.

Let h denote the maximum number of hops upto which our protocol will be able to achieve stage synchronization. h is a well-known, predefined number, which is constrained by the number of subcarriers that are supported in the PHY layer. For example, fortunately, a large number of subcarriers (512) are supported in a 160 MHz, 802.11ac channel, and hence,we can easily support, for example, $h = 480$ hops. We assume that this is sufficient for synchronizing nodes that are spread across a wide area.

Starting at node n, each hop, i, is identified by a unique subcarrier, s_i . The node n, that initiated the Stop signal will transmit a tone on , s_1, for a period of a slot. Any node that receives a tone on si will transmit a tone on s_{i+1}, again for a period of a slot. This process will continue until the last hop. Note that, during this entire process, all nodes are still constantly sending tones on STOP. The node that receives a tone on s_h will not relay any tone. Every node w in the network already knows of the entire duration that this relaying process will take (h slots). Moreover, by knowing the exact hop at which w is located at from the originator, w knows the amount of time, t_w, for which it should wait, before the entire relay process would finish. Every node, w will stop the STOP signal and switch back to the first stage, (DIFS in the case of FICA), after t_w period. Thus, all nodes in the connected region will now become synchronized.

If we have two or more nodes initiating the STOP signal, around the same time, then we can have the case, where a node w once hears a tone on s_i originated from one node I_1 and later on might hear a tone on s_j originated from I_2. In this case, w will calculate the new waiting time as follows. If the new waiting time, is greater than or equal to what w had before, then, w will ignore this new tone. Otherwise, w will update its t_w, and will relay a tone on s_{j+1}. If w hears two tones s_i and s_j , at the same time, then w will ignore one of the tones, and will update its t_w, in a similar manner. This will ensure synchronization across an enitre connected region in the network.

We have also evaluated the performance of our frequency-domain stage synchronization protocol, with arrival of new nodes, and we observed that it is effective in achieving stage-synchronization amongst nodes. With the first component of our protocol we achieve stage-synchronization with negligible overhead of several microseconds when a new node arrives. Whenever the second component of our protocol is triggered, we can achieve stage-synchronization within a few milliseconds.

Chapter 3

Adaptive Channelization for Efficiency and Fairness in Future WLANs

3.1 Introduction

In the previous chapter we discussed the key features of the FICA PHY/MAC protocol [102] that made it a promising scheme to study in high speed wireless LANs. Prior to the FICA work, in one of our works, we analytically showed that in high speed settings, standard 802.11-like CSMA/CA (DCF) protocols can severly degrade performance when using the entire channel as a single resource [69]. This is because, now, the packet transmission times are significantly small and bandwidth-independent overheads, such as DIFS, SIFS, back-off periods, etc., dominate, leading to a very poor utilization of the available bandwidth.

Furthermore, we also proposed the idea that one way of improving MAC layer efficiency is by splitting the available wide channel into multiple smaller subchannels and then having senders contend and transmit on these subchannels [69]. In such a system, the channel utilization can improve significantly, because on each of the subchannels the packet transmission takes a proportionately longer transmission time, thus, masking the penalty incurred due to the bandwidth-independent overheads, such as, backoff periods.

However, it is important to note that, there are also other potential MAC solutions that appear to improve efficiency in High Data Rate WLANs. These two schemes are the *Extended-Reservation protocol* and the *Pipelining protocol* [117, 118]. These protocols have not been evaluated and compared against a channelization approach in high speed settings. Additionally, the level of fairness that we can attain with all these three schemes, and how they compare with each other, is also unknown, despite that fairness is a very essential metric in any shared network.

In this chapter, we address the above gaps, and we conduct an in-depth study of throughput and fairness in high speed settings for (1) an adaptive channelization technique (2) The Extended-Reservation Protocol and (3) The Pipelining protocol.

It is worthy to mention here that the following other two approaches for increasing efficiency in high data rate WLANs are not practical and thus will not be evaluated. These two approaches are (a) the claim of reducing the slot time so that now the bandwidth-independent overheads, (e.g. backoffs) will take a smaller amount of time, with respect to the data transmission, and (b) increasing the packet size before transmission.

Claim (a) is not practical because, we cannot reduce the slot size to any arbitrary value, due to physical laws and contraints on current electronics. In fact, we cannot reduce the slot size to a value less than what is already mandated by the IEEE 802.11 standard, which is 9 μ seconds [12]. This is a lower bound and the slot size must be atleast the sum of the following parameters: the propagation delay, the transmit-receive turnaround time, the carrier sense time and the MAC processing time [33, 81].

Claim (b) is also not practical, since it has been observed in [64, 65, 69, 120] that for a given channel bit error rate (BER) ,signal strength and modulation , increasing the packet size will increase the packet error rate, which can inturn decrease the overall network throughput. A similar but better scheme than this is the Extended-Reservation protocol which we will be describing below. (Here, a sender can transmit multiple back-to-back packets, rather than one large packet, upon winning channel access.)

In summary we make the following contributions:

- We describe the above 3 schemes, namely, (1) An adaptive channelization scheme, (2) the Extended-Reservation Protocol and (3) the Pipelining protocol, and the incentives behind studying them in high speed wireless networks.

- We develop analytical models for the adaptive channelization scheme with ACKs enabled, as well as, for the Extended-Reservation protocol. In addition to the models, we also build a detailed simulator for all the above 3 schemes.

- We conduct an extensive evaluation of all the 3 protocols, in single collision domain settings, in terms of throughput and fairness, and in specific short-term fairness. We find that an adaptive channelization approach can outperform both the Extended-Reservation protocol, as well as, the pipelining protocol, significantly, when considering both throughput and fairness. Interestingly, we also found that the "optimized" pipelining scheme, in fact, provided higher throughput than the original pipelining scheme, however, at the cost of starving many nodes in the network. The adaptive channelization approach was devoid of such fairness problems.

Note that, for completeness, we also plot results for 802.11-like DCF in order to allow for a baseline comparison against all schemes. Our results and analysis in this chapter motivates using adaptive channelization schemes for efficiency and fairness in high data rate WLANs.

The rest of the chapter is organized as follows: In Section 3.2, we first describe an adaptive channelization scheme (AMC) for studying in high data rate WLANs [69]. We provide our insights on how such a scheme can have the potential to provide both long-term and short-term fairness amongst nodes in the network and why it is important to compare with the other approaches.

In Section 3.3, we describe the incentive behind evaluating the "Extended-Reservation" MAC protocol in a high speed setting. We develop an analytical model, and a simulator for this protocol, and we show that this scheme is significantly limited in terms of both throughput and fairness when compared to the AMC scheme, in a high speed setting.

In Section 3.4, we present the pipelining protocol [117, 118], and the motivation behind studying this protocol in a high speed setting. We develop a detailed simulator for the pipelining protocol. We show that, interestingly, the original pipelining protocol also performs poorly in high data rate wireless networks, in terms of both throughput and fairness. We show that by tuning the pipelining protocol slightly we can acheive an increase in throughput when compared to the original pipelining approach and to 802.11-like DCF with optimal contention window, however, fairness still suffers severely.

The related work appears in Section 3.5 and we conclude this chapter in Section 3.6.

3.2 Adaptive Channelization in High Data Rate Wireless Networks

Assuming single collision domain, in [69], the authors discuss the improvement in aggregate network throughput that can be acheived in high hata rate networks, over the single-channel 802.11-like DCF, by channelizing the wide channel. They discuss the throughput gain when *splitting* the single wide channel into multiple smaller channels of equal width, and *evenly* distributing the contenders across these channels. On each separate channel, the nodes contend using an ordinary 802.11-like DCF protocol with optimal contention window.

The main reason for the increase in the total throughput is that, now with channelization, each *smaller channel* supports a proportionately *smaller data rate*, hence, the packet transmission takes a *longer time* when compared to the wide single channel case. This longer packet transmission time on each channel masks the effect of the bandwidth-independent overheads, and thus, increases total channel utilization.

In [69], an analytical model based on the Bianchi model [33] has been developed in order to compute the aggregate saturated network throughput, $S(k)$, in a network of n nodes and k channels, where on average each channel has n/k nodes contending for transmission. Using the model they show that it is important to *dynamically* divide the wide spectrum, into optimal number of channels based upon network parameters, such as, number of nodes and guardband widths, in order to ensure optimal channel utilization at all times. So, For example, with an *adaptive* channelization protocol, if 0% guardband is required between channels and if there are 10 nodes in the network, then the wide spectrum will be split into 10 smaller channels, if the number of nodes change to 5, then the spectrum will be divided into 5 smaller channels, and so on and so forth.

For the rest of this chapter, such a protocol that divides the wide spectrum into multiple smaller channels, adaptively, based upon network parameters in order to achieve optimal throughput, will be refered to as the AMC protocol, or the "Adaptive Multichannel protocol." It is the property of the AMC protocol that nodes will be distributed *evenly* across the channels.

3.2.1 Models for the Adaptive Multichannel protocol (AMC)

In this subsection we will present the analytical models that we will be using throughout this chapter in order to determine the saturated normalized throughput for the AMC protocol in a single collision domain. In the models below, basic access method is assumed and RTS,CTS, DIFS and SIFS details are ignored in order to avoid distracting details.

3.2.1.1 AMC without ACKs

The Bianchi's model [33] is extended in [69] to find the saturated normalized network throughput for a network of k channels of equal width and n nodes that are distribued evenly across the k channels. The model that gives the saturated normalized network throughput appears below:

$$S_{no-ack}(k) = \frac{P_{tr}(k)\ P_s(k)\ \alpha}{(1 - P_{tr}(k))\sigma + P_{tr}(k)\ \alpha} \tag{3.1}$$

Where, n stands for the total number of nodes, k for the number of channels, B for the total bandwidth, g for the guardband width between two adjacent channels, and T_p for the packet transmission time on the single wide channel, and $\alpha = \frac{k \cdot B}{B-(k-1)g} T_p$. Morever, $P_{tr}(k)$ is the probability of a transmission in a time slot on a channel, and, $P_s(k)$ is the probablity of a successful transmission in a time slot on a channel, and they are both defined in [69].

Thus, equation 3.1 takes in any k and any n, and it gives the saturated normalized throughput obtained in a network of n nodes and k channels. Given equation 3.1, the saturated normalized thoroughput of the AMC protocol with n nodes and no ACKs is given by $S_{amc}(n)$ and is defined as follows:

$$S_{amc}(n) = max(\{S_{no-ack}(k) : k \in \{1, ..., n\}\}) \tag{3.2}$$

Equation 3.2 finds the saturated normalized network throughput attained with the optimal number of channels k^*, for a given network of n nodes.

3.2.1.2 AMC with ACKs

In order to model the saturated normalized throughput of the AMC protocol with per-packet ACK enabled, we have to first modify equation 3.1 to account for ACKs. The following model gives, $S_{ack}(k)$, which is the saturated normalized throughput in a network of k channels and n nodes, with ACKs enabled:

$$S_{ack}(k) = \frac{P_{tr}(k)\ P_s(k)\ \alpha}{(1 - P_{tr}(k))\sigma + P_{tr}(k)\ \eta}. \tag{3.3}$$

where, α is the same as in equation 3.1, and $\eta = \frac{k \cdot B}{B-(k-1)g} \cdot (T_p + T_{ack})$. Here, T_{ack} is the transmission time taken by an ACK packet in the single-channel case. This equation models the saturated normalized throughput for the case where channel time is wasted in the reception of an ACK after each successful transmission. Moreover, the equation also takes under consideration the time wasted in waiting for an ACK which the sender does not receive during a collision. Everything else is defined in the same way as in equation 3.1.

Now, the saturated normalized thoroughput of the *AMC* protocol with n nodes and ACKs enabled is given by:

$$S_{amc-ack}(n) = max(\{S_{ack}(k) : k \in \{1, ..., n\}\}) \tag{3.4}$$

3.2.2 Why and how is AMC fair?

The adaptive channelization scheme, described in Section 3.2, should also be very fair interms of per-node-throughput even within small time frames because of the following four properties that are assured by this sheme: (1) even distribution of nodes across channels, (2) small number of contenders on each channel (3) same-width channels and (4) support for simultaneous transmissions. The even distribution of nodes will on average cause the same number of nodes to contend on each channel. Overall, this will cause all the nodes to face the same level of contention and therefore all nodes will have a uniform probability of accessing their channels. To understand this better, let us look at a simple example. Let us assume that we have split the single channel into two channels, $c1$ and $c2$, both of width $B/2$, where B is the total bandwidth of the wide spectrum. Now let us assume that and we have 8 nodes contending on $c1$ and 2 nodes contending on $c2$. With optimal contention windows for all nodes on each channel, the normalized througput of $c1$ will be the similar to the normalized throughput for $c2$. (Such effects of tuned contention window can also be seen in [33]). However, the per-node throughput experienced by the nodes that are in $c1$ will be 4 times smaller than the nodes that are in $c2$. However, if we move three nodes from $c1$ to $c2$, then both $c1$ and $c2$ will have

5 nodes each, therefore, all the 10 nodes in the network will have an almost uniform per-node-throughput. With the AMC protocol, since the nodes are evenly distributed across the channels, we do not encounter the fairness issue that is explained above.

Moreover, just having even distribution of nodes across channels, is not enough to maintain fairness amongst nodes operating on different channels. The width of the channels have a direct effect in providing uniform per-node-throughput. To see this let us look at an example where the spectrum is divided into three channels $c1$, $c2$, and $c3$ such that $c1$ has a width of $(1/2) * B$ and $c2$ and $c3$ each has a width of $(1/4) * B$. Here, B is the bandwidth of the wide spectrum. Let us also assume that we have 1 node on each channel, i.e., $n1$, $n2$ and $n3$ on $c1$, $c2$ and $c3$, respectively. Since optimal contention window is used, neither of the nodes will experience backoff and they will contineously transmit. The packet transmission for $n1$ on $c1$ is twice as fast as the packet transmission time for $n2$, and $n3$. Thus, for every one packet that $n2$ and $n3$ transmits, $n1$ transmits two packets. Thus, the throughput for $n1$ is twice that of $n2$ and $n3$. However, if $c1$, $c2$ and $c3$ each had a width of $(1/3) * B$, then all the nodes would have experienced the same number of packet transmissions and hence the same throughput. Since the AMC protocol splits the single channel into multiple smaller channels of equal width, AMC will not encounter the above fairness problem.

Now, in terms of long-term fairness, the single-channel 802.11-like DCF provides an equal share of the spectrum to all nodes, eventually. However, if we are interested in fairness within short periods of time, the 802.11-like DCF might not provide as much fairness in the network as the AMC protocol can provide. The reason for this is that AMC maintains a small amount of contention on each channel and the presence of multiple channels allows multiple transmissions to go through simultaneously. For the purposes of understanding, we can consider the following example: For 0% guardband, packet transmission time of 1 time slot on the wide channel, and any n, the AMC protocol, will split the wide channel into n channels and will allow simultaneous transmissions with one transmission per channel. Hence, after the network is stabalized, if we monitor the network for a period of about n time slots, all the nodes have accessed the spectrum equally, and all nodes have transmitted one packet successfully. However, with the single-channel 802.11-like DCF, after monitoring the network for a period of n time slots, we might not see all the n nodes successfully transmit, because within this period, we can have more time spent in channel idle states, and collisions. Since, there can exist nodes that have not transmitted, within this short period, fairness amongst nodes drops for the 802.11-like DCF. Note that based upon the requirements of the network administrator, fairness within such a fine time scale might not be needed, however, the above discussion is just for clarification purposes.

Hence in this Section we discussed the reasons for why an adaptive channelization protocol (AMC), in a single collision domain can perform better than the ordinary 802.11-like DCF protocol, in high data rate networks, in terms of both throughput and short-term fairness. In the coming Sections, we will present results that further authenticates this, and we also find it important to compare the performance of the AMC protocol against other candidate MAC protocols for fast wireless networks.

3.3 Extended-Reservation Protocol in High Data Rate Wireless Networks

The Extended-Reservation approach is similar to the IEEE 802.11n standard's "Packet Aggregation" scheme [12]. Here, a sender contends for the channel in the same way as 802.11 DCF, but after winning the channel, instead of transmitting only one packet, the sender can now transmit a maximum of L packets, back-to-back. Here we refer to L as the *reservation limit*. The period of time that the back-to-back transmission of L packets will take is called the *reservation period*, and this period starts when a node gains access to the channel. A node cannot hold the channel longer than the reservation period. Once the reservation period is over, the sender reliquinshes control of the channel and goes into a random backoff before attempting to transmit the next packet.

This protocol can improve performance in high data rate wireless networks, since once a node wins the channel, a burst of data can be sent, which has a similar effect of transmitting packets of a longer transmission time, but with the benefit of reduced packet error rate. Moreover, the Extended-Reservation protocol also has the advantage of sending packets to different destinations upon channel access.

Unlike the DCF protocol, where there is a random backoff before every packet transmission, the Extended-Reservation protocol (when $L > 1$) incurs only one backoff period before L packets, thus amortizing the bandwidth-independent overhead of backoff over multiple transmissions.

Though the Extended-Reservation protocol appears to be a good candidate for high data rate wireless networks in terms of channel utilization, we hypothesize that for large values of L, fairness can issues arise.

It is also important to note that we can design two different acknowledgement schemes in such a system: (1) A Block-Ack scheme, where a sender after sending L back-to-back packets, expects to receive one ACK packet back, that contains information about which packets were received properly at the receiver, and (2) A per-packet ACK scheme, where the sender after sending a packet immediately expects an ACK back, before sending the next packet.

While scheme (1) above can reduce the channel time wasted due to ACK transmissions, it faces the follwing issues: Firstly, this approach can only be used if all the packets are towards one destination, and mostly APs have multiple packets to transmit to many clients at a time. Secondly, even if we can transmit many packets to one destination, then, as L grows large, while the ratio of channel idle time will decrease, the wastage due to collisions will increase, which negatively impacts network throughput. Without per-packet ACKs, the sender cannot detect if a collision happened, thus it cannot take any remedial action and will continue to transmit all L packets. Alternatively, if per-packet ACK is used in the protocol, senders can detect collisions when an ACK packet does not arrive and they can then immediately release the channel and go into a random backoff in order to prevent any further wastage.

Thus, in order to alleviate channel wastage due to collisions, we assume that per-packet ACKs are enabled in the Extended-Reservation protocol.[1]

So, now, with the ACK scheme enabled, the following are details of how the Extended-Reservation protocol operates: The sender, after gaining access to the channel by winning a contention, reserves the channel for a maximum of L back-to-back Data/ACK handshakes. The sender sends the first packet, waits for SIFS to hear an ACK, if it receives an ACK, it sends the next packet, and this process continues until a maximum of L back-to-back Data/ACK handshakes have been accomplished (Figure 3.1).Here we define a *reservation period* of a sender as the time it takes for L back-to-back Data/ACK handshakes to be accomplished with SIFS between packets. The reservation period begins when

[1]Note that, later in this Section, when we compare against the AMC protocol, we use the same per-packet acknowledgment scheme for the AMC protocol as well, to allow for a fair comparison.

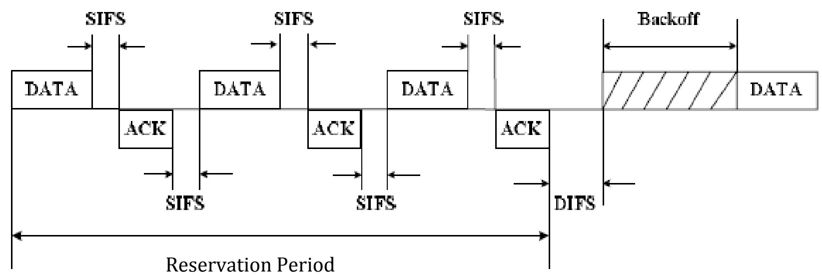


Figure 3.1: The events happening in a network that uses the Extended-Reservation MAC protcol is demonstrated on a time-line. Here $L = 3$. The data packets within the Reservation Period belong to the same transmitter.

the sender gains access to the channel. If at any point within the reservation period, the sender after sending a packet does not receive an ACK within SIFS period, then the sender assumes a collision and releases the channel immediately. The sender will also double its contention window, and contend for the channel again. Other nodes will be able to sense the channel idle for more than SIFS period (i.e.,DIFS), and will be able to continue counting down their backoff counters.

3.3.1 Analytical Modeling of the Extended-Reservation Protocol

We have analytically modeled the Extended-Reservation protocol with ACKs enabled, by modifying the Bianchi model [33]. Given a Reservation-Limit (L), the saturated normalized throughput for this protocol in a single collision domain, is given by:

$$S_L = \frac{P_{tr} P_s L T_p}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c}. \tag{3.5}$$

Here, T_s is the average amount of time for which the channel is sensed busy after a successful transmissison starts, hence, $T_s = L(T_p + T_{Ack})$, where T_p is the transmission time for a packet and T_{Ack} is the time taken by an ACK. Note that $T_p + T_{Ack}$ is multiplied by L to get T_s , because, we are assuming ideal channel conditions and a single collision domain. Hence, once a node starts a successful transmission it will be able to transmit all the L packets.

T_c stands for the average amount of time that is wasted with collisions, and hence, $T_c = T_p + T_{Ack}$. Note that we are not multiplying L here to $T_p + T_{Ack}$. This is because, in a single collision domain, collisions can happen only at the first packet. Thus, after the first packet collides, the collided senders will not send further packets in the reservation and release the channel. Only one Data/ACK exchange time is wasted in collisions.

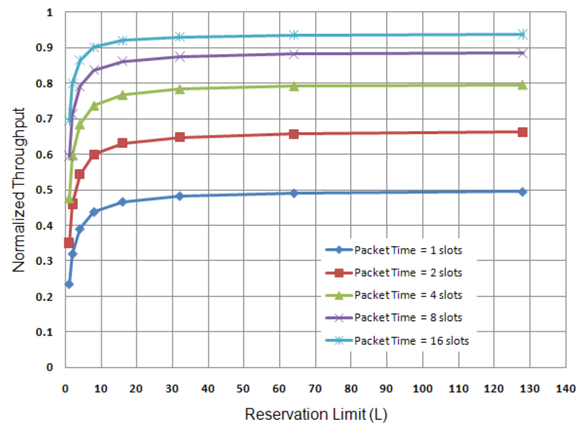


Figure 3.2: Normalized throughput versus Reservation Limit for packets of different sizes. The number of nodes is 25 and optimal contention window is assumed.

Note that P_{tr} is the probability that there is at least one transmission in the time slot under consideration. P_s is the probability that the transmission occuring on the channel is successful. Both of these probabilities are computed using the same formulae in [33].

We use the above model to get an idea about the performance of the Extended-Reservation protocol. In all the results, the contention window has been optimized in order to find the maximum performance benefit that is provided by the protocol. It is important to note that when the Reservation- Limit (L) is 1, then we have the same case as the ordinary 802.11-like DCF.

In Figure 3.2 we have numerically evaluated the normalized throughput of the Extended-Reservation protocol for different Reservation-Limits and packet times. Here, the number of nodes is fixed to 25, L is varied along the x-axis and the change in the normalized throughput, as L varies, is shown for packet transmission times of 1, 2, 4, 8 and 16 slots.

Looking at the graph pertaining to packet transmission time = 1 time slot, (i.e., high speed networks) we can see that as L increases, the normalized throughput increases, and as L becomes very large, the normalized throughput reaches a limit of 0.5. In fact, for packet time of 1 slot, we have an upper bound of 50% channel utilization. This result is as expected, because as L becomes large, we will eventually have a single node reserving a channel and sending packets for a very long time, and since we have a packet time of 1 time slot, half of the long reservation period will be wasted in ACKs, and only half of the reservation period will be spent in useful transmissions. Note that here collisions are not going to cost much, since, if a collision happens then only two slots will be wasted - one for the packet transmission, the other for waiting to receive an ACK, (which the sender does not receive).

As we increase L unboundedly, we will experience an approximately 28% increase in throughput, when compared with normal 802.11-like DCF, for the case of packet time = 1 time slot.

It is important to note that as L increases even though the network throughput reaches the maximum throughput that can be achieved when using the wide channel as a single resource, the network suffers from serious fairness issues. Such high throughput is obtained by starving almost all the nodes except a few. (We will present results showing unfairness in the Extended-Reservation protocol in Section 3.3.2.2.)

Similarly, in Figure 3.2 with packet times of 2, 4, 8 and 16 we can see a similar behavior. Also it is observed that for a given L, as the packet transmission time increases, the throughput also increases. This is because with larger packet times, the effect of the time spent in backoffs and ACKs is hidden by longer periods of useful transmssions.

For all the packet transmission times (i.e., PHY data rates) we can see that the Extended-Reservation protocol with L greater than 1, provides better throughput than 802.11-like DCF.

3.3.2 Extended-Reservation Protocol Vs. AMC and 802.11-like DCF

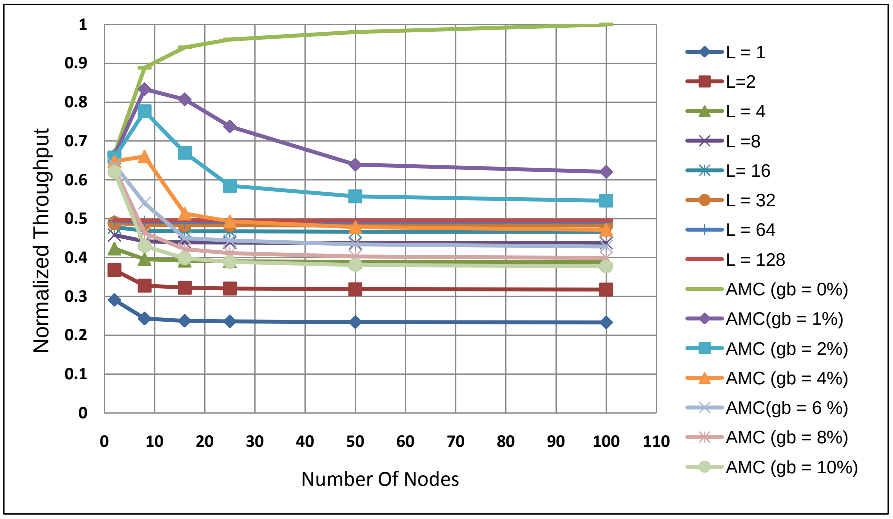
In the discussion below, we analytically compare the Extended-Reservation protocol with the AMC protocol and the 802.11-like DCF in terms of both throughput and fairness. Here the ACKs are enabled for all the protocols and equations 3.4 and 3.5 are used in order to obtain the saturated normalized throughput for the AMC protocol and the Extended-Reservation protocol, respectively. Moreover, optimal contention window is assumed for all the protocols.

3.3.2.1 Throughput Comparision

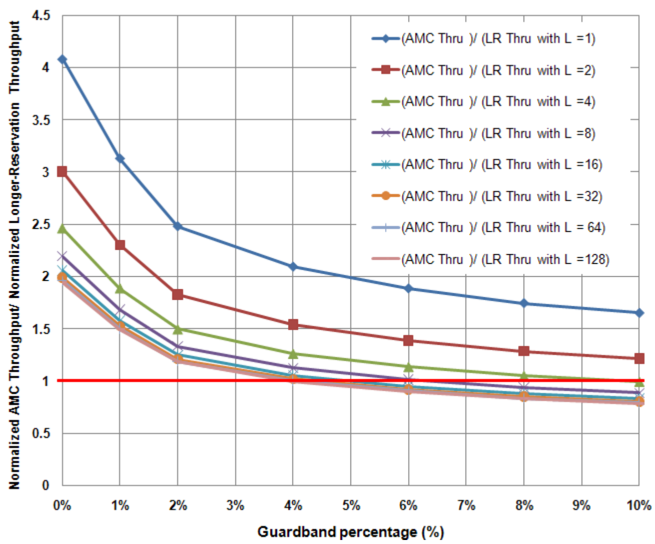
In Figure 3.3a, we are comparing the channel utilization of both the techniques. We vary the number of nodes along the x-axis, and we plot the normalized throughput of the Extended-Reservation protocol with L being 1, 2, 4, 8, 16, 32, 64 and 128. We also plot the normalized throughput of the AMC protocol with guardbands between adjacent channels of width 0 to 10 percent of the total bandwidth. Here the packet transmission time is 1 time slot,(i.e., we are dealing with a high data rate network).

We see that the 802.11-like DCF (case of $L = 1$), gives the worst normalized throughput, for all number of nodes, n. We can also see that as L increases, the normalized throughput for the Extended-Reservation protocol increases,but reaches a limit of 0.5 for all n. (The reason for this has been explained earlier). Also, for a given L , we get almost uniform throughput for all n, because optimal contention window is used in all cases.

Now, if we look at the AMC protocol, we can see that the AMC protocol with realistic guardbands of 0 to 3 percent, even uptil 100 nodes, significantly outperforms the Extended-Reservation protocol, even with large values of L, in terms of channel utilization. Note that when L is large for the Extended-Reservation Protocol, the total network throughput comes close to the ideal throughput throughput that we can achieve in a single-wide channel setting).



(a) Normalized throughput for the Extended-Reservation and AMC protocols Vs varying number of nodes.



(b) Ratio of normalized AMC throughput to normalized Extended-Reservation throughput Vs. varying guardbands. Number of nodes is fixed to 25.

Figure 3.3: Throughput Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot.

The underlying reasons behind these results are as follows: With large L, for the Extended-Reservation protocol, the channel idle time due to backoff and and the channel time wasted in collisions, become negligible. Here 50% of the channel time will be wasted due to ACKs. However, the AMC protocol allows for much better channel utilization, since it proves effective in also masking the ACK overheads. By splitting the single channel into an optimal number of smaller-width channels, the AMC scheme, allows for longer transmission times for packets before encountering an ACK.

Recall that, given n and guardband width between adjacent channels, g, the AMC protocol *splits* the single channel into the best possible number of channels, k^*, that will provide the highest possible throughput, when compared with other number of channels. With small g, we can have very large k^* (i.e., $k^* \approx n$), and at the same time the penalty incurred by the guardbands between channels, is not as much to to cause the channel utilization to drop below 50%(which is the ideal case for the Extended-Reservation protocol).

For example, if $n = 25$, and guardband width between adjacent channels is 1% of the total spectrum width, then, the AMC protocol , will divide the single channel into 25 channels, ($k^* = 25$).The nodes will be evenly distributed across the channels and thus on average there is one sender on each channel. Since we are having optimal contention window for all protocols for a fair comparison, for the AMC case the senders are given a minimum Contention Window of 1. Senders can transmit on the different smaller channels simulatenously, and after a successful transmission only 1 time slot will be wasted due to ACKs for all 25 transmissions, thus, both backoff and ACK overheads are masked. In numbers, the normalized throughput attained for this the AMC protocol and the best case of the Extended-Reservation protocol, for 25 nodes, is 0.76 and 0.5, respectively. It is clear that the AMC protocol with small guardbands of 0% to 3%, provides a higher channel utilization than the Extended-Reservation protocol, even if the latter one is operating with very large values of L.

We should also note that as the guardbands increase beyond 4% the AMC efficiency degrades, since now more portion of the bandwidth is not usable due to large guardbands. To better show the impact of guardbands on the performance of AMC and how it compares with the Extended-Reservation protocol, we now focus on Figure 3.3b. Here, the number of nodes have been fixed to 25, and the packet time in single channel is assumed to be 1. We vary the guardband width on the x-axis, and we plot the ratio of the normalized throughput of the AMC protocol to the normalized throughput of the Extended- Reservation protocol, for different L. We would like to understand that uptil what point does the AMC protocol perform better than the Extended-Reservation protocol. Each of the curves is realated to a separate Reservation Limit. If the curve is above the horizontal line crossing at 1.0, then the AMC protocol is performing better, but if the curve goes below the red line, then the Extended-Reservation protocol with the associated L, is performing better.

We can see that uptil a guardband of approximately 4%, the AMC protocol is always performing better than the Extended-Reservation protocol, even when L is very large. We can see that when AMC protocol operates on a network with huge guardband (uptil 10%), then it still performs better than Extended-Reservation protocols with small Reservation Limits of uptil 4, but performs worse than Reservation-Limits greater than 4. Thus, we can see that the AMC protocol with small guardbands perform better than even the best case of the Extended- Reservation protocol. However, with guardbands of 4% and beyond, we can see that we can find Reservation-Limits, with which the Extended-Reservation protocol would perform better.

However, it is important to note, that guardband widths of 0% to 3% is more than sufficient for preventing channel leakage [37]. Additionally, while the Extended-Reservation protocol with $L \geq 16$ provides a better channel utilization than the AMC scheme with guardband widths of greater than 4%, as we show later it does so at the cost of severely degrading fairness. On the other hand, the AMC scheme does not face any such fairness issues. Hence, again making the channelization approach a better strategy to use in high data rate wireless networks.

It is also important to note here that as the guardbands increase to even very large amounts, the AMC protocol is always better than the single-channel 802.11-like DCF. If for a given n , g is substantial to the point that splitting the channel into two channels will cause a degradation in throughput when compared to the single channel case, then the AMC protocol will chose $k^* = 1$.

3.3.2.2 Fairness Comparision

In the subsection 3.3.2.1,we saw that the Extended-Reservation protocol reaches the optimal single channel throughput as L grows large.We also showed that with guardbands of 4% , the Extended-Reservation protocol with Reservation-Limit of 16 or more, will outperform the AMC protocol. However, now we are going to show that **this higher throughput of the Extended-Reservation protocol comes at the expense of reducing fairness in the network.**

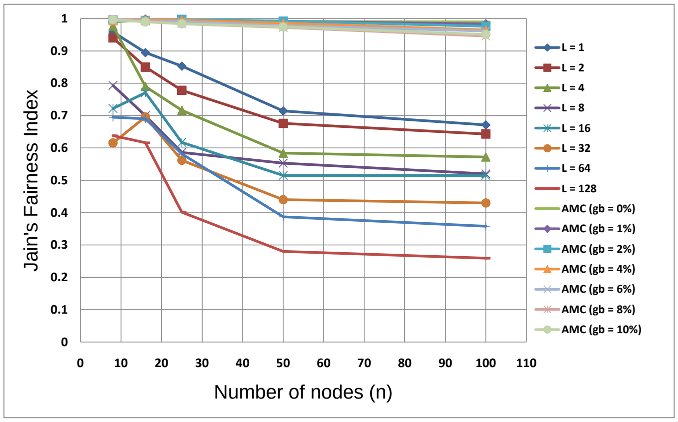
In order to study fairness, we developed a smiulator for the Extended-Reservation protcol and we use the simulator for the AMC protocol in [69].

We are assuming an optimal contention window and packet time of 1 time slot in the single channel setting. We run both the simulators for an equal amount of time (10,000 time slots) and under saturated load. In Figure 3.4a, we vary the number of nodes (n) along the x-axis and we plot the Jain's Fairness Index for the Extended Reservation Protocol with the Reservation-Limit (L) being 1, 2, 4, 8, 16, 32 and 128. We also represent the Jain's Fairness Index for the AMC protocol with guradbands of 0,1,2,4,6,8 and 10 percent. The Jain's Fairness Index [60] is defined below:

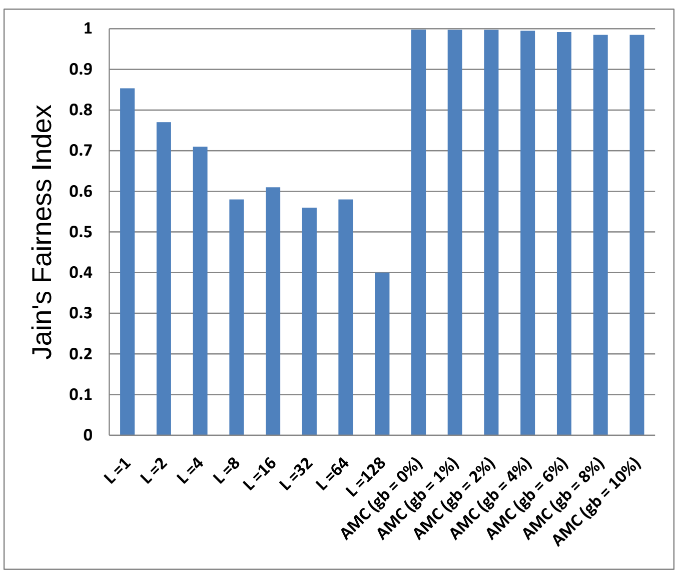
$$Jain'sFairnessIndex = \frac{(\sum_{alli} x_i)^2}{n \sum_{alli} x_i^2}. \tag{3.6}$$

Here, x_i stands for the throughput of node i, and n stands for the total number of the nodes in the network.

If the Fairness Index is close to 1,then the network is in the best state in terms of fairness. On the other hand if the fairness index is close to $1/n$ then the network is in the worst state in terms of fairness. Now, if we look at Figure 3.4a we can see that for the AMC protocol, for uptil 100 nodes, and for all guardband percentages, the Fairness Index is close to 1. This means that even uptil 100 nodes and huge guardbands, every node is given an equal share of the network. In contrast, if we look at the Extended-Reservation protocol, we can see that as L grows large, the Fairness Index drops. In fact, for any L greater than 1, the fairness in the network becomes poorer than the ordinary 802.11 DCF. This happens because, as L grows large, one node transmits packets for even a longer period of time, despite that there are other nodes in the network with the same priority that are waiting to transmit their packets. We can see that when L becomes very large, (and we acheive a high throughput close to the ideal single channel throughput as shown in Figure 3.3a), the fairness index comes closer to the worst case of $1/n$, for a given n. This happens because, as L grows very large, there is just one node that is transmitting, but other nodes starve.



(a) Jain's Fairness Index for the Extended-Reservation protocol and AMC protocol Vs. varying number of nodes. The Extended-Reservation protocol is evaluated for L =1,2,4,8,16,32,64 and 128. The AMC protocol is evauated for guardbands of 1,2,4,6,8 and 10 percent



(b) Jain's fairness index for the Extended-Reservation protocol and the AMC protocol for the case of 25 nodes.

Figure 3.4: Fairness Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot.

Moreover, we can see that as the number of nodes increase, for a given L, the fairness decreases also. The reason for this is that now we have more nodes wanting a share in the network, but one node occupying the channel for L data/Ack handshakes.

Such type of unfairness, as n increases, cannot be seen with the AMC protocol even with large guradband sizes. If we look at the AMC protocol with 0% guardband, as the number of nodes increase, the fairness index remains approximately constant and very close to 1. The reason for this is that with 0% guardband the optimal number of channels is equal to n, and in average one node is assigned a seperate channel. Hence, all nodes get an equal share of the bandwidth, simultaneously. This improves short-term fairness. With guardbands greater than 0%, we can see that as n increases the Fairness Index decreases by a slight amount. The reason for this is that, due to the limitation imposed by the guardbands, we cannot have a channel per sender as n grows large. Therefore, with large n, we have several nodes assigned to the same channel, which causes contention, and we get a slight reduction in fairness. Despite this, we can see that uptil 100 nodes and 10% guradband width the AMC protocol maintains a fairness index of more than approximately 0.95.

In Figure 3.4bwe have fixed the number of nodes to 25, and we have bar graphs that represent the Jain's Fairness Index for the Extended-Reservation protocol and AMC protocol, with different Reservation-Limits and guardband sizes, respectively. As, we can see for all the AMC cases, the fairness index is close to 1.0. Which means that the AMC protocol provides an equal share of the bandwidth to all nodes. However, with the Extended-Reservation protocol, the index value drops as L increses. For example, when $L = 4$, we see that approximately 30% of the nodes suffer from unfairness, and when $L = 128$ we see that 60% of the nodes suffer from unfairness. From Figure 3.3b we can see that for the case of 25 nodes, L needs to be at least greater than 4, in order to provide a better throughput than the AMC protocol with greater than 4% guardband. However, it is undesirable to use the Extended- Reservation protocol, with L greater than or equal to 4 here, since as shown in Figure 3.4b, at least 30% of the nodes will suffer from unfairness.

Going back to figure 3.4a, we can see that when the number of nodes is 50 or more, then even the Extended-Reservation protocol, with small Reservation-Limits become unusable if we are concerned about fairness. We can see that with L =1, and the 50 nodes case, we will have 30% of the nodes facing unfairness, and with L=4 we have 41%, and with $L = 128$ we can see that 72% of the network face unfairness. We can see that L=4 and more becomes undesirable if n is equal to 20 nodes or beyond.

Thus, from our study in this Section, we conclude that in terms of both throughput and fairness, the AMC protocol with small guardband widths (from 0% uptil 3%) is significantly better in terms of both throughput and fairness than even the best case of the Extended-Reservation protocol. It should be noted that it is practical to design and implement radios that could use such small guardbands for preventing channel leakage problems [37].

3.4 The Pipelining Protocol in High Data Rate Wireless Networks

In this section we will discuss the performance of the pipelining protocol in high data rate wireless networks. One of the main underlying reasons for the design of the piplining protocol was to increase the efficiency of ordinary 802.11 networks by reducing the channel idle time. Since channel idle time is the main reason why we have substantial degradation in channel utilization in the high speed regime, therefore, we find it important to investigate the performance of this protocol in the context here.

The authors in [117, 118], have presented four versions of the pipelining protocol, namely, the Dual Channel Pipelined Scheduling Scheme, One-Phase Busy Tone Pipelining Scheme, Two-Phase Busy Tone Pipelining Scheme, and Implicit pipelining. The first three of these schemes use an explicit control channel in order to allow the contention resolution to happen in parallel with the data transmission. In Implicit pipelining, the same idea is implemented however, without using a control channel.

It is obvious from [117, 118], that the pipelining scheme that allows for the best throughput when compared with the ordinary 802.11 protocol, is the "Two-Phase Busy Tone Explicit Pipelining Scheme." Hence, we will choose this version of the pipelining protocol to explain briefly, and then we will evaluate this scheme in a very high speed setting.

3.4.1 Description of the Pipelining Protocol

The basic observation of the authors of [117,118], is that, with ordinary 802.11, the nodes contend for a channel. When a node wins and transmits, all other nodes *freeze* their backoff counters until the transmission finishes, and then all nodes start the contention resolution phase again as seen in figure 3.5(taken from [117, 118]). During the contention resolution phase, the channel can experience idle times that will reduce the network efficiency. However, we can improve efficiency, if *during* the transmission phase, we allow the contention resolution phase before the next transmission to *overlap* in time with this on going transmission. This will allow for lesser channel idle time, when compared with that of 802.11, where the transmission phase and the contention resolution phase happen sequentially. This is the core principle behind the "pipelining" protocol.

In the "Two-Phase Busy Tone Pipelining Scheme", a data and control channel is used. Every contention resolution period is divided into two phases. The first phase happens on the control channel and overlaps, in time, with an ongoing successful transmission that is happening on the data channel, as seen in figure 3.6 (taken from [118]). The second phase of the contention resolution happens on the data channel after the successful transmission finishes.

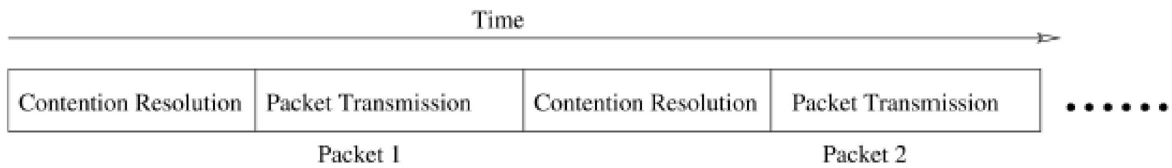


Figure 3.5: Sequence of events happening in a standard 802.11 network.

Nodes that are in phase 1 of the contention resolution are said to be in stage 1. Similarly, the nodes in the second phase of the contention resolution is said to be in stage 2. Unlike the 802.11-like DCF, where, only one contention window is involved, here with this scheme, two contention windows, namely $CW1$ and $CW2$ are involved, one for each stage respectivey. Moreover, both stages have *static* minimum and maximum contention window values. $CW1_{min}$ and $CW1_{max}$ are 32 and 1024, and $CW2_{min}$ and $CW2_{max}$ are 8 and 128, respectively. Moreover, we also have two backoff counters, one for each stage. $bc1$ for stage1 is chosen randomly from the interval $[0, CW1 - 1]$ and $bc2$ for stage2 is chosen randomly from the interval $[0, CW2 - 1]$.

While a successful transmission is occuring, other nodes instead of “deferring” continue reducing their $bc1$, while listening on the control channel. (Here is where pipelining reduces the channel idle time of the network.) When a node wins on the control channel (stage 1), it sends a busy tone on the control channel, then advances to stage 2 where it is called a “pipelined station.” All other nodes that are in stage 1 will freeze their $bc1$ when they hear the busy tone. Once the data transmission finishes, the *pipelined stations* will start the

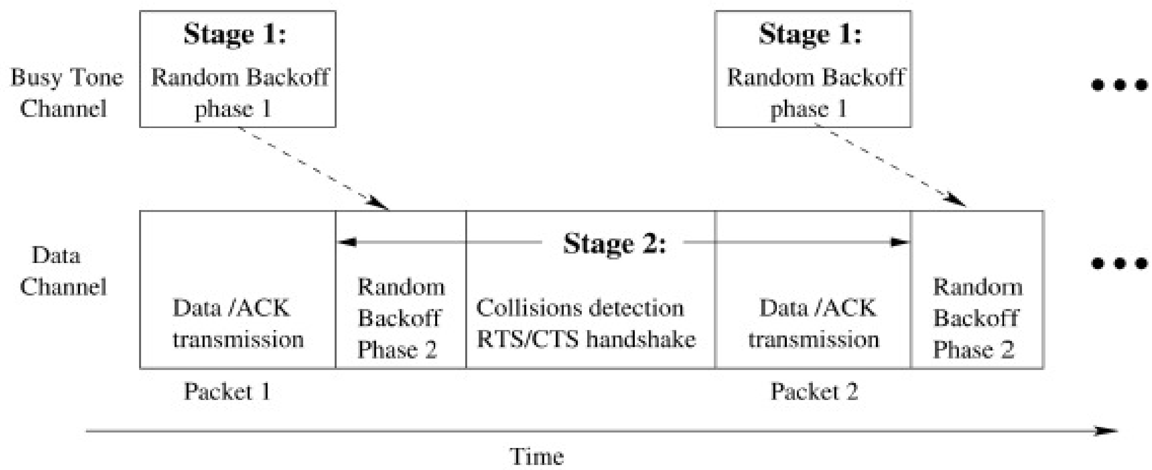


Figure 3.6: Sequence of events happening in an pipelined network.

phase 2 of the contention resolution. (Here is where the pipelining protocol reduces the bandwidth dependent overhead, i.e., collisions.) The phase 2 of the contention resolution continues, until a node wins and starts a successful transmission. Once a successful transmission starts all other nodes resume phase 1 contention resolution.

It is important to note here that the pipelining protocol is designed to provide better throughput than the *standard* 802.11-like DCF, where the contention windows are *static*. Hence, we find it important to first investigate how much channel utilization improvement can be acheived over the standard 802.11-like DCF, if the pipelining MAC layer protocol is used, in high data rate wireless networks.

3.4.2 Pipelining protocol Vs. Standard 802.11-like DCF With Static Contention Window

We developed a simulator for the Two Phase Busy Tone pipelining protocol and the 802.11-like DCF with static contention window. In figure 3.7 we have varied the number of nodes along the x-axis and we have plotted the saturated normalized throughput for the pipelining protocol for different physical layer speeds. The plot related to packet transmission time of 1 time slot corresponds to a high speed network, and the plot related to packet time of 16 time slots refers to a 16 times slower network. As we expect, we can see that as we shift to very high data rate networks, the channel utilization that is acheived by the pipelining MAC protocol degrades drastically. For example, we can see that with packet time of 1 time slot and 64 nodes, the channel utilization with the pipelining protocol is as low as approximately 24%. However, with packet time of 16 time slots and 64 nodes, we can see that the channel utilization is approximately 75%.

The reason for this is that, for *low data rate networks*, the pipelining protocol proves more effective in reducing the channel idle time and the cost of collisions. We notice that, this is because *the packet transmission time itself plays an important role in the effectiveness of the pipelining protocol*.

With low data rate networks, where the packets have a *long* transmission time, a *large* portion of the contention resolution period, (that would otherwise

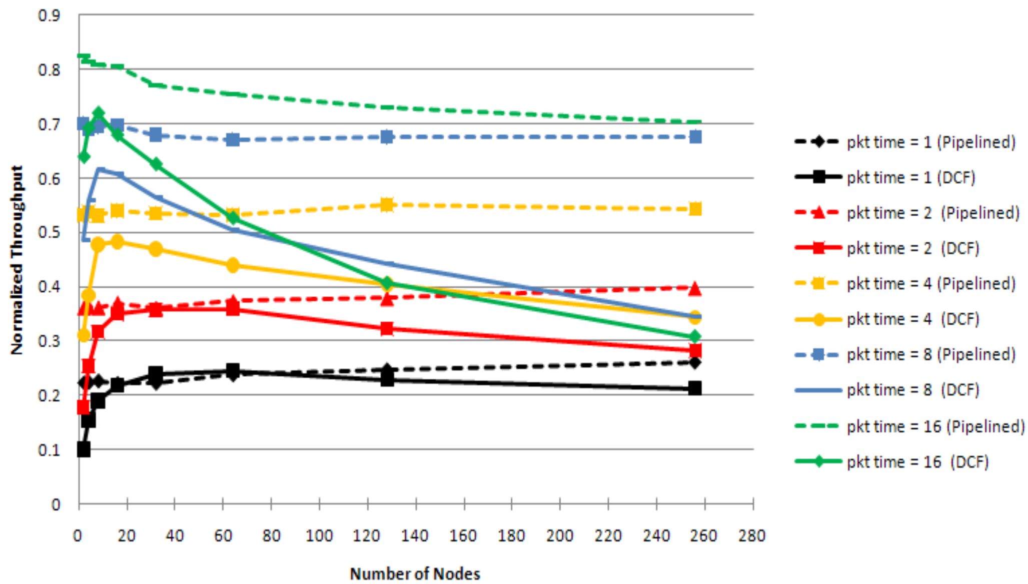


Figure 3.7: Saturated Normalized Throughput Vs. Varying Number of Nodes for the pipelining protocol and 802.11-like DCF with static contention window.

happen after the data transmission and cause channel idle times), is already finished when an on going transmission ends. Moreover, in low data rate networks the cost of collisions is large, since now packets take a longer transmission time. If a collision happens, the senders cannot "detect" that a collision has occured and they continue to transmit the entire packet, thus wasting a *large* amount of the channel time in collisions. The piplining protocol reduces the number of collisions, by allowing the nodes that have reduced their backoff counters to 0 at the same slot to go through another "short" phase (phase 2) of contention resolution. This can add a small period of channel idle time between packet transmissions, however, since the packet times are long, this penalty will be masked. Moreover, the penalty of bandwidth-independent overhead that is incurred by phase 2 of the contention resolution is not as much when compared with the penalty incurred due to collisions. Hence, in low data rate networks, pipelining protocol provides a higher channel utilization.

However, now focusing on high data rate networks, we see that the pipelining protocol performs poorly here, when compared to low data rate networks. Here, the packet transmission times are small and hence, only a *small* portion of the contention resolution period can overlap with the data transmission. Thus, between packet tranmsissions the number of channel idle time still remains substantial. Moreover, since the packet times are small, therefore, the total channel time that can be wasted in collisions is not as severe as low data rate networks. Though the second phase of the contention resolution still proves effective in reducing the number of collisions, however, it can introduce a short period of channel idle time between transmissions. Since now packet transmission times are small, even if the phase 2 of the contention resolution introduces just a "few" additional idle time slots between transmissions, it can still severly bring down the channel utlization.

Thus, the pipelining protocol performs poorly in high data rate networks because it does not prove effective in reducing the bandwidth-independent overheads,(that are introduced due to backoff periods), by a sufficient amount. We further prove this via our results in figure 3.8. Here, again we have used the simulator for the pipelining protocol, in order to find the percentage of channel

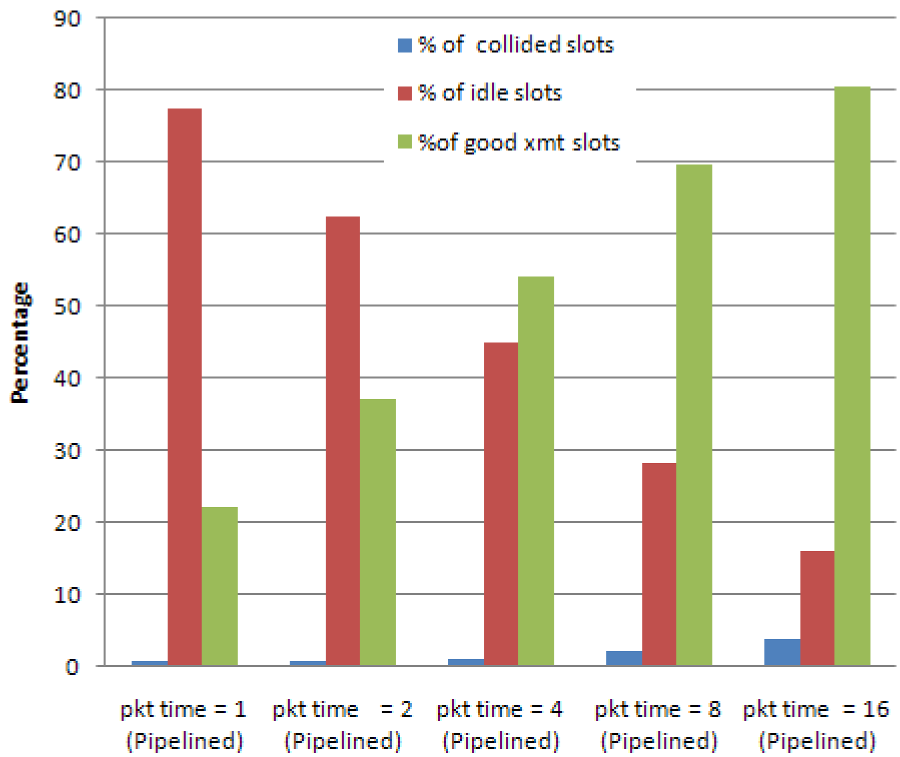


Figure 3.8: Percentages of channel time spent in no transmissions, collisions and successful transmissions with the pipelining protocol for different packet times. Number of nodes is assumed to be 16.

time wasted due to bandwidth independent overheads and collisions, and we also find the percentage of channel time spent in successful transmissions, for packet times of 1 , 4 and 16 time slots. It is clear over here, that for small packet times the penalty incurred due to bandwidth-independent overhead is high. However, as the packet time increases, we can see that the idle time decreases. For example, 77%, 45% and 16% of the channel time is idle in networks with packet transmission times of 1, 4 and 16 time slots, respectively.

We can also see in figure 3.7, that for all different physical layer data rates, with the pipelining protocol we can have a better saturated normalized throughput than the standard 802.11-like DCF with static windows. Moreover, it is clear that as the packet time decreases the channel utilization difference between the pipelining protocol and the standard 802.11-like DCF protocol also decreases. For example,for the case of 128 nodes, if packet time = 16 slots, then we have a 23% difference in channel utilzation of the two protocols; if packet time = 4 we experience a 10% difference between the two protocols; and if packet time = 1 slot, then pipelining provides only 2% better channel utilization than the standard 802.11-like DCF.

Thus, we can see that with high data rate wireless networks, the pipelining protocol does not provide much throughput benefit over the standard 802.11-like DCF with static windows.

3.4.3 Original Pipelining Vs. 802.11-like DCF With Optimal Contention Window

It is shown that the standard 802.11-like DCF with *static* contention window does not always give the best possible throughput [33,34]. However, in a single-collision domain setting and under saturated conditions, theoretically, we can optimize the performance of 802.11-like DCF, by tuning the the minimum contention window, CW_{min}, based on the number of nodes in the network [33, 56].

While attaining the optimal CW_{min} might not always be achievable in practice and can be a complex and an expensive process, from an analytical point of view and in order to get a better understanding, we still find it important to compare the Pipelining protocol with 802.11-like DCF with optimal Contention Window. In this way we will get insights about how Pipelining compares with 802.11 DCF, even when we have the best settings for 802.11-like DCF.

We will be referring to the standard 802.11-like DCF protocol with static contention window, as, *Standard 802.11-like DCF*. On the other hand, we will refer to 802.11-like DCF with optimal contention window, as, *Optimal 802.11-like DCF*.

Figure 3.9 is similar to figure 3.7, except that we have plotted the Optimized 802.11-like DCF's saturated normalized throughput for different packet times and different number of nodes, instead of the Standard 802.11-like DCF. Here, we use the bianchi's model [33] to attain the optimal Contention Window, and to numerically evaluate the saturated normalized throughput of the Optimal 802.11-like DCF for different network settings. We have also plotted the saturated normalized throughput for the pipelining protocol for different packet times. Note that, given a packet time, for Optimal 802.11-like DCF, we observe a similar Normalized Throughput even for different number nodes. This is expected since it is a direct artifact of tuning the Contention Window for achieving the maximum possible throughput for each case.

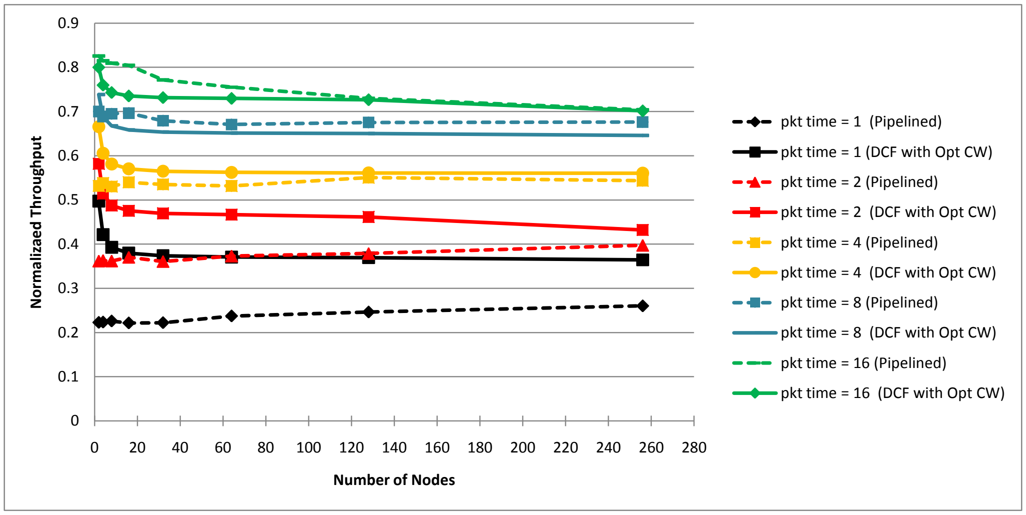


Figure 3.9: Saturated Normalized Throughput Vs. Varying Number of Nodes for the pipelining protocol and 802.11-like DCF with *optimal* contention window

From Figure 3.9, it is interesting to note that in low data rate networks, the pipelining protocol performs slightly better than even Optimal 802.11-like DCF (i.e., cases of packet times of 8 and 16 time slots, respectively).

On the other hand, in high speed networks, we can see that the Optimal 802.11-like DCF provides a higher throughput than the pipelining protocol. For example, we can see that for 32 nodes and the case of packet time = 1 time slot, the pipelining protocol gives a 22% channel utilization, whereas 802.11-like DCF provides us with a 37% channel utilization.

3.4.4 Original Pipelining Vs. Tuned Pipelining

Now, it is clear that for a high speed wireless network, the Pipelining protocol performs worse than the 802.11-like DCF with optimal contention window. However, We make the observation that the reason for the low channel utilization of the Pipelining protocol comes from the *static* nature of $CW1_{min}$ and $CW2_{min}$. Similar to the Optimal 802.11-like DCF, if for the pipelining protocol $CW1_{min}$ and $CW2_{min}$ are dynamically chosen, based upon the network settings, then the channel idle time introduced by the pipelining protocol can be reduced in high data rate wireless networks.

So, we give pipelining a second chance, by using optimal $CW1_{min}$ and $CW2_{min}$ values. We compute the optimal values for the contention windows by executing the simulator with 20 different values of $CW1_{min}$ and $CW2_{min}$, taken from the range [2, 1024], for each n and p. We then pick the maximum saturated normalized thorughput which is obtained by one of the choices of $CW1_{min}$ and $CW2_{min}$. Here, n is the number of nodes in the network and p is the packet transmission time. We call the pipelining protocol that uses optimal contention window in different network settings as the *tuned-pipelining* protocol.

In figure 3.10, we have varied the number of nodes along the x-axis, and plotted the saturated normalized throughput of the tuned-pipelining protocol for various packet times (i.e., PHY layer data rates). Moreover, we have also plotted the Optimal 802.11-like DCF's saturated normalized throughput for different packet times. It is clear here that the tuned-pipelining protocol provides a better normalized throughput than the 802.11-like DCF, for all number of nodes and packet times. For example, for packet transmission time of 1 time slot and 25 nodes we can see that the 802.11-like DCF gives us 37% channel utilization, however, with the pipelining protocol we achieve a higher channel utilization of 66%.

Hence, we showed that tuned-pipelining provides better throughput than the original pipelining protocol and the Optimal 802.11-like DCF in high speed settings. However, we are now going to show that despite the results shown in this section, the Pipelining protocol, as well as, the Tuned-pipelining protocol are not suitable for high data rate wireless networks, when compared to the AMC scheme.

3.4.5 Tuned Pipelining Vs AMC protocol

In this section we are going to compare the throughput of the tuned-pipelining protocol against the AMC protocol.

We use the same simulators as before in order to obtain the saturated normalized throughput for the pipelining protocols. Moreover, we use equation 3.2 and the Bianchi model [33], in order to achieve the saturated normalized throughput for the AMC scheme and Optimal 802.11-like DCF, respectively.

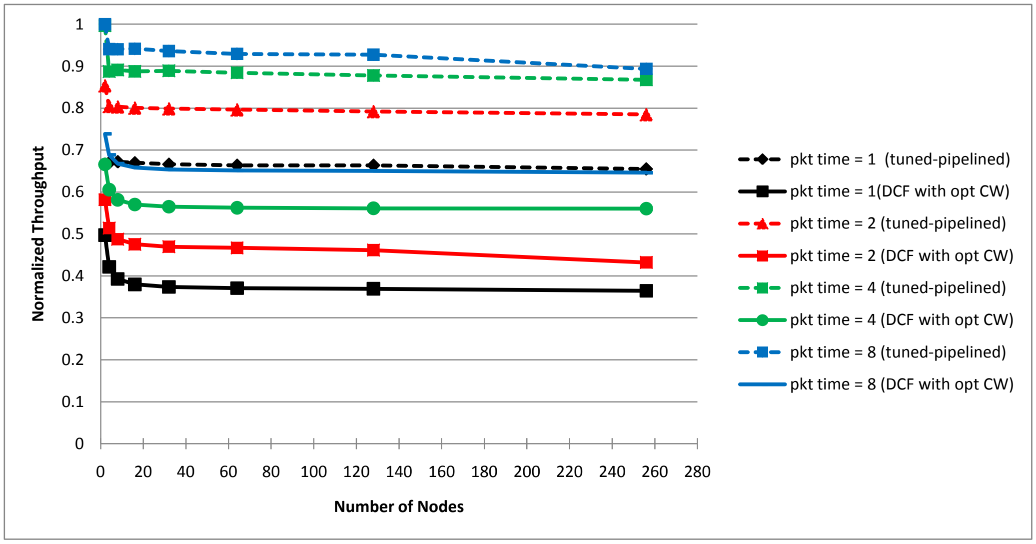


Figure 3.10: Saturated Normalized Throughput Vs. Varying Number of Nodes for the *tuned-pipelining* protocol and 802.11-like DCF with *optimal* contention window.

In figure 3.11 we are comparing the channel utilization of the three protocols, namely, tuned-pipelining, AMC and Optimal 802.11-like DCF, in a high data rate wireless network. The number of nodes are varied along the x-axis and the packet time is 1 slot in the single channel setting. The saturated normalized throughput for the AMC protocol with guardbands of 0 to 7 percent of the total bandwidth between adjacent channels, is plotted. We have also plotted the performance of the original pipelining protocol.

It is clear here, that even with large guardband sizes between adjacent channels, (upto 7% of total bandwidth), the AMC approach significantly outperforms both the original pipelining scheme (with static contention windows)

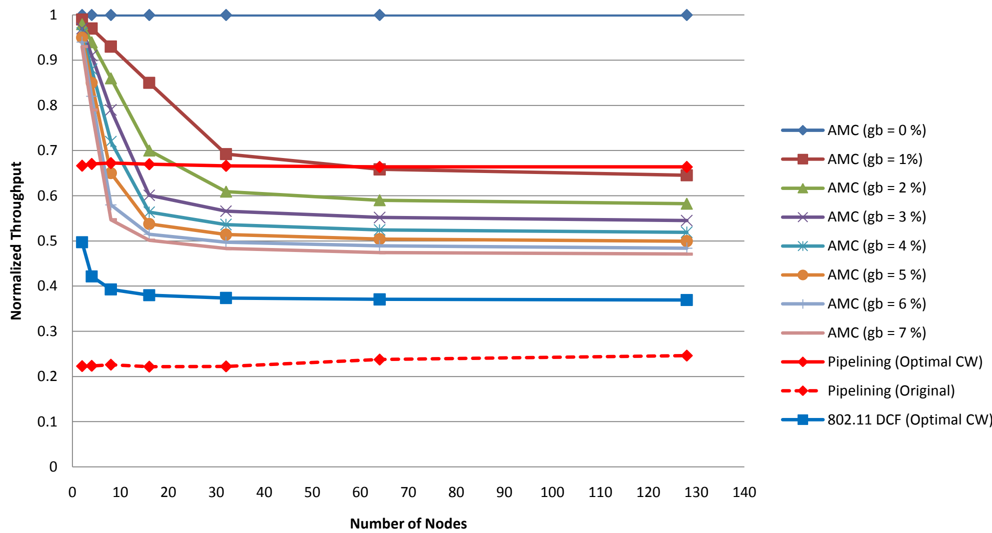


Figure 3.11: Saturated Normalized Throughput Vs. Varying Number of Nodes for the AMC protocol with varying guardband widths, Pipelining protocols and the 802.11-like DCF

and the Optimal 802.11-like DCF. Additionally, we can see that the AMC protocol performs better than the tuned-pipelining protocol, for small guardbands of 0% to 1%.[2]

With guardbands of uptil 1% the AMC protocol wins over both pipelining protocols, because, as explained earlier, the AMC protocol can provide almost each sender a seperate channel and the senders experience close to zero backoff periods, which is further masked by the long transmission times on the individual subchannels. Moreover, here, the total wastage due to guardbands is not as

[2]Real radio implementations show that such small guardband between adjacent channels is in fact more than sufficient for preventing channel leakage between adjacent channels [37].

much to cause the AMC protocol to perform worse than the tuned-pipelining protocol. The tuned-pipelining protocol performs worse because it still faces channel idle time followed by small periods of data transmission, as well as, collision overheads.

We can also notice here that as the width of the guardbands increase beyond 1% then the AMC protocol still wins over the pipelining protocol, with smaller and smaller number of nodes. For example, with guardbands of 4% of the total bandwidth and 8 nodes, the AMC protocol has a channel utilization of 72%, however the pipelining protocol has a channel utilization of 66%.

Thus, we conclude here that with guardbands of uptil 1%, (which is more than sufficient), the AMC protocol proves better than the tuned-pipelining protocol, original pipelining protocol, and the Optimal 802.11-like DCF. With guardbands of greater than 1% but with small number of nodes, we might be able to achieve a better throughput than the pipelining protocol. However, in general as the guardband sizes become greater than 1% , then the tuned-pipelining protocol provides a better throughput in high data rate wireless networks.

However, we are going to show that the tuned-pipelining protocol causes severe unfairness between nodes, the like of which we do not experience in 802.11-like DCF or in the AMC protocol, thus, making it an unsuitable protocol for high data rate wireless networks.

3.4.6 Fairness Comparision of Tuned-pipelining protocol, 802.11-like DCF and AMC

Our fairness evaluations, reveal important and interesting insights into the behaviors of the Tuned-Pipelining scheme, as well as, the AMC protocol. In order to measure the Jain's Fairness Index (i.e., Equation 3.6), we again use our simulators for the tuned-pipelining protcol and the 802.11-like DCF. The simulator in [69] is used to measure the fairness index of the AMC protocol.

In figure 3.12, we have plotted the Jain's Fairness Index for the tuned-pipelining protocol, 802.11-like DCF, and the AMC protocol with different guardband widths in a network setting of 64 nodes and packet transmission time of 1 time slot in the single channel setting. Moreover, the time period for which we are measuring fairness is approximately 1 second.

We can see that the AMC protocol provides a high degree of fairness even with guardbands of uptil 8%. We see that here the Jain's Fairness Index is above 0.95. The reason for why this happens is given in Section 3.2.2. Also, we can see that the 802.11-like DCF gives a Jain's Fairness index of 0.79, which is smaller than what AMC provides but still significantly higher than the pipelining protocol.

Moroever, we can see that the pipelining protocol performs very poorly in terms of fairness amongst nodes. It gives us a Jain's Fairness Index of 0.16.

This means that 84% of the nodes in the network suffer from unfairness. We find that there are certain aspects of the pipelining protocol that naturally results in unfairness amongst nodes. For example, with the pipelining protocol a node has to win *two* contention resolution stages, in order to be able to finally transmit on the data channel. When a node advances to stage 2 of the contention resolution from stage 1, and contends for the data channel but another station wins, the losing node doubles its $CW1$ and returns back to stage 1 for contention. Moreover, if a node in stage 2 of the contention resolution suffers from a collision on the data channel, then the node doubles its $CW2$ and contends again on the data channel. All the nodes will return back from stage 2 to stage 1, only if a successful transmission starts on the data channel. Only the winner in stage2 can reset its $CW1$ and $CW2$ to the minimum values. Moreover, the winning station upon finishing a successful transmission immediately advances to stage2 and starts to contend on the data channel again rather than contending on stage 1.

It is clear here, that the winning node always has a higher chance of transmitting on the data channel. Moreover, with tuned-pipelining what is happening is that the $CW1_{min}$ is set to very large values in comparision to $CW2_{min}$. Moreover, the packet time is very small (1 time slot in this example). Now, when a node successfully finishes its transmission, there is a huge chance that no node in stage 1 has counted its $bc1$ to 0, and has become a pipelined station. After the transmission finishes, all the nodes from stage 1 advance to stage 2,

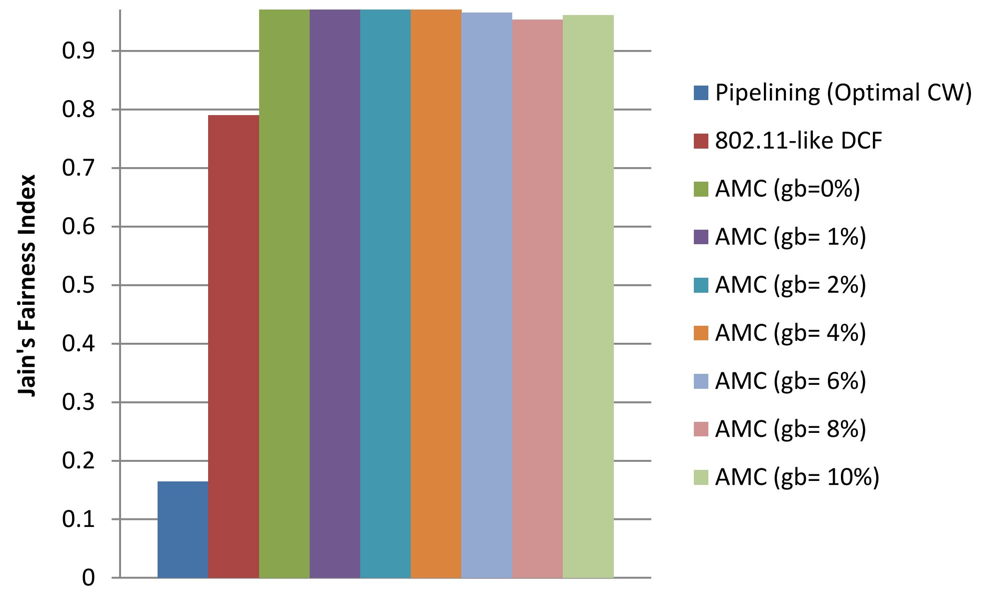


Figure 3.12: Jain's Fairness index for tuned-pipelining protocol and the AMC protocol with different guardband widths. 64 nodes is assumed.

set $bc2$ to the same value as the remaining $bc1$ and continue counting down their back off counter. In the mean time, the winning station just chooses a small $bc2$ and can transmit again. This will cause all the remaining nodes to *further* double their $CW1$ and then again contend on the control channel. When the successful transmission finishes again, we will again have all nodes advance from stage 1 to stage 2 but with even larger backoff counters than before, and the winning station has infact a higher chance than the previous round to win the data channel again. This process continues. Other nodes can transmit, however, with a much smaller chance than the winning station. Therefore, the tuned-pipelining protocol suffers from severe fairness problems.

Moreover, similar unfairness issues are also observed with the tuned pipelining protocols and different number of nodes.

Thus, in this section we can conclude that the pipelining protocol is not a suitable protocol for high data rate wireless networks, even though it is developed for the purpose of reducing bandwidth independent overheads. If we tune the pipelining protocol, we can achieve higher throughput than the Optimal 802.11-like DCF, and than the AMC protocol with guardband widths of greater than 1%, however, at the cost of drastically reducing fairness amongst nodes. On the other hand, the AMC protocol can significantly outperform the other protocols, in terms of both throughput and fairness. AMC provides a higher throughput with guardbands of uptil 1%, than the tuned-pipelining protocol while maintaing a very high degree of fairness amongst nodes. Moreover, with guardbands of even uptil 10% the AMC protocol still performs better than the 802.11-like DCF, while also maintaing a very high degree of fairness amongst nodes. Hence, here we again showed that a channelization scheme such as the AMC (Adaptive Multichannel) protocol is a suitable scheme for improving performance in high data rate wireless networks.

3.5 Related Works

To the best of our knowledge, no prior work investigated the performance of the pipelining protocol and Extended Reservation protocol in High Data

Rate WLAN settings, and compared their performance against an adaptive channelization scheme. Our work addresses this gap.

In [114], [81], [113], [97] and [80], some of the problems of high data rate wireless networks in the 60 GHz band are discussed, and in [114], [81], [113] potential solutions are also presented. In [80] mainly the global spectrum regulations for the UWB and 60 GHz band are discussed. In [114], [81] and [113] the main problems discussed are the issues of robustness and range in the 60 GHz band, which are related to the PHY layer. They also propose techniques for increasing transmission range and robustness by using a multi-band system, that is capable of operating on both the 2.4/5 GHz and on the 60 GHz band. In [81] the problem of bandwidth-independent overheads dominating in high data rate wireless networks with CSMA/CA MAC is mentioned. However, no potential solutions are proposed or evaluated.

Note that, while the above bodies of work deal with high data rate wireless networks, they are orthogonal to our work, as we are interested in understanding and solving problems in high data rate WLANs, that mainly operate in the 2.4/5GHz band and not the 60 GHz band. While several Gbps can be supported at the PHY layer in the 60 GHz band, the physical layer characteristics here are very different than 2.4/5 GHz bands, requiring different protocols to be developed and evaluated for the two bands.

In contrast to the above work, there is a body of literature for the 2.4/5GHz bands that has identified the problem of reduced channel utilization when the network is shifted from low data rate to high data rate [38,55,65,66,74,103,109–111,122]. These works have identified that the reduction in channel utilization is coming due to bandwidth independent overheads. However, in all these works there is only one basic idea that is used in one way or the other, in order to enhance the MAC throughput at high data rates. The basic idea is to (1) allow the node to contend for the single-channel in the same way as 802.11 DCF, however, after winning the channel allow the sender to send *multiple* packets, and (2) use *block ACKs* inorder to reduce the ACK overhead. While all of the works in [44,65,66,103,109–111,122] evaluate the *network throughput* that can be achieved with such protocols, none of the works, evaluate *fairness* amongst nodes that have the same priority traffic, and none of the works compare the performance against a channelization approach.

We developed our own Extended-Reservation protocol, that is similar to the above work, however, with an even further enhancement, of releasing the channel as soon as an ACK packet does not arrive, instead of the sender contineously transmitting during its reservation period. We developed our own analytical models for the saturated throughput for this scheme and run simulations, and unlike the past work, we *compare* the Extended-Reservation protocol against an Adaptive Channelization Approach (AMC), in order to see the performance gain interms of both fairness and throughput that the AMC protocol is capable of achieving.

Now, there are also some other approaches that are geared towards increasing MAC efficency wireless networks. The authors in [56] and [51], propose the MAC protocol *Idle Sense*, as an alternative to the 802.11 DCF, in order to improve both throughput and short-term fairness in wireless networks. Here, all nodes sense the average number of idle time slots between any two transmissions on the channel. The hosts then compare this observed value with a theoretically derived value for the optimal average idle time between transmissions. Based, upon this comparision the nodes then use an AIMD (Additive Increase and Multiplicative Decrease) technique to dynamically converge their contention window to similar values with which eventually optimal number of idle time slots will be observed between transmissions. A large number of idle time slots between transmissions indicate that too much time is spent in nodes waiting to transmit, hence, the nodes additvely increase their transmission probability. This is acheived by decreasing the CW additively. Moreover, a small number of idle time slots between transmissions indicate that the network is suffering from collisions, and hence, all the nodes reduce their transmission probability multiplicatively. This is acheived by increasing the CW in a multiplicative way.

While Idle Sense provides a better throughput than the standard 802.11 DCF, Idle Sense cannot provide very high throughput in high data rate networks when compared with AMC. Infact, we can see in table 4, taken from [56], that the channel utilization for "IdleSense" drops as we shift to faster and faster physical layers. In fact, with a physical layer bit rate of only 100 Mbps and

10 nodes the channel utlization drops to approximately 45%. From here, we can estimate that with 1 Gbps the channel utilization for IdleSense will drop much lower than 45%. The reason for this poor performance in high data rate networks is because the bandwidth-independent overhead (idle slots between transmissions) that is introduced by Idle Sense, is still substantial.

Also, the authors in [84], propose a new architecture for 802.11 networks named "FARA" (Frequency Aware Rate Adaptation). FARA operates on wide unchannelized frequency bands and improves the aggregate network throughput via a new physical and MAC layer technique. FARA's physical layer technology enhances per-link throughput in the network by taking under consideration the frequency diversity in wide channels. Here, the transmitter uses the OFDM physical layer technology, in order to send data. Each sender uses the entire frequency band, and FARA increases the physical layer throughput, by allowing different appropraite modulation-scheme/bit-rate for different subcarriers of the spectrum. This is different, than the previous works where all the subcarriers were given the same modulation-scheme/bit rates, without taking under consideration the fact that now, for wide bands certain subcarriers might not be suitable for certain modulations due to the SNR observed at the receiver.

Moreover, FARA also attempts to increase the throughput at the MAC layer by allowing the sender to send to multiple destinations simultaneously once it gains access to the channel. After gaining access to the channel, FARA's MAC protocol distributes the subcarriers between the packets intended to different destinations. Here, FARA attempts to determine the best set of subcarriers to be used for each of the destinations, with which the throughput across the receivers will be maximixed. Moreover, it allows all the packet transmissions to happen simultaneously and take approximately the same amount of time, in order to avoid wastage due to unused portions of the spectrum.

However, FARA's MAC protocol also faces certain issues in high data rate wireless networks. FARA gains channel access by using the Idle Sense technique and it is important to observe here that FARA's MAC layer protocol will not provide any throughput gain, if the sender has packets just for one destination. In infrasturcture WLANs this is always the case for uplinks that the clients have only one destination - the access point. Here, when multiple clients are contending to gain access to the channel, the FARA's MAC layer protocol behaves exactly like Idle Sense. Thus, in FARA also, the problem of reduced channel utilization due to bandwidth-independent overheads remain.

In Section 3.2 the AMC protocol channelizes the wide spectrum dynamically based on demand. There is a rich body of already existing multichannel MAC protocols that are developed for the purposes of achieving higher throughput in different types of wireless networks [27,28,59,70,85,93,98,108]. However, our work is different than these works because in all the above works the number of channels, width of channels and location of channels on the wide spectrum are already predefined/static. The above works just propose techniques that attempt to use these available predefined channels efficiently. However, the above work is not geared towards conducting a study of understanding problems in high data rate networks, and they do not provide any comparisons against the Extended-Reservation protocol or the pipelining protocol. Additionally, unlike this body of work, the AMC protocol makes use of the technique of dynamically defining channel widths and channel numbers based upon the load in the network, in order to provide very high throughput and fairness amongst nodes.

There also exist other schemes that dynamically distribute the spectrum amongst APs or nodes in the network instead of using static channels [77,121]. However, in [77], in the case of a single cell (one AP and clients associated to AP), again there is no channelization and we get the same performance as IEEE 802.11-DCF. Also, in [121] both centralized and distributed algorithms are proposed for allocating dynamic time-spectrum blocks to nodes in cognitive radio networks. However, this work does not study the problem from an angle of high data rate wireless network, and does not provide any study of comparing performance with the Extended-Reservation protocol or the Pipelining protocol.

Moreover, there are several works that deal with allocating the available spectrum dynamically in cellular networks [100, 101]. In [100], for example, the spectrum is dynamically allocated to base stations however, with the goal of optimizing the total revenue, and not throughput or fairness. Hence, these works are orthogonal to the goal of this chapter, which is to understand the performance benefits that one can achieve in high data rate wireless networks, with the Extended-Reservation protocol, pipelining protocol and an Adapative Channelization MAC scheme, and how these protocols compare with respect to each other.

3.6 Conclusion

In this chapter we have rigorously investigated the performance of different plausible MAC protocols in high data rate wireless networks, in a single-collision domain setting. We have described the problem that single-channel 802.11-like DCF faces, as the physical layer data rate increases. Here, the 802.11-like DCF performs poorly interms of channel utilization, because the channel wastage due to bandwidth-independent overheads introduced by the MAC layer becomes substantial. To alleviate this problem, we investigated the performance of alternative MAC layer protocols, namely, the Extended-Reservation, Pipelining, and Adaptive Multichannel (AMC) protocols in a high speed regieme. We developed analytical models and simulators for the different schemes. We showed that while appealing, both the Extended-Reservation Protocol, as well as, the Pipelining protocol, (even when optimized for performance) are not suitable for high data rate wireless networks. There exists a tradeoff between throughput and fairness with both these schemes. On the other hand, the AMC protocol designed for single collision domains can attain significantly higher network throughput and fairness amongst nodes at the same time. We also analyzed and described the core reasons for *why* the AMC protocol attains these gains. Based on the results we attained in this chapter, we promote a channelization approach that can be adapted based on traffic, for both efficiency and fairness in high data rate WLANs.

Chapter 4

Adaptive Spectrum Distribution in Future WLANs with Multiple Collision Domains

4.1 Introduction

In the previous chapter we conducted an extensive analytical and simulation study, in a single collision domain setting, to show that a MAC approach that adaptively channelizes the wide spectrum based on traffic, can achieve far better performance in emerging WLANs, than several other MAC protocols. In this chapter, we extend our study of channelization to Infrastructured WLANs with multiple APs, clients and multiple-collision domains[1]. Unlike single collision domain settings, now, adaptively channelizing the wide spectrum, to achieve high spectrum utilization and fairness, becomes a challenging and hard problem. We tackle this problem of how to adaptively distribute spectrum in such networks to improve performance, from a theoretic and an algorithmic perspective. We design efficient algorithms and show via analysis and simulations the throughput and fairness gains achieved with our approaches. We show that our techniques can significantly outperform single-channel, fixed-channelization and other related schemes.

It should be noted that developing strategies for enhancing spectrum utilization and fairness in wireless LANs has long been of interest in the research community. It is well known that the 802.11 standard *statically* channelizes the wide spectrum into a fixed number of channels, of fixed and equal widths [9]. Each AP and its associated clients can use one of these channels for communication. Depending upon the channels assigned to interfering APs[2] and the distribution of clients across APs, many issues can arise, such as inefficient spectrum utilization, low network throughput, low per-user-throughput and unfairness amongst clients of different APs. In order to alleviate these problems, many techniques have already been proposed, that use the predetermined channels efficiently [105], [75], [76], [72], [79], [31], [104].

[1] In a multiple-collision domain network, all nodes do not necessarily hear each other.

[2] The terms *interfering APs*, *links*, *interfering links* and *single-channel 802.11 DCF* have been defined in Section 4.2.

In contrast to these works, a recent work [77], proposes to *dynamically change* the channel *widths* and *central frequencies* of APs, with the varying traffic demands at each AP, in order to achieve high spectrum utilization and fairness amongst clients. Here, an AP with many clients associated to it, is provided a wider portion of the spectrum than neighboring APs with lower load. This approach of dynamic spectrum distribution, based upon the traffic in the network, has several benefits over just using the fixed 802.11 channels efficiently [77]. For example, with fixed channelization, an AP which is a hotspot will not be given a channel that is wider than 20 MHz (or 40 MHz with 802.11n), even if the neighboring APs have no load. Also, by appropriately adapting channel-widths at APs, fairness amongst clients in the network is naturally achieved without causing the clients to move from heavily congested APs to APs with lower load.

However, the technique described in [77] provides *non-overlapping* channels to interfering APs. It can very well be that some links[2] that are related to one of the interfering APs might not interfer[2] with some links of the other interfering AP. For example, in figure 4.2a AP_1 and AP_2 are interfering, but, the links L_7 and L_8 of AP_2 are not interfering with any of AP_1's links. However, with the technique in [77] such *non-interfering* links belonging to *interfering* APs are not allowed to use the same portions of the spectrum, and hence spectrum reuse opportunities are missed. In this paper, we propose a *new* dynamic spectrum distribution technique, that exploits such spectrum reuse opportunities by allowing overlapping channels to interfering APs. Our goal is to find channel-widths and channel-locations for individual *links* in the network in such a way that (1) high level of max-min fairness is achieved amongst all clients in the network, and (2) high spectrum utilization and thus high network throughput is achieved. We will refer to the problem of finding the "proper" channel-width and channel-location (i.e., central frequency) for each link as the ***Channel-Configurations Problem*** or ***(CC-Problem)***.

It is also worthy to mention here, that our dynamic spectrum distribution technique can also be used as a solution to reduce the penalty incurred due to bandwidth-independent overheads, in high data rate networks [69], [66]. In high speed networks [69] the single-channel 802.11 DCF[2] gives poor channel utilization, because the wastage due to bandwidth-independent overheads,(e.g., backoff periods), becomes substantial at such high speeds where now the packet transmission time is shorter. With our technique, based upon the current traffic, the wide spectrum is *channelized* into multiple smaller channels on which the senders do not face any backoff periods. We show in Section 4.5 that our technique performs better than the case where all the nodes contend on the wide spectrum using single-channel 802.11-like DCF.

4.2 Definitions and Terminologies

- A *communication link* or simply a *link* refers to communications that takes place between an AP and one of its associated clients.

- The *Pairwise Secondary Interference Model* is used in this work, under which two links either interfer or they do not interfer. Moreover, *two links, l_1 and l_2 interfer*, if an endpoint of l_1 is *within* a one-hop distance from an endpoint of l_2. For example, in figure 4.2a, L_1 interfers with L_6, since AP_1 and the client for L_6 are within a one-hop distance from each other. Note that this model is used to capture interference in 802.11 networks.

- For the *Network Graph*, NG, all the clients and the APs form the vertices in NG , and all the links in the network form the edges in NG. (e.g., figure 4.2a).

- A *Conflict Graph*, CG, represents the interference in the network. Each link in NG is represented as a vertex in CG, and there exists an edge between any two vertices in CG if these two vertices (i.e. links in NG) interfer with each other.

- Two APs, AP_1 and AP_2 are said to be interfering if there exists two links l_1 and l_2 of AP_1 and AP_2, respectively, that interfer with each other.

- A rate allocation to links is said to be max-min fair iff the rate of any link cannot be increased without decreasing the rate of a link with an already smaller or equal rate.

- *Single-channel* in the phrase "single-channel 802.11-like DCF" , refers to the whole wide available spectrum.

- A *collision-free* channel or *conflict-free* channel, c_1, for a link, l_1 means that c_1 does not overlap with the channel related to a link that interfers with l_1.

4.3 Network Architecture

Our technique is designed to operate in enterprise infrastructured wireless LANs, where all APs are connected via a high speed backbone network [4] to a central server, and to each other. All APs can collect their current *local neighborhood* view and current *traffic load* information. The local neighborhood view of an AP is defined as the clients that are associated with the AP, and the one-hop neighbors of the AP and its associated clients. Moreover, in order to avoid distracting details, in this work, the number of clients associated with each AP is refered to as the traffic load at that AP.

Every now and then, all the APs forward this collected information to the central server, and the central server will use the technique proposed in this paper, to determine the current channel width, and current channel location for each link in the network. The central server, then reports back this information to the APs, and the APs inform their associated clients of the channel-width and channel-location on which they should be operating on. Each client/AP will then use ordinary 802.11-like DCF, (with the minimum contention window provided by the central server), to contend on the channel that is provided to it.

It is important to note, that here there are two separate issues involved: (1) After receiving information, how should the central server distribute the spectrum amongst links and (2) What type of MAC layer protocol should be used in the WLANs, that collects local neighborhood and traffic information, and that notifies clients of the changes that need to be done in their channel configurations. In this paper, we are focusing on providing a solution to the first case, and we show the potential performance benfits that can be achieved with our technique.

We also assume OFDM Physical layer technology which is already used in the WiFi standard [12]. OFDM splits the wide spectrum into many narrow *orthogonal* subcarriers, and a group of these subcarriers, which we call *a channel*, is allocated to each link for data transmission. The presence of several thousand subcarriers [99], can not only provide a rich variety of possible channel widths, but it can also support separate channels for large number of links, easily. Note that, the guardbands between subchannels can be negligible with real radio implementations [37].

The APs should be able to receive on multiple channels simultaneously.[3] Moreover, the APs should also be capable of transmitting on multiple channels simultaneously.[4] However, note that we do not require such capabilities on the clients.

The APs should also be able to receive and transmit simultaneously, in order to support both uplinks and downlinks at the same time. Also, the APs and the clients should be able to switch their channel-widths and central frequencies on the fly.

4.4 Solving the Channel-Configurations Problem

We can solve the CC-problem in the following two steps:

1. Find the *channel-width* for each link.
2. Find the *channel-location*, i.e., central frequency, for each link on the wide spectrum.

[3] In fact, some progress in this area is seen, in [39], where, USRP2 and GnuRadio technology is used to simultaneously receive on five 802.15.4 channels. Additionally, the WiFi-NC radio also have similar characteristics [37].

[4] Progress in this aspect is seen in [84], [115], [37].

4.4.1 Step 1 - Determining Channel-Widths for Links

After receiving the local neighborhood view from the APs, the network graph, NG, is built, and the corresponding global conflict graph, CG is built. Once we have attained the conflict graph, we can now *assign* max-min fair shares of the *spectrum* to all the links in NG (i.e., vertices in CG).

We say that a spectrum-share assignment for links is max-min fair iff the following two properties hold [58]:

1. The assignment is feasible. This means that for any clique Cl in CG the total sum of the bandwidths given to all of the vertices in Cl is less than or equal to the total bandwidth of the whole spectrum.

2. For any vertex l_1 in CG, the given bandwidth b_1 cannot be increased while maintaing feasibility, without decreasing the bandwidth b_2 of another link l_2 , where $b_2 \leq b_1$.

The authors in [58] have already provided a technique for finding the max-min fair shares of a resource for vertices in a conflict graph. We directly use their technique for our work. Here, firstly, all the maximal cliques in the conflict graph are found, and each of these cliques are initially given a capacity of 1. Afterwards, the vertices of that clique that gets the smallest per-vertex-share are assigned their shares. This clique is deleted and all the remaining cliques are

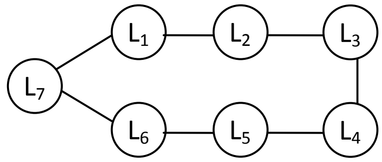


Figure 4.1: A conflict graph where max-min fair share of each link is $B/2$. Here, we cannot get collision-free channels of width $B/2$ for all links. B is the total spectrum bandwidth.

updated in terms of number of vertices, and remaining capacity. This process continues until all vertices are assigned their shares.

Note that the problem of finding all maximal cliques is known to be NP-hard. However, as mentioned in [58], instead of using the *global* conflict graph altogether, we can use the *local* portion of the conflict graph for each conflict-graph-vertex, and leverage the same technique mentioned above to find the fair share of that vertex. Since now we have smaller graphs to work on, the max-min fair spectrum-shares for the vertices in CG can be found in a reasonable amount of time.

The figure 4.2b represents a conflict graph with the assigned max-min fair shares, for the network graph 4.2a. ***We will consider the max-min fair spectrum-share for each link as the channel-width for that link.***

4.4.2 Step 2 - Determining Channel-Locations for Links

Now that we know the channel-width for each link, if we can divide the spectrum in such a way that each link is given a channel of the assigned width and all the channels are positioned in such a way in the wide spectrum, that every link is given a *conflict-free* channel, then we will experience very high network throughput, and at the same time all senders will achieve max-min fair throughput. We achieve high network throughput for the following reasons:

- Each link will get a *collision-free* channel, and hence will not face any contention and backoff periods, (assuming optimal contention window).
- By the very definition of max-min fair allocation, the lowest given share to a link is *maximized*, then the second lowest is *maximized*, and this continues until the largest share given to a link is *maximized* [5]. Thus, naturally, each link receives the *largest* possible channel-width, (without violating fairness), which results in high spectrum utilization and thus high network throughput.

However, note here, that even though for many network topologies, finding conflict-free channels for all links is possible, there still exists some network scenarios for which we can get the max-min-fair-share assignment for links, however, we cannot achieve conflict-free channels for *all* links, with the given channel-widths [58]. We will call such network topologies, Case A topologies. An example of such a scenario is shown in figure 4.1. Here, we will not have any other choice but to give at least two interfering links the same channel-location on the wide spectrum, if we maintain channel-widths determined by the max-min fair shares of the links[5].

Nevertheless, for a very wide range of network topologies, which we call Case B topologies, we can still locate conflict-free channels for all links, with the respective max-min fair widths. Figure 4.2 shows an example of such a case. However, we have proven via a reduction from the vertex-coloring problem [19], that the problem of *finding channel-locations for links in such a way that no two interfering links are given overlapping channels*, is an NP-hard problem. We will call this problem *the channel-location problem*.

4.4.2.1 Formulating the Channel-location Problem as a CSP (Constraint Satisfaction Problem)

For Case B topologies, we formulate the channel-location problem as a CSP problem, and we use the *backtracking search algorithm* described in [90] to find the exact solution to this problem. Moreover, we also apply optimizations such as Forward Checking and Constraint Propagation [90], in order to reduce the

[5]For finding the channel-locations for Case A topologies, we can use our approximation-algorithm in Section 4.4.2.2, because here, all links will still be provided a *conflict-free* channel on the given spectrum, by *reducing* the given channel-widths for links to make room in the spectrum for all links.

search space. The following are the three main components that define the CSP:

- *Set of variables* - In our case, this is the set of links.
- *Set of possible values for each variable* - In our case this is the set of all possible $(s_i, s_i + b_i)$, for each link i. Here, b_i is the given channel-width for link i. Moreover, s_i represents a *possible* starting frequency for i's channel. $s_i \geq F_s$ and $(s_i + b_i) \leq F_e$, where F_s and F_e are the starting and ending frequencies of the wide spectrum.
- *Constraints Between Variables* - In our case, the conflict graph is used to represent constraints between variables. Two interfering links in the conflict graph should not be assigned overlapping channels.

For Case B topologies, by using the backtracking technique, we will eventually find a solution to the *channel-location* problem, and all the links will be given a collision-free channel of width that is the same as the max-min fair share for the respective link. Figure 4.2c, shows the actual channels that are allocated to the links of the network shown in figure 4.2a. Note that AP_1 and AP_2 are given overlapping channels.

4.4.2.2 Approximation Algorithm for Solving the Channel-location problem

Since solving the above CSP takes exponential time and thus is only suitable for small networks, we present an efficient *8-approximate algorithm* for finding a conflict-free channel-location for each link. The algorithm greedily *attempts* to provide each link a conflict-free channel of width that is given in section 4.4.1. Let B be the total bandwidth of the given spectrum, and F_s and F_e be the same as in section 4.4.2.1.The following is the description of our algorithm:

1. Sort the list of all links, L', in decreasing order of given channel-widths.
2. Pick the first link, l_2, from the list, and allocate it the lowest possible portion on the spectrum (starting from F_s), that does not overlap with c_1, where c_1 is an already assigned channel for any link l_1 that interfers with l_2. Allocate such a channel even if the channel goes beyond F_e. Remove l_2 from L'.
3. Repeat (2) until L' is empty.
4. If $B' = B$ then the algorithm returns and we will have *optimal* network throughput[6]. Optimal spectrum utlization is achieved because the algorithm was successful in providing each link a conflict-free channel of

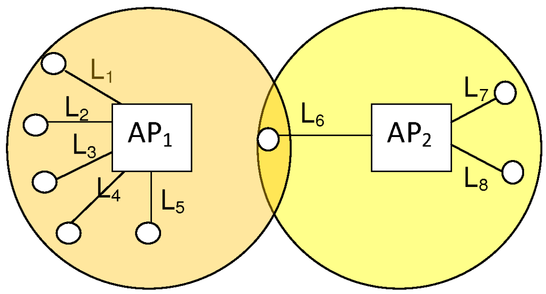
[6]Let F'_e be the end-frequency of the channel with the highest given end-frequency. Then $B' = F'_e - F_s$.

width that is determined by the max-min fair spectrum-share for the link.

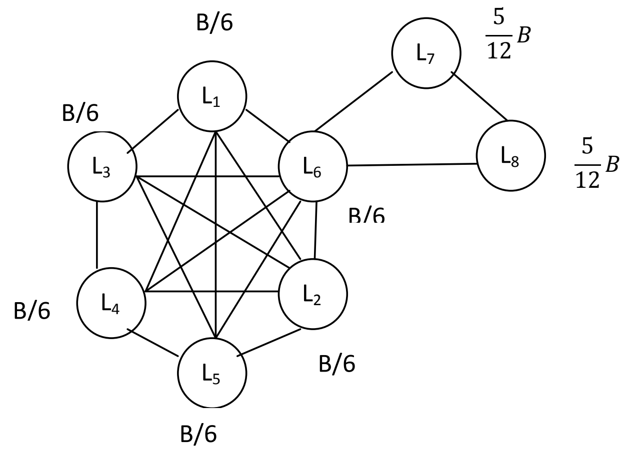
5. If $B' > B$, then we *squeeze* all the allocated channels into the given spectrum. Hence, for each link l, with channel central-frequency cf and channel-width w, adjust the central frequency to $F_s + \frac{B \cdot (cf - F_s)}{B'}$ and *reduce* the *given width* to $\frac{B \cdot w}{B'}$.

Now, all links will be provided a *conflict-free* channel on the given spectrum and the senders will not face any backoff periods. If the algorithm *reduces* the given *channel-widths* to fit everyone in the spectrum, then we have proven that the algorithm gives network throughput that is *at least* 1/8 times the optimal, (assuming unit disk graph model). By *optimal network throughput*, we mean the total throughput that will be achieved if we had the same channel-widths that are computed in Section 4.4.1.

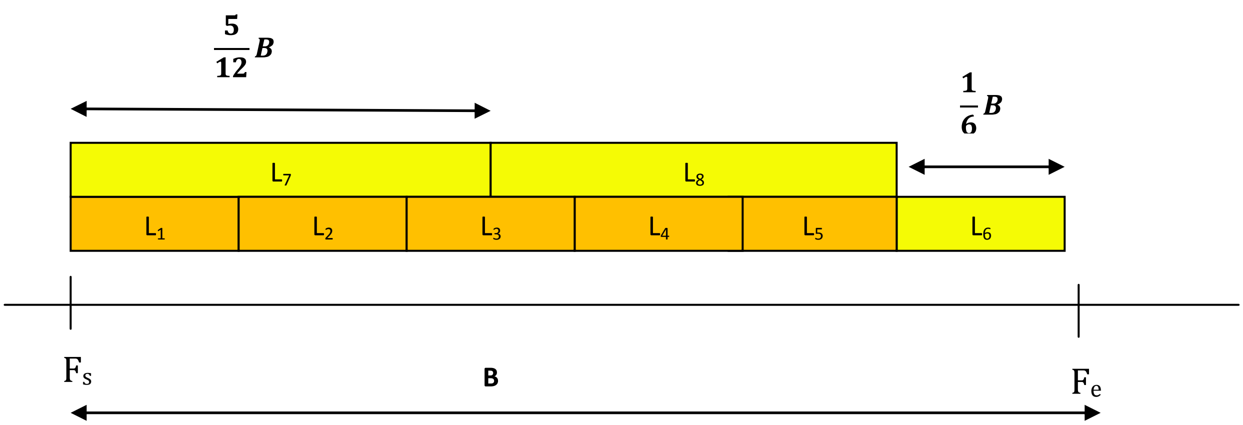
We show in Section 4.5, via simulations, that our algorithm still manages to provide high network throughput and a high level of max-min fairness amongst clients in the network. By providing channels to links starting from the lowest possible portion of the spectrum, we reduce wastage of spectrum due to fragmentations, and hence, we increase the chances of reducing the given channel-widths by a *lesser amount* in order to provide everyone a conflict-free channel, if we could not provide channels of the *given max-min fair widths*. Moreover, for the same reason we start providing channel-locations for the wider chan-



(a) Infrastructure WLAN with two interfering APs. L_6 and all links associated to AP_1 interfere.



(b) Conflict graph with max-min fair shares.



(c) Channels given to links. Orange portions represent AP_1's channels and yellow portions represent AP_2's channels.

Figure 4.2: Solving the CC-Problem in an Infrastructured WLAN. We get conflict-free channels for all the links. B is the total bandwidth of the wide spectrum.

nels first. Note that our algorithm guarantees no starvation in the network, elimination of hidden and exposed terminal problems, and elimination of contention overheads e.g., backoff periods, for the current traffic in the network. Moreover, since with our algorithm, we *do not* have two channels of different widths, belonging to interfering links, overlapping with each other, and contention happening on these channels, therefore, we naturally *do not* face the rate anomoly problem [56].

4.5 Performance Evaluation

In this Section we evaluate the network performance achieved with our dynamic spectrum distribution technique, single-channel 802.11-like DCF, classic fixed channelization techniques and the state-of-the-art dynamic spectrum distribution technique in [77], which we will refer to as T_1. We show that even when conditions are made favorable for other techniques, our technique still outperforms them, in terms of network throughput and max-min fairness amongst clients. We have implemented simulators for all the above cases. We assume saturated conditions(i.e., the senders for all links always have a packet to send), slotted time, and 160MHz wide spectrum with packet transmission time of 1 time slot in the single-wide-channel setting. Also, here we are presenting our results for the case where we have only downlinks. Similar gains were observed with our scheme for the case of when both uplinks and downlinks were active.

In order to allow for a fair comparision, we assume that all other techniques, use our radio described in Section 4.3. The authors in [77] provide variable-width but *non-overlapping* portions of the spectrum to interfering APs, and they say that for communication between an AP and its associated clients standard 802.11 DCF can be used. However, with our radios their network throughput is further enhanced, because the spectrum allocated to each AP can be further divided into smaller channels of equal width, one for each associated link. In this way, no contention overhead and no hidden terminal problems will be experienced in their work. T_1 gives each AP_i a channel-width of $\frac{L_i}{L_i+\sum_{j\in N(i)} L_j}B$, where, L_i is the number of clients associated to AP_i and $N(i)$ is the set of APs interfering with AP_i. We further assume ideal conditions for their work, where all the APs are always granted channels of the *described width*.

Similarly, for the fixed channelization technique, we further enhance network performance by assigning the fixed-width channels to APs in the best possible way and by using our radios in the same way as we did for the work in [77].

For the single-channel 802.11-like DCF, we just tune our radios on the APs and the clients to operate on the entire wide spectrum, rather than on one of the fixed 802.11 channels, and we allow all the senders to contend using basic 802.11-like DCF.

We evaluate all the techniques in terms of both network throughput and level of max-min fairness achieved amongst clients. In order to evaluate the level of max-min fairness for clients 1 to n in the network, we use the following metric:

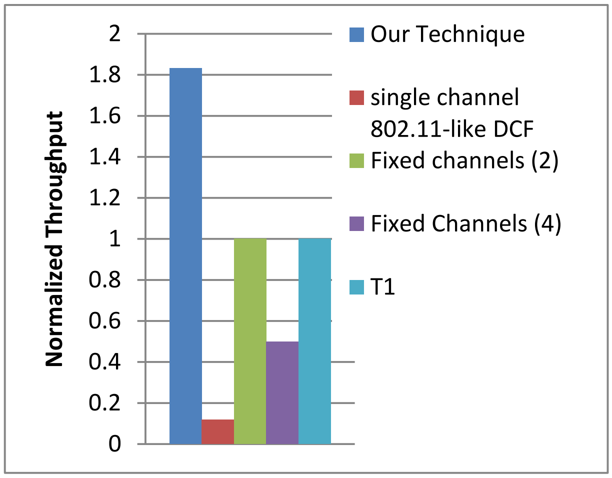
$$I = average(\left\{ \frac{|share_i - t_i|}{share_i} : i \in \{1, ..., n\} \right\}) \tag{4.1}$$

where, t_i stands for the actual achieved throughput for *client* i, and $share_i$ stands for the max-min fair share of throughput that i should have received. Thus, I, will represent the amount of deviation from the clients' respective max-min fair shares, on an average. Hence, if $I = 0$, then this means that all the clients are achieving their max-min fair shares. The higher the value of I, the lower the level of of max-min fairness experienced in the network[7].

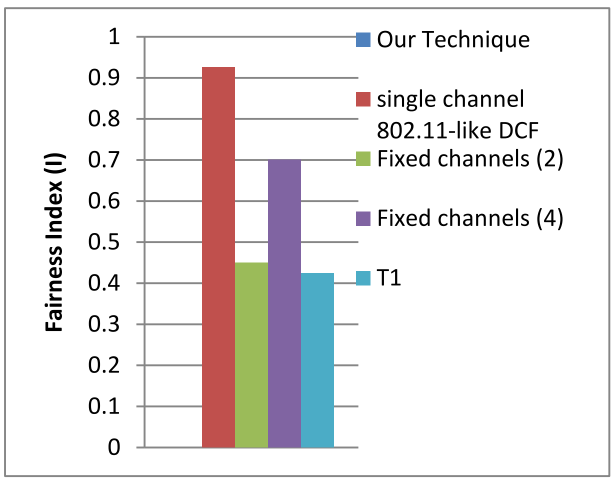
4.5.1 Part 1 - Evaluating Performance in Sample Network

Figure 4.3 shows the normalized throughput, and the fairness Index I attained with the 5 different techniques for the network setting in figure 4.2a. For the fixed-channels case, the best assignment is to give both AP_1 and AP_2 two different channels. It is clear here, that in terms of both normalized throughput

[7]Note, that we are not using the the Jain's Fairness Index [60] over all the clients' achieved throughputs, because, it is a measure of *uniformity* amongst these throughputs. It cannot measure the level of *max-min* fairness in the network. The same argument holds for the concept of relative Jain's Fairness Index in [74], which measures the *uniformity* amongst ratios of max-min fair shares, achieved by the respective links.



(a) Network Throughput is normalized to the total capacity of the wide spectrum.

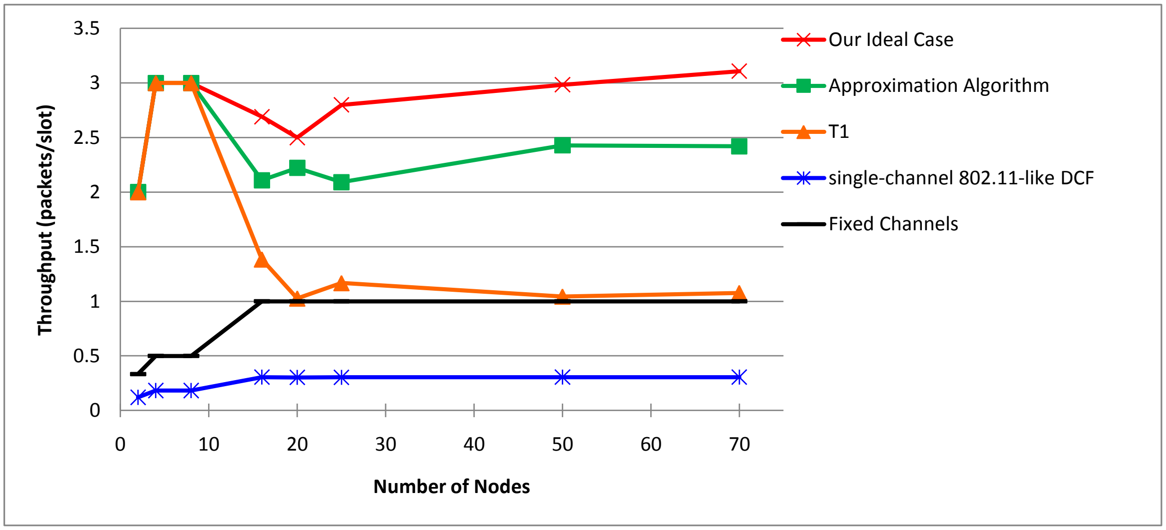


(b) Max-min fairness Index for different techniques. 0 shows that all the clients are receiving their complete max-min fair share.

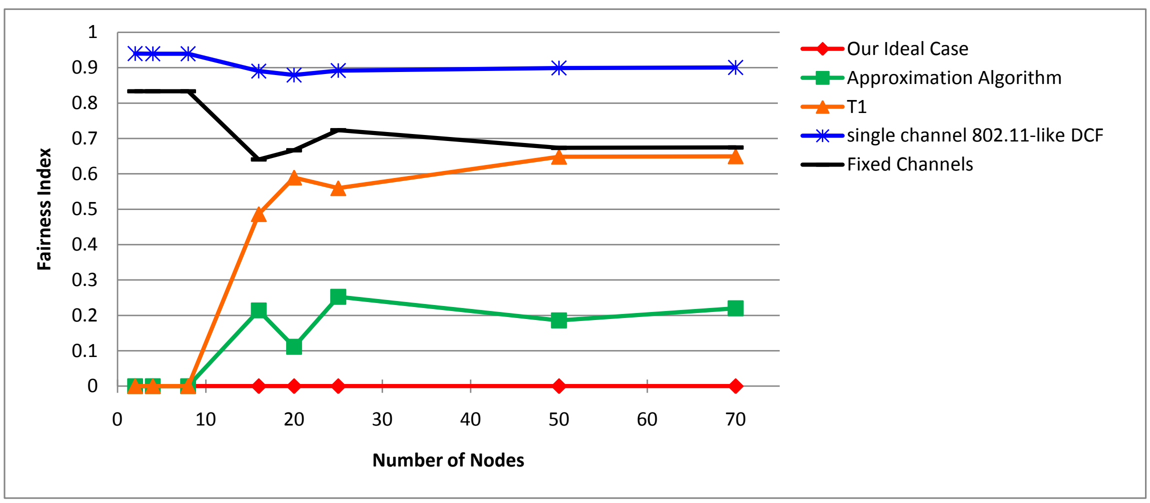
Figure 4.3: Normalized throughput and max-min fairness measure achieved with different techniques. *Fixed Channels (y)*, means that the wide spectrum is statically divided into y channels of fixed and equal widths.

and I, our technique ***significantly*** outperforms all the other techniques, *including* T_1. In figure 4.3a we can see that T_1 gives a normalized throughput of 1. However, with our technique we achieve a normalized throughput of **1.83**. Our technique outperforms T_1 because it allows for a *better usage of the available spectrum.* With T_1, the wide bandwidth B is split into two non-overlapping channels of width $\frac{5}{8}B$ and $\frac{3}{8}B$, and these are given to AP_1 and AP_2 respectively. In contrast to this, our technique exploits spectrum reuse opportunities and allows the two APs to have *wider* and overlapping channels as seen in figure 4.2c. For similar reasons, our technique performs better than the fixed 2-channels case. Moreover, the channel utilization for the fixed 4-channels case, falls even lower then that of T_1 and the fixed 2-channels case, because with the 4-channels case not only a similar problem to that of T_1 is faced, but also half the spectrum remains completely unused. We can also see that the single-channel 802.11-like DCF, provides the worst normalized throughput, because of bandwidth-independent overheads ,e.g., backoff periods [69]. Note, that no other technique is suffering from contention overheads.

In figure 4.3b, with our technique all the client are receiving their max-min fair shares of throughput. In other words, here, all the clients are receiving the *highest possible* throughput without violating fairness amongst clients. With other techniques and especially 802.11-like DCF, the clients experience lower level of max-min fairness in the network.



(a) Total network throughput vs. Number of nodes.



(b) Max-min fairness Index vs. Number of nodes.

Figure 4.4: Total Network throughput and max-min fairness measure achieved with different techniques in random networks.

4.5.2 Part 2 - Evaluating Performance in Random Networks

We have also evaluated the above mentioned techniques in random networks. Here, we show the results for 6 APs *randomly* placed in a 100m x 100m space. We assume a transmission range and interference range of 25m. Some APs lie within the transmission range of each other while some don't. We have randomly placed 2 to 70 clients in the network and for each client we randomly chose an AP in its vicinity for association. In figure 4.4a, we have varied the number of nodes along the x-axis and we have plotted the network throughput achieved with the different techniques. Here, *Our Ideal Case*, stands for the case when all the links are given *conflict-free* channels of widths that is determined by the respective max-min fair shares of the links. Note that our approximation algorithm provides network throughput close to our ideal case, and both our ideal case and our algorithm provides a drastic improvement in network throughput over all other techniques, even with increasing number of clients. Note that since all our 6 APs can be potential interferers to each other, therefore, we take an optimistic approach towards the fixed channels case by statically channelizing the wide spectrum into 6 channels of equal width, where each AP receives a different channel.

Moreover, in figure 4.4b, we have varied the number of nodes along the x-axis and we have plotted our fairness index I for all the different techniques.

Again, it is clear here, that with our ideal case, all the clients will receive their max-min fair shares of throughput. Moreover, our algorithm still provides a better level of max-min fairness in the network, than the other techniques.

4.6 Related Works

FLEX [115] allows APs to dynamically access a portion of the spectrum, based upon traffic demand, in a distributed fashion with the goal of achieving overall proportional fairness amongst clients. Here also, we face the same problem of non-interfering links of interfering APs not being allowed to access the same portions of the spectrum. Moreover, the goal of our work is different since we want to achieve max-min fairness amongst clients in the network.

In FARA [84] dynamic channelization takes place when the AP, after contending on the wide spectrum and winning access, has packets to send to multiple distinct destinations. Here, the AP divides the wide channel into multiple channels one for each packet. However, FARA faces multiple problems limiting its usefulness for high data rate WLANs. Firstly, no channelization takes place when clients have packets to send to the AP and when an AP has packets to send to only one destination. Hence, here, the wastage due to bandwidth-independent overheads can still remain substantial, similar to single-channel 802.11-like DCF, thus, limiting the network throughput achieved with FARA in high data rate WLANs. Additionally, FARA relies on Idle Sense, which is not even defined for multiple collision domain settings.

The authors in [77], discuss a scheme for adaptive channelization in WLANs based on demand, however, their approach provides interferring APs with non-overlapping channels, which misses spectral reuse opportunities. In contrast our dynamic spectrum distribution technique provides overlapping channels to interferring APs, thus, expoliting spectral reuse opportunities. Also, our techniques and study of adaptive channelization is different than the work in [77].

Additionally, more related literature has also been discussed within the chapter.

4.7 Conclusion

Recent work has shown that adapting channel-widths and channel-central frequencies for APs, based upon the traffic load at the APs, provides better network performance than using the fixed 802.11 channels [77]. In this paper, we have presented a new technique for dynamically distributing spectrum amongst APs, based upon the current traffic in the network. Here interfering APs are allowed to operate on wider and overlapping channels, in order to achieve high network throughput and a high level of max-min fairness amongst clients in the network. We have shown via simulation results that our dynamic spectrum distribution technique has the potential to significantly supersede various spectrum distribution techniques. With our technique achieving a network throughput improvement of a factor of 2 or higher over different techniques is possible.

Chapter 5

A First Look at Performance in Emerging Mobile Virtual Network Operators

5.1 Introduction

In the previous chapters we studied performance in Next-Generation High Speed WLANs. Like WLANs, cellular networks have also experienced a variety of changes over the past years, and in this chapter [1] we focus on a recent growing phenomenon in Cellular Networks. One of the changes in the last few years is that a new trend has been emerging in the cellular market both in the US

[1] The work reported in this chapter originally appeared in the Proceedings of the ACM SIGCOMM Internet Measurement Conference 2014 [124].

and in Europe—the rise of mobile virtual network operators or *MVNOs* [23, 25, 26]. At a high-level, MVNOs use the existing cellular infrastructures that are owned by the traditional cellular operators. MVNOs do not incur significant infrastructure or spectrum licensing costs and offer services that are different from traditional cellular operators (e.g., better pre-paid plans and multiple quotas).

While MVNOs started to appear in the market in the early 2000s, they have only recently become more mainstream in terms of market share. The growth is a culmination of several factors: increasing prices of traditional cellular providers for consumers, users' preference in avoiding contractual lock-ins, the popularity of "pre-paid" services, regulatory intervention to ensure competition and market segmentation focusing on niche demographics (e.g., tween markets) [47]. As of Q1 2014, there are 20, 23, 11 and 35 MVNOs running on top of the AT&T, T-Mobile, Verizon and Sprint networks in the US, respectively [13].

Even as MVNOs grow in market share, there are concerns among users about their performance. For example, a quick sampling of popular consumer complaints forums shows significant concerns related to both cost, billing, and service issues (e.g., poor coverage/signal, 3G/4G promised but getting 2G, poor application performance, and frequent disconnections). Shown below are some actual quotes from user forums about MVNOs:

[22]: I know that AIO is capped at 8mbps download speed. Are all the other MVNOs like Straight Talk, Net10 and AirVoice also limited to 8mbps download speeds? Do they suffer from higher latency?

[20]: I have been throttled every day since last week so each day I lose my 4g/E symbol and once I regain it Im throttled ... I've used 1.4gb and I have only 3 days left on my 30 plan.

[24]: Does Sprint have means of degrading service to Ting (and other MVNO) customers in favor of Sprint customers in a particular crowded cell?

[21]: My only concern is if the service quality of the service. With Straight Talk, for example, I'd be on AT&T's GSM network in Boston, I think ... but I wonder if as an MVNO customer I'd get second-tier access or service.

Motivated by the growth of MVNOs and the aforementioned user concerns, this paper presents a first study to shed light on the performance of different MVNOs. While there is a lot of previous work in analyzing mobile performance (e.g., [7, 42, 43, 57]), they have not systematically analyzed performance in MVNOs. To address this gap, we study two major MVNO families in the US. In our study, each family includes the base carrier and three popular MVNOs running on top of the base carrier. While this sample study does not cover all base carriers in US or all MVNOs atop any base carrier, the carrier/MVNO choices have been done systematically, based on popularity (Section 5.2). In the performance analysis, we hide the actual names of the carriers and MVNOs to protect their business interests. To simplify presentation, we refer to the two base carriers as carrier `A` and `B`. We refer to the MVNOs within the carrier `A` as `A1`, `A2` and `A3`, and within the carrier `B` as `B1`, `B2` and `B3`. The base carrier along with its MVNOs (e.g., `A`, `A1`, `A2`, `A3`) are refered to as 'MVNO family' or just 'family.'(e.g., MVNO family `A`)

As a starting point, we analyze the performance for three dominant usage modes: web access, video streaming and voice calls. Using over 13,000 measurements collected across 11 locations over a period of 3 months, we address the following questions:

- Does the performance vary across the MVNOs running atop the same base carrier? (e.g., is MetroPCS worse than Straight Talk given that they are both MVNOs running on T-Mobile network?)
- Do MVNOs perform worse compared to the base carrier in each case? (e.g., is H20 Wireless, an MVNO on AT&T network, worse than AT&T)?
- Are there differences across different MVNO families? (e.g., do all MVNOs in a family, say the T-Mobile family, show significantly worse performance than those in another family, say the AT&T family?)

We analyze application-specific quality-of-experience (QoE) metrics to address these questions. We also perform factor analysis to correlate the observed application-level performance with network-level performance, such as, TCP throughput, round-trip times (RTTs), packet loss rates, DNS look up times, and PHY-layer characteristics to attribute the observed performance differences (if any) to structural differences across the operators.

Our key findings are:

- The base carrier often performs better than the MVNOs and sometimes significantly so. For instance, some MVNOs over base carrier `B` fail to load a non-trivial ($\geq$10%) fraction of YouTube video requests and can have up to 6$\times$ worse page load time.
- There is significant diversity across MVNOs within the same MVNO family, for both the `A` and `B` MVNO families. For instance, often `B2` performs considerably worse than `B1` and `B3` in MVNO family `B`.
- There are non-trivial differences between the two MVNO families; overall the MVNOs running atop `A` have better performance w.r.t the base carrier compared to their `B` counterparts.
- Finally, we see key differences across applications as well. While voice quality is largely similar across all MVNOs and base carriers, there is huge discrepancy in data performance both for web access as well as video streaming.

We hope that this chapter serves as a motivation for future large-scale measurement studies in this direction, that would span wider areas, larger number of MVNOs and wider variety of data plans.

5.2 Measurement Setup

In this section, we begin by describing the choice of phone, carriers, and cellular plans. Then, we describe our data collection methodology.

Choice of phone: To ensure we do not have phone-specific effects (e.g. CPU speed/memory access latency/cache size) in our measurements, we use the same phone model for all carriers – Google's Nexus 4 with 2GB RAM, Quad-core CPU and 2G/3G/4G support (i.e., EDGE/UMTS/HSPA/HSPA+). All of our phones run the Android 4.2.2 (JellyBean) OS. Since Nexus 4 only supports GSM-based carriers, this study is limited to such carriers and their MVNOs only. We leave the investigation of performance in CDMA-based carriers and their MVNOs for future work.

Choice of carriers and plans: We chose popular and widely-used MVNOs that run atop two major base carriers in the U.S. We call them carriers `A` and `B`, respectively. We used Google Trends [8] and the list of all the available MVNOs [13] to find the top 3 MVNOs for each of these base carriers. The 6 MVNOs are summarized in Table 5.1. `A1`, `A2` and `A3` run atop `A`; `B1`, `B2` and `B3` run atop `B`. Cellular providers offer a range of plans with different prices.

Carrier	Type	Plan (all pre-paid except B)	$/Month
A	Base	Unlimited talk/text, 2.5GB data @ 4G	60
A1	MVNO	Unlimited talk/text, 2.5GB data @ 4G	45
A2	MVNO	Unlimited talk/text, 3GB data @ 4G	50
A3	MVNO	Unlimited talk/text, 2.5GB data @ 4G	60
B	Base	Unlimited talk/text, 2GB data @ 4G	65
B1	MVNO	Unlimted talk/text, 2.5GB data @ 4G	50
B2	MVNO	Unlimited talk/text, 2GB data @ 4G	50
B3	MVNO	Unlimited talk/text, 2GB data @ 4G	50

Table 5.1: Mobile carriers and plans used in our study.

Hence, to provide a fair comparison between carriers, we select *similar* plans for all the carriers (as summarized in Table 5.1), in terms of features.When the exact plan was not available we picked the closest comparable plan.

Data collection: We selected 11 reasonably diverse locations spanning different usage scenarios: urban/suburban, shopping areas, residential, office/lab and hospital. Figure 5.1 shows the geographical spread of our measurement locations. We acknowledge a potential limitation, that all our measurements occurred in the Long Island/New York region. However, this region is a major population hotspot, covering part of New York city metro area and associated suburbs.

We developed a suite of custom scripts and mobile applications for web browsing, video streaming, and voice calls, representing common usage modes. Our custom tools collect relevant user quality-of-experience (QoE) metrics, and we defer application-specific details to the following sections.

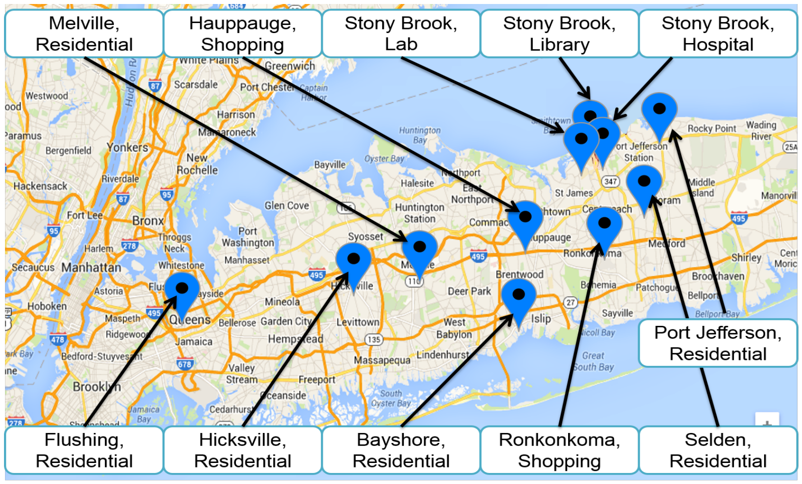


Figure 5.1: We conduct measurements at 11 different locations spanning across a 3000 km^2 wide area. The annotations show the names of the measurement locations along with the type of location.

At each location, we use four identically configured Nexus 4 phones (one for base and three for MVNO carriers) to run the same suite of experiments concurrently at that location. Our scripts run these applications at each location typically hourly or half-hourly for most of the day – often starting at early morning and going until late night – over different days of the week modulo practical constraints (e.g., shop/mall closures).

On average, we conducted about 150 sets of measurements for each carrier, across different locations, during Jan-Mar 2014, on different days. Each measurement set consists of a series of application runs, e.g., web page access for a set of chosen web sites, video streaming, voice calls, TCP upload throughput test, etc.

Concurrent with the QoE measurements, we also log packet traces using the `tcpdump` tool and a range of relevant phone characteristics using the Android API (e.g., radio stats), to enable further factor analysis. We verified separately via the `top` utility, that this additional monitoring adds only a modest CPU overhead ($\approx 5\%$). This does not bias our measurements. Prior to conducting actual measurements, we performed tests over WiFi, where we ran our apps with and without additional logging, and we measured the performance for web, video and voice applications, as well as, network tests. The attained results showed negligible difference in performance with this logging enabled or disabled.

Our analysis did not reveal any significant location, time-of-day or day-of-week specific change in terms of performance of carriers with respect to each other. Thus, we present only aggregate statistics (over all locations, times and days) and focus on performance differences across carriers and MVNO families. Since the experiments for base and MVNO carriers are always colocated in both space and time, we believe it is a fair characterization of the performance issues we describe in the rest of the paper.

5.3 Application Performance

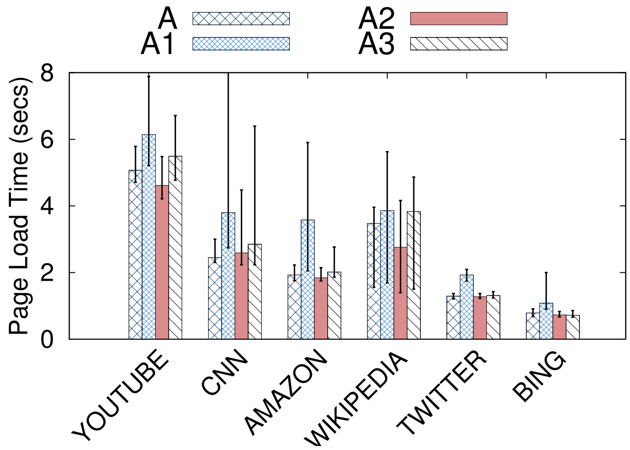
In this section, we analyze the performance of the MVNOs and the base carriers for three common modes of mobile usage: web access, video streaming, and voice calls. In each case, we describe the application-specific setup, and the relevant QoE metrics we measure. We also correlate the observations to key network-level and PHY-layer characteristics.

5.3.1 Web Browsing

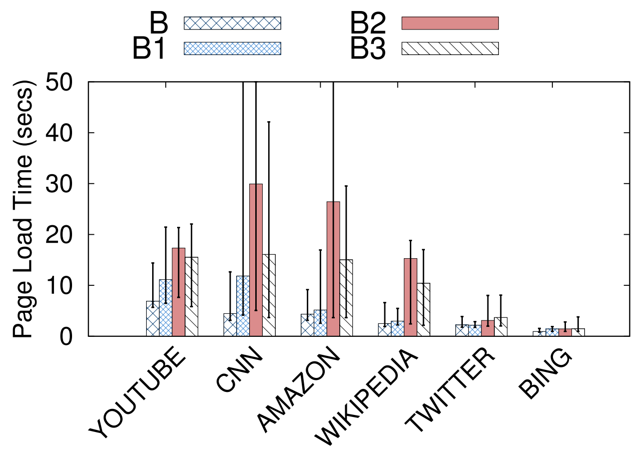
Setup and QoE Metric: We choose six popular websites with diverse characteristics: Youtube, Amazon, Wikipedia, Twitter, Bing, and CNN. All of these sites had an overall Alexa rank ≤ 20 in April 2014. We developed a custom browser application using Android WebViewClient. At each measurement site, the app visits each website's mobile landing page (in random order across carriers) and records the *page load time* QoE metric. We measure page load time as the difference between the time the URL is requested from the browser and the time when all the web objects (html text, images, etc.) are fetched and the `onPageFinished` event [15] is triggered.

Note that the set of webpages accessed is diverse in terms of structure and content size, with CNN and Amazon constituting the two largest median content sizes in the set ($\approx$ 570 KB and 400 KB, respectively) and Twitter and Bing having the smallest ($\approx$ 89 KB and 100 KB, respectively).

Evaluation of Page Load Times: Figure 5.2 shows the distribution of page



(a) MVNO family A



(b) MVNO family B

Figure 5.2: Distribution of page load times (median, 25th and 75th percentiles): We see that (a) MVNO family A usually performs better than MVNO family B; (b) within each MVNO family one or more MVNOs is worse than the base carrier; and (c) some MVNOs (e.g., B2, B3) suffer more than others).

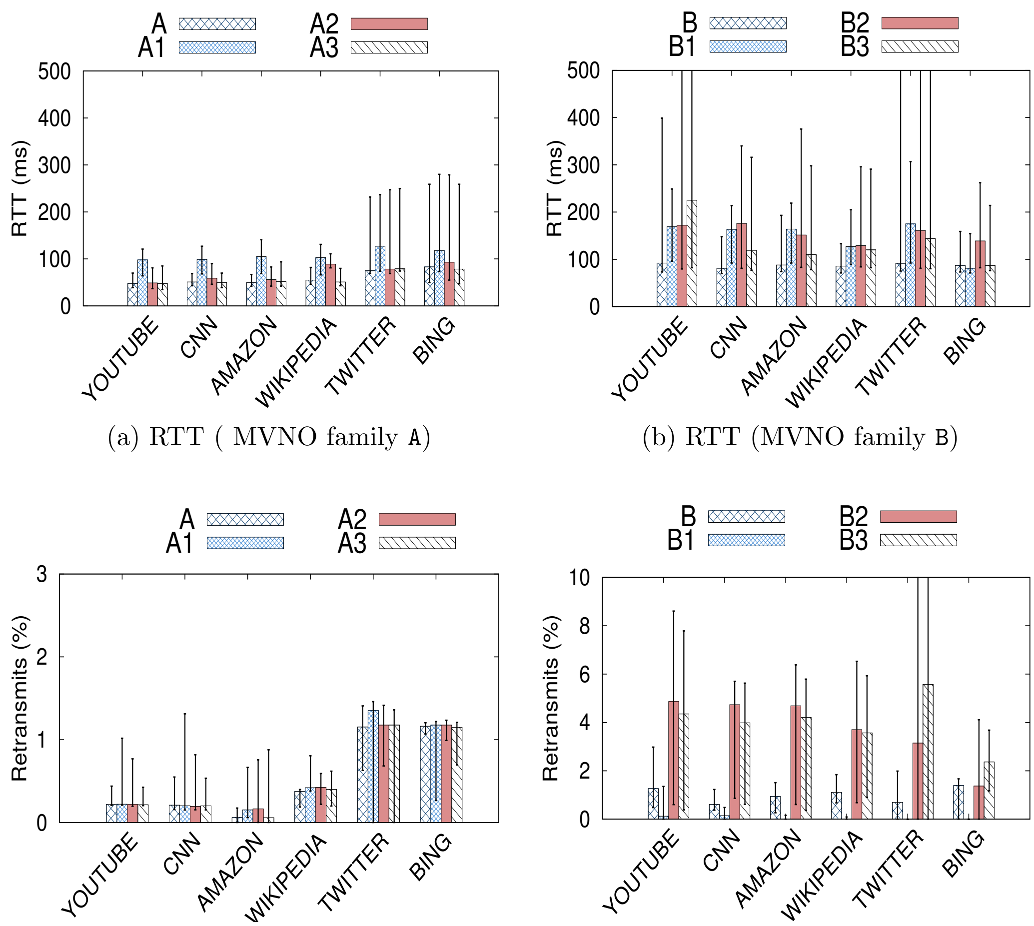
load time across all runs for the six websites.[2] There are three key observations

[2]Note that, >10 sec page load times are not surprising on a mobile platform, as seen

here. First, typically the carriers in MVNO family `A` perform better than their `B` counterparts; e.g., for CNN all carriers in MVNO family `A` perform better than all carriers in MVNO family `B`, and sometimes significantly so. Second, while the differences between base carrier `A` and its MVNOs are only modest, we see significant differences between base carrier `B` and some of its MVNOs; e.g., `B2` is almost 6× worse than `B` for CNN. Finally, we see non-trivial variability across MVNOs within the same MVNO family; e.g., `B2` is often considerably worse than other MVNOs in MVNO family `B`, and `A1` is slightly worse than other MVNOs in MVNO family `A`. We confirmed that these differences between carrier page load times are statistically significant using the Kolmogrov-Smirnov (K-S) [17] statistical test, but do not show these results for brevity.

Factor Analysis: To understand the causes of these performance differences, we looked at lower-layer metrics such as DNS lookup time, RTT, TCP retransmission rates, and signal strength. We computed the Pearson's correlation between the difference of page load times for the base carrier and the MVNOs and that of different lower-layer metrics, for every website. Based on this analysis, we zoom in on two key factors – RTT and TCP retransmissions (Figure 5.3). First, we can see that MVNO family `A` has generally lower RTTs and retransmission rates than MVNO family `B`. As prior studies have shown, lower RTTs imply lower page load times, which is consistent with our obser-

in prior work [41, 96, 107]. For example, Welsh reports 75 seconds page load time for a webpage over a cellular link [107].



(a) RTT (MVNO family `A`)

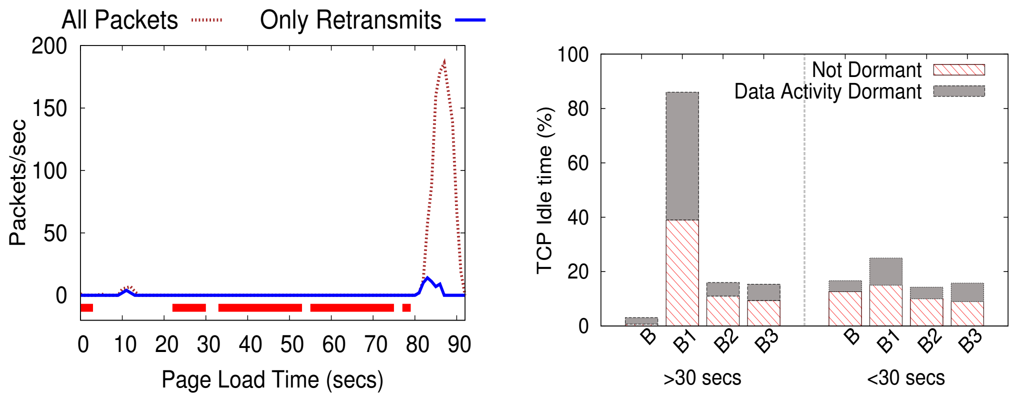
(b) RTT (MVNO family `B`)

(c) Retransmissions (MVNO family `A`)

(d) Retransmissions (MVNO family `B`)

Figure 5.3: Focusing on the key observed factors shows that generally speaking the MVNOs in family `A` with higher page load times have higher RTT and the MVNOs in family `B` with higher page load times tend to have high retransmission rates.

vations that `A` and its MVNOs have lower page load times [32]. Second, we see in Figure 5.3a that within the MVNO family `A`, `A1` which had higher page load



(a) Timeseries of one high page load (b) High dormancy impacts load times for `B1`

Figure 5.4: Higher page load times for `B1` relative to `B` are not due to retransmissions but rather due to high radio dormancy periods. The red line in (a) shows intervals when data activity is dormant.

times, indeed has higher RTTs.[3] Finally, Figure 5.3d shows that the MVNOs in MVNO family`B` (`B2` and `B3`) with the highest page load times see very high retransmission rates.

We also observe that `B1` has the lowest retransmission rates in its MVNO family, however, still higher RTTs than `B`, thus resulting in `B1` having a lower page load time than `B2` and `B3`, but, higher than `B`. However, this still does not explain some of the very high (> 30s) page load times for `B1` (e.g., CNN in Figure 5.2b). Further analysis of the packet traces showed significant TCP idle times as shown in one example timeseries in Figure 5.4a. Figure 5.4b breaks down the page load measurements in the `B` MVNO family in two bins (< 30s

[3]Higher RTTs for both Twitter and Bing, as compared with other webpages, could be due to the content-server locations that were accessed for these sites, or due to the path from carrier `A`'s gateway routers to these servers [89, 123].

and > 30s) and shows that these TCP idle periods have non-trivial influence on the page load times. This is specifically true for B1 where the long page load times have about 80% idle periods.

Further inspection reveals that many (but not all) of these idle periods are actually due to physical link being dormant, (as revealed by the DATA_ACTIVITY_DORMANT flag [2]). We suspect that this is influenced by the RRC state machine at the radio layer as defined in the 3GPP standard [50], but we do not have visibility to actual RRC states using the commodity Nexus 4 phone to examine this further. Prior work (e.g., [41,88]) has also shown that inappropriately tuned RRC states impact web access performance. Overall, this suggests some potential misconfiguration or service differentiation at the radio layer for the MVNO B1 running over carrier B. In contrast, TCP idle/dormancy issues are negligible for MVNO family A and are not shown.

We also analyzed signal strengths, handoffs and the pool of cell-ids that the carriers are associating with and found no significant differences between carriers within the same MVNO family. This implies that these radio-layer aspects did not play a significant role in the performance difference observed between MVNOs. Some prior work (e.g., [48, 73]) also noticed little correlation between signal strengths and performance when analyzing their collected measurements. This is likely due to the signal strengths usually falling above a certain threshold.

Our investigation in this section also revealed interesting information about the structural differences across the `A` and `B` MVNO families. In MVNO family `B`, all web traffic goes through an explicit proxy server that terminates TCP connections while MVNO family `A` appears to use a transparent proxy that relays the connections to the webservers.[4]

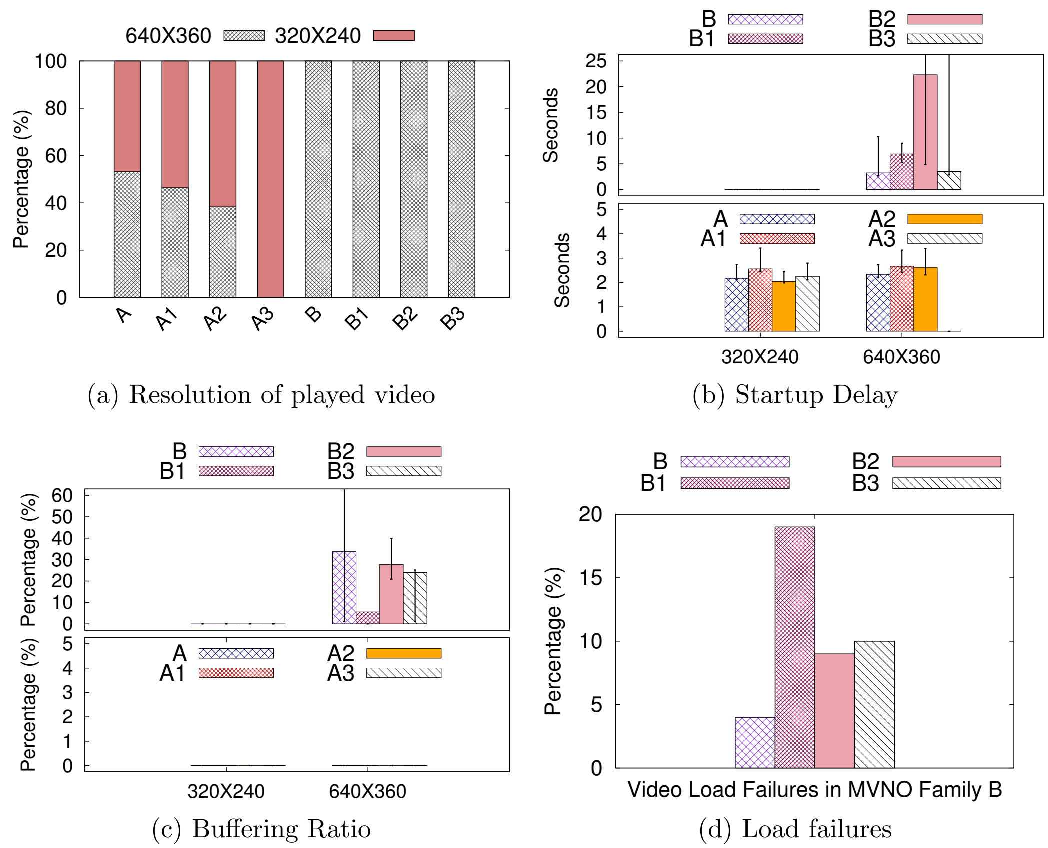
5.3.2 Video Streaming

Setup: We choose a 3-minute YouTube video available in both high/low quality and play it in a custom app. We use the Android YouTube APIs [3] to extract player states (paused, playing, buffering) to compute the QoE metrics described below. Similar to the web experiments, we run measurements for both the base carrier and associated MVNOs, simultaneously, at multiple locations and at different times of the day.

QoE metrics and Evaluation: The key video QoE metrics are: (1) *video resolution* being delivered;[5] (2) *startup delay* or the time between the user clicking on the play button and the time the video starts playing; (3) *buffering ratio* or the percentage of the session duration spent in buffering state; and

[4]We were able to detect the transparent proxy using Netalyzr which showed HTTP header modifications [14].

[5]The YouTube API does not perform dynamic video resolution adaption on mobile. It selects a resolution that it considers suitable for the current connection at the start and uses it for the entire session.



(a) Resolution of played video

(b) Startup Delay

(c) Buffering Ratio

(d) Load failures

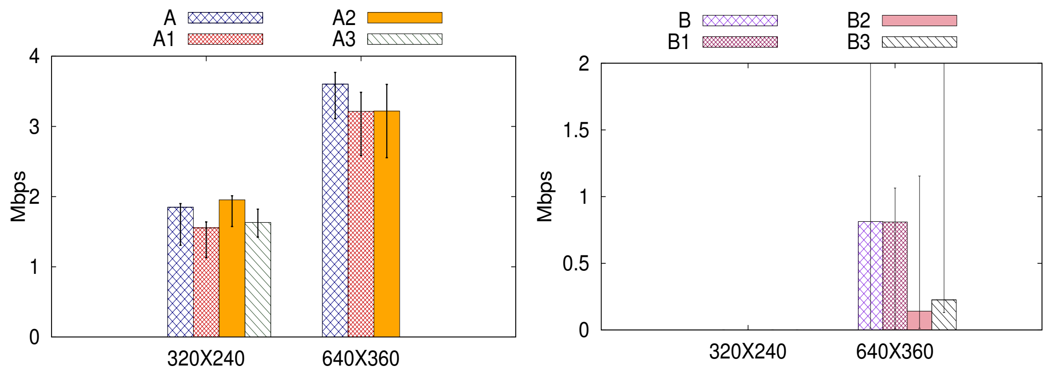
Figure 5.5: Video quality-of-experience metrics for the MVNOs and base carriers. Note that MVNO family `B` always plays the high-quality resolution and suffer significant buffering, startup delay, and video load failures.

(4) *load failures*, where the video fails to load. Figure 5.5 summarizes the distribution of these metrics.

First, we observe that, with respect to resolution quality, carrier `B` and its MVNOs always use the high-resolution version of the video. On the other hand, carrier `A` and its MVNOs play a mix of resolutions, except for `A3` that always plays the lower resolution video. Second, in terms of startup delay, MVNO family `B` overall shows a higher startup delay than MVNO family `A`, for the higher-resolution cases. Also, consistent with the web measurements, we find that in MVNO family `B`, the base carrier outperforms its MVNOs in terms of startup delay, and amongst the MVNOs, `B2` performs the poorest with a median startup delay of 23 seconds. Third, we see that MVNO family `A` outperforms MVNO family `B` in terms of buffering ratio as well, and `B2` again performs the worst amongst the MVNOs in its family. Finally, we find a non-trivial number of video load failures for the MVNOs in the `B` family; e.g., `B1` fails $\approx 20\%$ of the time.

Factor Analysis: As before, we use the correlation coefficients to zoom in on key network-level factors. The startup delay and buffering states are (unsurprisingly) mostly influenced by *TCP throughput*. Figure 5.6 shows the difference in the measured TCP throughput across the carriers and confirms the earlier observations about video QoE. Surprisingly, MVNO family `B` chooses the higher quality video even though it has lower TCP throughputs than MVNO family `A` (and hence incurs more buffering). We suspect that this is related to the explicit proxy described earlier; i.e., the bitrate negotiation at the beginning of the session is done by the proxy and does not account for the actual "last hop" throughput achievable by the client.

To further understand the load failures, we analyzed the packet-level traces and find two reasons behind these failures: (1) the proxy *blocks* the video



(a) MVNO family `A` (b) MVNO family `B`

Figure 5.6: TCP throughput influences video quality

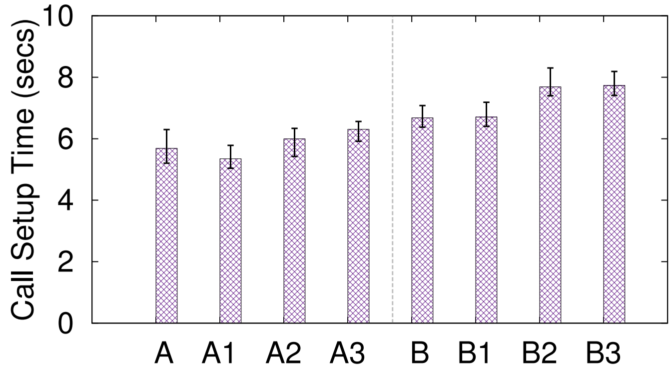
requested by the client by sending an HTTP response with the status code 403 ('access forbidden') and (2) the proxy does not respond to the initial request from the client causing the client to timeout. Specifically, `B1` experiences the largest number of type (2) video load failures; this is related to radio dormancy issues discussed in Section 5.3.1.

5.3.3 Voice calls

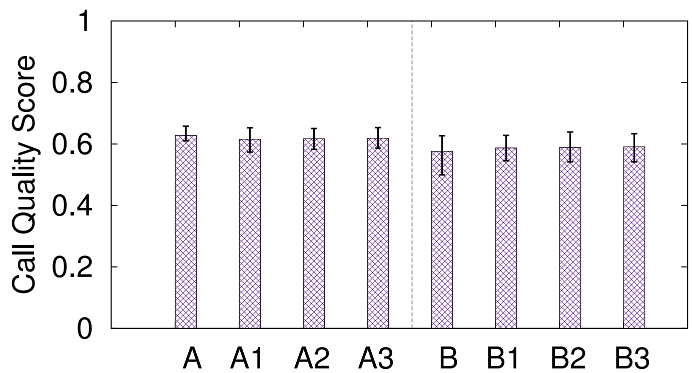
Setup and QoE metrics: We created a custom *auto dial* application that is scripted to automatically call a number. This application runs on all 4 phones at different locations and times of the day. We also setup a recipient phone in the lab, and we build and run another custom application on this phone to log the time of the first ring, accept the call, and then immediately end the call. Because the Android top-level framework does not allow us to automatically answer/end a call, we mimicked a Bluetooth Headset request to automatically answer the call and used lower-layer APIs for ending the call. With this setup we compute the *Call Setup Time* or the time elapsed between the time the caller makes a call and when the callee receives it. To ensure that the caller/receiver are in sync, we use the `ClockSync` [29] Android app. We separately verified that the synchronization error was $\leq$ 10 ms (not shown); this suffices for our analysis as we look for user-perceivable (e.g., $\geq$ 100ms) differences in performance.

To measure the *audio quality*, we establish calls between each of the 4 phones and a Google Voice number on a laptop. We play a 3 minute (based on average audio call durations) audio file on the laptop and record the incoming audio to the phone.To minimize background noise we direct the audio output from the phone to the recorder via a standard 3.5mm cable. We compute the cross-correlation of the Mel-frequency Cepstral Coefficients (MFCCs), (recommended in the audio/speech processing literature [67]), between the reference audio file and the recorded audio file. We normalize this value by dividing it by the score attained when cross-correlating the MFCCs of the original file with itself, and we call this normalized value the *Call Quality Score*.

Evaluation: Figure 5.7a shows the distribution of the call setup time for the different carriers. MVNO family `A` showed fairly similar values for call setup times (median of 5-6 seconds). With MVNO family `B`, we notice that `B2` and



(a) Setup time



(b) Audio quality

Figure 5.7: Call quality in terms of setup time and the audio quality. While there is no significant difference in the call quality, we do observe that some of the MVNOs in MVNO family `B` have a higher call setup time.

`B3` have a 1.5 second higher median call setup time. However, this difference cannot be attributed to any client-side metric we collected. Figure 5.7b shows the Call Quality Score for the two MVNO families. In this case, we do not observe significant differences across the providers.[6] Since the discrepancy in quality is low, unlike the data experiments, we do not perform any further factor analysis.

5.4 Other Applications

In addition to the three usage scenarios discussed in the previous section, we also conduct smaller-scale measurements to capture other common user concerns. We briefly summarize the main observations from these experiements.

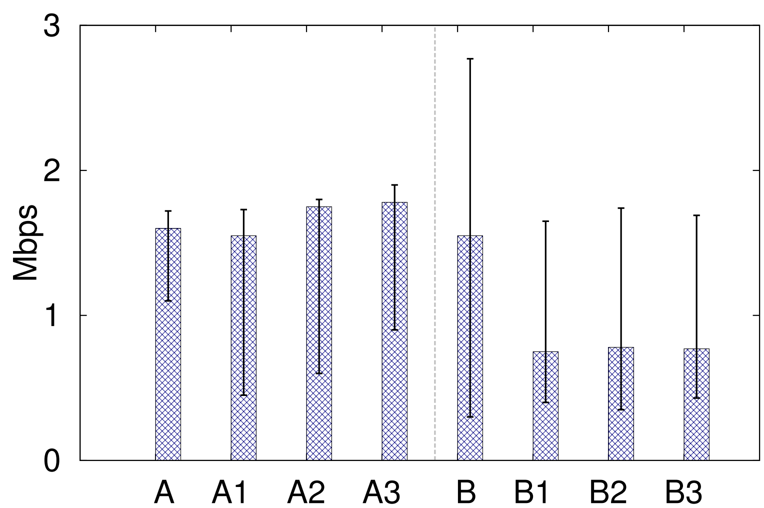


Figure 5.8: TCP Uplink Throughputs.

[6]Note that Nexus 4 phones do not support VoLTE (Voice over LTE). Hence, audio voice calls are sent over a channel separate from the data channel, and thus voice is not impacted by differences on the data channel.

Uploads: Web and video workloads are largely download-bound. Users are increasingly using phones to upload content (e.g., Instagram, Vine). To understand the impact on such applications, we measure the upload speeds obtained by different carriers to a reference campus server in Stony Brook. Figure 5.8 shows similar characteristics to our previous experiments— MVNOs in MVNO family `B` perform significantly worse than base carrier `B`, whereas base carrier `A` and its MVNOs perform roughly similarly.

.

Video Chat (Google Hangout): We pick a popular video chat application – Google Hangout and evaluate its performance across carriers. We establish a chat (5 minutes long) from the phone to a well-provisioned laptop and play a video in front of the phones and the laptop using another screen. We collect packet traces at both ends. We repeat our experiments at 3 different locations, lab and two residential areas. We analyze frames/sec received at the laptop as well as the sending and receiving UDP throughputs. We did not observe any significant difference between the base carrier and their MVNOs for both `A` and `B` MVNO families. One interesting observation is that video chat shows no performance differences on MVNOs in MVNO family `B`, which is unlike the case of video streaming (Section 5.3.2). We speculate that this is due to a combination of two reasons: (1) chat traffic uses UDP and does not go through the explicit proxy and (2) unlike the YouTube video which chooses a static bitrate at the start, Hangout uses dynamic bitrate adaptation. (Recall that most of the problems in MVNO family `B` was the poor choice of initial bitrate via the proxy.)

Traffic Shaping and Port Blocking: Users like to know if MVNOs throttle or block less common applications. This is particularly relevant as we have seen use of proxies (Section 5.3.1) and use of middleboxes in cellular networks [106] is well-known. We use two tools, Bonafide [30] and Netalyzr [14], as they provide complementary coverage over the set of application tests. We used this to study 3 different types of applications: (1) BitTorrent, (2) VoIP-H323 and (3) RTSP-based apps. We found no evidence of application-specific traffic shaping for these protocols, in both MVNO families. Additionally, Netalyzr reveals that while MVNO family `A` does not exhibit any port blocking, MVNO family `B` exhibits more diverse blocking behavior. For example, `B` blocks TCP-based SIP and UDP access to NetBIOS-NS servers. While `B3` and `B2` do not block any ports, `B1` blocks many application ports (FTP, PPTP, NTP, NetBIOS-NS, NetBIOS-DGM, IKE Key Exchange).

Coverage: As seen in our user quotes, users want to know if MVNOs get the same coverage/treatment as the base carriers (e.g., [21]). As discussed earlier, we logged relevant lower-layer information—serving cell-id, signal strength (RSSI/RSCP/RSRP), link layer technology used (e.g., EDGE, HSPA, HSPA+). In addition, we did a number of driving experiments covering major routes within the map in Figure 5.1. We found that in general, the carriers in each MVNO family connect to a similar set of cell-ids in a given location and that there was no statistically significant difference in signal strength or link-layer technology used.

Quota usage: Another common concern for users is whether carriers start throttling before the actual usage quotas are reached (e.g., [20]). We correlated the performance for different applications vs. the data usage amount for every billing cycle. We did not observe throttling behavior for either MVNO family. A detailed study of this subject via more controlled experiments is an interesting direction of future work, especially in light of known accounting discrepancies (e.g., [83]).

5.5 Related Work

With the growth of mobile traffic, there are several prior and ongoing efforts in mobile measurement. While the tools and techniques they use are similar to our work, the key difference is that these have not focused on the MVNO phenomenon to characterize differences across MVNOs or MVNOs vs. base carriers.

Mobile measurements: Previous studies have measured mobile performance from the infrastructure-side [63, 71, 82, 112] and the client-side [42, 43]. These focus primarily on characterizing traffic usage patterns, which is orthogonal to our work. Wang et al. showed how middlebox effects (e.g., timing out idle TCP connections) can have a huge impact on the mobile application performance [106]. Huang et al. compare different carriers on a range of applications across different smartphone hardware [57]. However, their study did not cover MVNOs. More recent studies analyze performance variability within carriers [78] and diagnose causes of high latency in cellular networks [123]. These are interesting factors to further dissect MVNO performance.

Tools and datasets: Several crowd-sourced solutions gather mobile measurements; e.g., FCC's broadband measurement tool [7], OpenSignal (`www.opensignal.com`), Mobiperf (`www.mobiperf.com`), OOKLA Speed Test (`www.speedtest.net`). These focus mostly on network-level metrics (e.g., latency, throughput, signal strength) and do not measure user-perceived QoE metrics which is our primary focus. Bashko et al. developed the Bonafide tool to detect traffic shaping and service differentiation [30]. Netalyzr is also a powerful tool for detecting port blocking, proxies, and DNS issues [14]. We leverage these two tools and apply them to study MVNOs.

5.6 Conclusions

In this chapter, we presented a first study to shed light on a recent and growing trend in the mobile market: mobile virtual network operators or MVNOs. While these have been growing in market share, there are natural concerns about their performance and there has been little work done on systematically understanding this area. To fill this gap, we conducted a systematic measurement study with two major MVNO families in the US. Our analysis shows that while the MVNOs share the network infrastructure of the base carriers, there is visible performance degradation in quality of experience metrics for common mobile phone applications for some MVNOs. Further, MVNOs in the same MVNO family do not perform equally, and the two MVNO families behave differently. Deeper analysis reveals a range of structural and lower-layer differences across MVNO families and MVNOs, including use of proxy, varying latencies and loss rates, data activity dormancy issues and various forms of blocking/denials. We hope that our observations motivate and trigger future deeper and large-scale studies, across larger regions, more MVNOs and more variety of data plans, perhaps by using mobile measurement platforms being deployed in the wild.

Chapter 6

Conclusion

In this dissertation we have extensively studied performance and fairness issues that can arise with next-generation WLANs. We developed and proposed solutions to address these problems. We also extensively studied performance in emerging MVNOs, and achieved insights that are beneficial to not only end-users, but also to MVNOs, underlying carriers and application developers.

In the first part of this dissertation we have focused on future WLANs, and we extensively studied the FICA MAC protocol, which is one of the leading schemes designed for the purpose of improving efficieny in high data rate WLANs. We have identified, for the first time, the problems that can easily arise with the FICA MAC protocol, when packets of different sizes are present in the network. We have quantified the impact of these problems on the performance of FICA via extensive simulations. We showed that these problems can severely degrade channel utilization and fairness in the network, if left unaddressed.

Based on our insights, we designed a new MAC protocol, btFICA, for improving channel utilization in high data rate WLANs, that is based upon the FICA framework. btFICA effectively and comprehensively addresses all the three problems that arise with the FICA MAC protocol, while maintaining the positive features of the original FICA scheme. We show, via extensive simulations that btFICA can improve channel utilization in WLANs by upto 40 times when compared to the original FICA scheme.

In addition to the FICA and btFICA work, we also studied the performance of other plausible MAC protocols in high data rate wireless networks, in a single collision domain setting. We have described the problem that single-channel 802.11-like DCF faces, as the physical layer data rate increases. Here, the 802.11-like DCF performs poorly interms of channel utilization, because the channel wastage due to bandwidth-independent overheads introduced by the MAC layer becomes substantial. To alleviate this problem, we investigated the performance of alternative MAC layer protocols, namely, the Extended Reservation, Pipelining, and Adaptive Multichannel (AMC) protocols in a high speed regieme. We developed analytical models and simulators for the differ ent schemes. We showed that while appealing, both the Extended-Reservation Protocol, as well as, the Pipelining protocol, (even when optimized for per formance) are not suitable for high data rate wireless networks. There exists a tradeoff between throughput and fairness with both these schemes. On the other hand, the AMC protocol designed for single collision domains can attain significantly higher network throughput and fairness amongst nodes at the same time. We also analyzed and described the core reasons for why the AMC protocol attains these gains. Based on the results we attained in this chapter, we promote a channelization approach that can be adapted based on traffic, for both efficiency and fairness in high data rate WLANs.

In addition to the above, single-collision domain study, we also conducted a theoretical, algorithmic study of how to adapt channel widths in Infastructured WLANS, to improve performance. We develop our own efficient algorithms that performs better than related approaches. Recent work has shown that adapting channel-widths and channel-central frequencies for APs, based upon the traffic load at the APs, provides better net- work performance than using the fixed 802.11 channels. In this paper, we have presented a new technique for dynamically distributing spectrum amongst APs, based upon the current traffic in the network. Here interfering APs are allowed to operate on wider and overlapping channels, in order to achieve high network throughput and a high level of max-min fairness amongst clients in the network. We have shown via simulation results that our dynamic spec trum distribution technique has the potential to significantly supersede vari ous spectrum distribution techniques. With our technique achieving a network throughput improvement of a factor of 2 or higher over different techniques is possible.

In the second part of this dissertation, we studied for the first time, the performance in a recent and growing trend in the mobile market: mobile virtual network operators or MVNOs. While MVNOs have been growing in market share, there are natural concerns about their performance and there has been little work done on systematically understanding this area. To fill this gap, we conducted a systematic measurement study with two major MVNO families in the US. Our analysis shows that while the MVNOs share the network infrastructure of the base carriers, there is visible performance degradation in quality of experience metrics for common mobile phone applications for some MVNOs. Further, MVNOs in the same MVNO family do not perform equally, and the two MVNO families behave differently. Deeper analysis reveals a range of structural and lower-layer dif- ferences across MVNO families and MVNOs, including use of proxy, varying atencies and loss rates, data activity dormancy issues and various forms of blocking/denials. We hope that our observations motivate and trigger future deeper and large-scale studies, across larger regions, more MVNOs and more variety of data plans, perhaps by using mobile measurement platforms being deployed in the wild.

Bibliography

[1] 802.11ac: Gigabit Wi-Fi Devices will be shipping in 2012. http://www.zdnet.com/blog/networking/80211ac-gigabit-wi-fi-devices-will-be-shipping-in-2012/1867.

[2] Android TelephonyManager Class. http://developer.android.com/reference/android/telephony/TelephonyManager.html#DATA_ACTIVITY_DORMANT.

[3] Android YouTube Player API. https://developers.google.com/youtube/android/player/.

[4] Definition of Backbone Network. http://en.wikipedia.org/wiki/Backbone_network.

[5] Definition of Max-min fairness. http://en.wikipedia.org/wiki/Max-min_fairness.

[6] Dr. Kun Tan, Microsoft Research Asia, Personal Email Communication, May 2012.

[7] FCC - Measuring Broadband America. http://www.fcc.gov/measuring-broadband-america.

[8] Google Trends. http://www.google.com/trends/.

[9] "IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications".

[10] IEEE 802.11 TGac. http://mentor.ieee.org/802.11/dcn/10/11-10-1361-03-00ac-proposed-tgac-draft-amendment.docx.

[11] IEEE 802.11n Specification. http://standards.ieee.org/getieee802/download/802.11n-2009.pdf.

[12] IEEE 802.11n Standard. https://standards.ieee.org/findstds/standard/802.11-2012.html.

[13] List of MVNOs in the U.S. http://en.wikipedia.org/wiki/List_of_United_States_mobile_virtual_network_operators.

[14] Netalyzer, HTTP proxy detection through HTTP header or content modification. http://n1.netalyzr.icsi.berkeley.edu/info_httpproxy_header.html.

[15] onPageFinished method: Android WebViewClient. http://bit.ly/1loeaVP.

[16] Packet Traces Collected in the Sigcomm 2008 Conference. http://www.cs.umd.edu/projects/wifidelity/sigcomm08_traces/.

[17] The K-S Statistical Test. http://en.wikipedia.org/wiki/Kolmogorov-Smirnov_test.

[18] Usage of Antenna Diversity to Reduce Fading. http://en.wikipedia.org/wiki/Antenna_diversity.

[19] Vertex-Coloring problem. http://en.wikipedia.org/wiki/Graph_coloring.

[20] Straight Talk has upped the ante on throttling users! http://www.howardforums.com/showthread.php/1773165-Straight-Talk-has-upped-the-ante-on-throttling-users!, 2012.

[21] Is there a catch to MVNOs? http://ask.metafilter.com/235587/Is-there-a-catch-to-MVNOs, 2013.

[22] AT&T vs. other MVNO's on latency, capped download speeds. http://www.howardforums.com/showthread.php/1813979-AT-amp-T-vs-other-mvno-s-on-latency-capped-download-speeds, 2013.

[23] M2M and MVNOs driving US connections growth. https://gsmaintelligence.com/analysis/2013/08/m2m-and-mvnos-driving-us-connections-growth/397/, 2013.

[24] Roaming, Regional, Mobile to Mobile, and Other Questions. https://help.ting.com/entries/21227328-roaming-regional-mobile-to-mobile-and-other-questions, 2014.

[25] Sprint, T-Mobile execs explain the MVNO explosion. http://www.fiercewireless.com/special-reports/mvno-explosion-will-latest-wave-virtual-operators-survive-0, 2014.

[26] Western Europe-Evolution of MVNO Business Models-Market Growth, Trends and Opportunities. http://www.releasewire.com/press-releases/western-europe-evolution-of-mvno-business-models-market-growth-trends-and-opportunities-493894.htm, 2014.

[27] Ehsan Aryafar, Omer Gurewitz, and Edward W Knightly. Distance-1 constrained channel assignment in single radio wireless mesh networks. In *Proceedings of The IEEE 27th Conference on Computer Communications*, Infocom '08. IEEE, 2008.

[28] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 216–230. ACM, 2004.

[29] Sergey Baranov. Android ClockSync Application. https://play.google.com/store/apps/details?id=ru.org.amip.ClockSync&hl=en.

[30] Vitali Bashko, Nikolay Melnikov, Amit Sehgal, and Jurgen Schonwalder. Bonafide: A traffic shaping detection tool for mobile networks. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 328–335. IEEE, 2013.

[31] Yigal Bejerano, Seung-Jae Han, and Li (Erran) Li. Fairness and Load Balancing in Wireless LANs Using Association Control. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 315–329. ACM, 2004.

[32] Mike Belshe. More Bandwidth Doesn't Matter (much). http://bit.ly/1rNCOlC, 2010.

[33] G. Bianchi. Performance analysis of the IEEE 802.11 Distributed Coordination Function. In *IEEE Journal on Selected Areas in Communication*, volume 18, pages 535–547, 2000.

[34] Frederico Cali, Marco Conti, and Enrico Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking*, 8:785–799, 2000.

[35] Calì, Frederico and Conti, Marco and Gregori, Enrico. Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit.

In *IEEE/ACM Transaction on Networking*, volume 8, pages 785–799. IEEE Press, December 2000.

[36] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. A Case for Adapting Channel Width in Wireless Networks. In *Proceedings of the 2008 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '08, pages 135–146. ACM, 2008.

[37] Krishna Chintalapudi, Bozidar Radunovic, Vlad Balan, Michael Buettener, Srinivas Yerramalli, Vishnu Navda, and Ramachandran Ramjee. WiFi-NC: WiFi over Narrow Channels. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 4–4. USENIX Association, 2012.

[38] Soon-Hyeok Choi and Scott M Nettles. An overhead controlled MAC protocol for high data-rate wireless networks. In *IEEE 62nd Vehicular Technology Conference (VTC), 2005.*, volume 1, pages 675–679. IEEE, 2005.

[39] Leslie Choong. Multi-Channel IEEE 802.15.4 Packet Capture Using Software Defined Radio. Master's thesis, UCLA, 2009.

[40] Romit Roy Choudhury and Nitin H Vaidya. Deafness: A MAC problem in ad hoc networks when using directional antennas. In *Proceedings of*

the 12th IEEE International Conference on Network Protocols. ICNP 2004. , pages 283–292. IEEE, 2004.

[41] Jeffrey Erman, Vijay Gopalakrishnan, Rittwik Jana, and K. K. Ramakrishnan. Towards a SPDY'Ier Mobile Web? In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 303–314. ACM, 2013.

[42] Hossein Falaki, Dimitrios Lymberopoulos, Ratul Mahajan, Srikanth Kandula, and Deborah Estrin. A First Look at Traffic on Smartphones. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 281–287. ACM, 2010.

[43] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 179–194. ACM, 2010.

[44] Yaser Pourmohammadi Fallah and Hussein M. Alnuweiri. Modeling and Performance Evaluation of Frame Bursting in Wireless LANs. In *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, pages 869–874. ACM, 2006.

[45] Seyed K Fayaz, Fatima Zarinni, and Samir Das. Ez-Channel: A distributed MAC protocol for efficient channelization in wireless networks. *Ad Hoc Networks*, 31:34–44, 2015.

[46] Xiaojun Feng, Jin Zhang, Qian Zhang, and Bo Li. Use your frequency wisely: Explore frequency domain for channel contention and ACK. In *Proceedings of 31st Annual IEEE International Conference on Computer Communications. IEEE INFOCOM*, pages 549–557. IEEE, 2012.

[47] K. Fitchard. Why are MVNOs so hot right now? Thank the carriers. http://gigaom.com/2012/06/25/why-are-mvnos-so-hot-right-now-thank-the-carriers/, 2012.

[48] Aaron Gember, Aditya Akella, Jeffrey Pang, Alexander Varshavsky, and Ramon Caceres. Obtaining In-context Measurements of Cellular Network Performance. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 287–300. ACM, 2012.

[49] Andreas Goldsmith. *Wireless Communications.* Cambridge University Press., 2005.

[50] Ilya Grigorik. *High Performance Browser Networking.* Chapter 7, O'Riley, 2013.

[51] Yan Grunenberger, Martin Heusse, Franck Rousseau, and Andrzej Duda. Experience with an implementation of the idle sense wireless access method. In *Proceedings of the International Conference on emerging*

Networking EXperiments and Technologies, CoNEXT '07, pages 24:1–24:12, New York, NY, USA, 2007. ACM.

[52] Zygmunt J Haas and Jing Deng. Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks. In *IEEE Transactions on Communications*, volume 50, pages 975–985. IEEE, 2002.

[53] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MOBICOM '08, pages 339–350, 2008.

[54] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 802.11 with multiple antennas for dummies. *ACM SIGCOMM Computer Communication Review*, 40(1):19–25, 2010.

[55] Rong He and Xuming Fang. A fair mac algorithm with dynamic priority for 802.11 e wlans. In *International Conference on Communication Software and Networks, 2009. ICCSN'09.*, pages 255–259. IEEE, 2009.

[56] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. In *Proceedings of the 2005 Conference of the the ACM Special Interest Group on Data Communication*, SIGCOMM '05, pages 121–132. ACM, 2005.

[57] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z. Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing Application Performance Differences on Smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 165–178. ACM, 2010.

[58] Xiao Long Huang and Brahim Bensaou. On Max-min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '01, pages 221–231. ACM, 2001.

[59] Nitin Jain, Samir R Das, and Asis Nasipuri. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the Tenth International Conference on Computer Communications and Networks (ICCCN)*, pages 432–439. IEEE, 2001.

[60] Raj Jain, Dah-Ming Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. Technical report, Eastern Lab, Washington University, 1984.

[61] Andrzej Kochut, Arunchandar Vasan, A Udaya Shankar, and Ashok Agrawala. Sniffing out the correct physical layer capture model in 802.11 b. In *Proceedings of the 12th IEEE International Conference on Network Protocols. ICNP 2004.*, pages 252–261. IEEE, 2004.

[62] Jeongkeun Lee, Wonho Kim, Sung-Ju Lee, Daehyung Jo, Jiho Ryu, Taekyoung Kwon, and Yanghee Choi. An Experimental Study on the Capture Effect in 802.11a Networks. In *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WinTECH '07, pages 19–26. ACM, 2007.

[63] Youngseok Lee. Measured TCP Performance in CDMA 1x EV-DO Network. In *Proceedings of Passive and Active Measurement Conference*, PAM '06, 2006.

[64] Paul Lettieri and Mani B Srivastava. Adaptive frame length control for improving wireless link throughput, range, and energy efficiency. In *Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 564–571. IEEE, 1998.

[65] Tianji Li, Qiang Ni, David Malone, Douglas Leith, Yang Xiao, and Thierry Turletti. Aggregation with fragment retransmission for very high-speed WLANs. In *IEEE/ACM Transactions on Networking*, volume 17, pages 591–604. IEEE Press, 2009.

[66] Li, Tianji and Ni, Qiang and Malone, David and Leith, Douglas and Xiao, Yang and Turletti, Thierry. A new MAC scheme for very high-speed WLANs. In *Proceedings of the 2006 International Symposium on*

on World of Wireless, Mobile and Multimedia Networks, WoWMoM '06, pages 171–180. IEEE Computer Society, 2006.

[67] M. Lindasalwa, M Begam, and I Elamvazuthi. Voice recognition algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) techniques. In *arXiv preprint arXiv:1003.4083*, 2010.

[68] Eugenio Magistretti, Krishna Kant Chintalapudi, Bozidar Radunovic, and Ramachandran Ramjee. WiFi-Nano: Reclaiming WiFi Efficiency Through 800 Ns Slots. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 37–48. ACM, 2011.

[69] Ritesh Maheshwari, Jing Cao, Anand Prabhu Subramanian, Fatima Zarinni, and Samir Das. Adaptive channelization for high data rate wireless networks. Technical report, WINGS Lab, Stony Brook University, 2009.

[70] Ritesh Maheshwari, Himanshu Gupta, and Samir R Das. Multichannel MAC protocols for wireless networks. In *Proceedings of 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks. SECON'06.*, volume 2, pages 393–401. IEEE, 2006.

[71] Gregor Maier, Fabian Schneider, and Anja Feldmann. A First Look at Mobile Hand-held Device Traffic. In *Proceedings of the 11th Interna-*

tional Conference on Passive and Active Measurement, PAM '10, pages 161–170. Springer-Verlag, 2010.

[72] Suparerk Manitpornsut, Björn Landfeldt, and Azzedine Boukerche. Efficient Channel Assignment Algorithms for Infrastructure WLANs Under Dense Deployment. In *Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '09, pages 329–337. ACM, 2009.

[73] Justin Manweiler, Sharad Agarwal, Ming Zhang, Romit Roy Choudhury, and Paramvir Bahl. Switchboard: A Matchmaking System for Multiplayer Mobile Games. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 71–84. ACM, 2011.

[74] Anni Matinlauri. Fairness and Transmission Opportunity Limit in IEEE 802.11e Enhanced Distributed Channel Access. Master's thesis, Helsinki University of Technology, 2008.

[75] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for WLANs. In *ACM SIGMOBILE Mobile Computing and Communications Review*, volume 9, pages 19–31. ACM, 2005.

[76] Arunesh Mishra1, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs.

In *Proceedings of 25th IEEE International Conference on Computer Communications*, INFOCOM '06, pages 1–12, 2006.

[77] Thomas Moscibroda, Ranveer Chandra, Yunnan Wu, Sudipta Sengupta, Paramvir Bahl, and Yuan Yuan. Load-aware spectrum distribution in wireless LANs. In *IEEE International Conference on Network Protocols*, ICNP 2008, pages 137–146. IEEE, 2008.

[78] Ashkan Nikravesh, David R Choffnes, Ethan Katz-Bassett, Z Morley Mao, and Matt Welsh. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *Passive and Active Measurement*, PAM '14, pages 12–22, 2014.

[79] I. Papanikos and M. Logothetis. A Study on Dynamic Load Balance for IEEE 802.11b Wireless LAN. In *8th International Conference on Advances in Communication and Control*, 2001.

[80] Cheolhee Park and Theodore S Rappaport. Short-Range Wireless Communications for Next-Generation Networks: UWB, 60 GHz Millimeter-Wave WPAN, And ZigBee. In *IEEE Wireless Communications* , volume 14, pages 70–78. IEEE, August 2007.

[81] Minyoung Park, C. Cordeiro, E. Perahia, and L.L. Yang. Millimeter-wave multi-Gigabit WLAN: Challenges and feasibility. In *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2008.* , pages 1–5, Sept. 2008.

[82] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R Das. Understanding traffic dynamics in cellular data networks. In *Proceedings of IEEE Conference on Computer Communications*, INFOCOM '11, pages 882–890. IEEE, 2011.

[83] Chunyi Peng, Chi-yu Li, Guan-Hua Tu, Songwu Lu, and Lixia Zhang. Mobile data charging: new attacks and countermeasures. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 195–204. ACM, 2012.

[84] Hariharan Rahul, Farinaz Edalat, Dina Katabi, and Charles G. Sodini. Frequency-aware Rate Adaptation and MAC Protocols. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 193–204. ACM, 2009.

[85] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3 of *INFOCOM '05*, pages 2223–2234. IEEE, 2005.

[86] Shravan Rayanchu, Vivek Shrivastava, Suman Banerjee, and Ranveer Chandra. FLUID: Improving Throughputs in Enterprise Wireless Lans Through Flexible Channelization. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 1–12. ACM, 2011.

[87] Bogdan Roman, Frank Stajano, Ian Wassell, and David Cottingham. Multi-carrier burst contention (MCBC): Scalable medium access control for wireless networks. In *In Proceedings of Wireless Communications and Networking Conference, 2008. WCNC 2008.*, pages 1667–1672. IEEE, 2008.

[88] Sanae Rosen, Haokun Luo, Qi Alfred Chen, Z. Morley Mao, Jie Hui, Aaron Drake, and Kevin Lau. Discovering Fine-grained RRC State Dynamics and Performance Impacts in Cellular Networks. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 177–188. ACM, 2014.

[89] John P. Rula and Fabian E. Bustamante. Behind the Curtain: The Importance of Replica Selection in Next Generation Cellular Networks. In *Proceedings of the 2014 ACM Conference of the Special Interest Group on Data Communication*, SIGCOMM '14, pages 135–136. ACM, 2014.

[90] Stuart Russell and Peter Norvig. *Artificial Intelligence, A Modern Approach.* Prentice Hall, 2003.

[91] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. CSMA/CN: Carrier Sense Multiple Access with Collision Notification. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 25–36. ACM, 2010.

[92] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. No Time to Countdown: Migrating Backoff to the Frequency Domain. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 241–252. ACM, 2011.

[93] Jingpu Shi, Theodoros Salonidis, and Edward W. Knightly. Starvation Mitigation Through Multi-channel Coordination in CSMA Multi-hop Wireless Networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, pages 214–225. ACM, 2006.

[94] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. CENTAUR: Realizing the Full Potential of Centralized Wlans Through a Hybrid Data Path. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 297–308. ACM, 2009.

[95] Vivek Shrivastava, Shravan Rayanchu, Jongwoon Yoonj, and Suman Banerjee. 802.11n Under the Microscope. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pages 105–110. ACM, 2008.

[96] Ashiwan Sivakumar, Vijay Gopalakrishnan, Seungjoon Lee, Sanjay Rao, Subhabrata Sen, and Oliver Spatscheck. Cloud is Not a Silver Bullet: A

Case Study of Cloud-based Mobile Browsing. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, HotMobile '14, pages 21:1–21:6. ACM, 2014.

[97] Peter Smulders. Exploiting the 60 GHz band for local wireless multimedia access: prospects and future directions. *IEEE Communications Magazine*, 40(1):140–147, 2002.

[98] Jungmin So and Nitin H. Vaidya. Multi-channel Mac for Ad Hoc Networks: Handling Multi-channel Hidden Terminals Using a Single Transceiver. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '04, pages 222–233. ACM, 2004.

[99] S. Srikanth, V. Kumaran, and C. Manikandan. Orthogonal Frequency Division Multiple Access: is it the multiple access system of the future? Technical report, AU-KBC Research Center, Anna University, India.

[100] Anand Prabhu Subramanian, Mahmoud Al-Ayyoub, Himanshu Gupta, Samir R Das, and Milind M Buddhikot. Near-optimal dynamic spectrum allocation in cellular networks. In *Proceedings of3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN 2008.*, pages 1–11. IEEE, 2008.

[101] Anand Prabhu Subramanian and Himanshu Gupta. Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks.

In *Proceedings of 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN 2007* , pages 320–330. IEEE, 2007.

[102] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-grained Channel Access in Wireless LAN. In *Proceedings of the 2010 ACM Conference of Special Interest Group on Data Communication*, SIGCOMM '10, pages 147–158. ACM, 2010.

[103] Ilenia Tinnirello and Sunghyun Choi. Efficiency analysis of burst transmissions with block ACK in contention-based 802.11 e WLANs. In *Proceedings of 2005 IEEE International Conference on Communications*, volume 5 of *ICC 2005*, pages 3455–3460. IEEE, 2005.

[104] Hector Velayos, Victor Aleo, and Gunnar Karlsson. Load balancing in overlapping wireless lan cells. In *2004 IEEE International Conference on Communications*, volume 7, pages 3833–3836. IEEE, 2004.

[105] Bo Wang, William Wu, and Yongqiang Liu. Dynamic Channel Assignment in Wireless LANs.

[106] Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Zhuoqing Mao, and Ming Zhang. An untold story of middleboxes in cellular networks. In *Proceedings of the 2011 ACM Conference of Special Interest Group on Data Communication*, SIGCOMM '11, pages 374–385. ACM, 2011.

[107] Matt Welsh. Why Mobile Performance is Hard. ACM Mobicom Indutsrial Paper Session . http://www.sigmobile.org/mobicom/2013/MobiCom2013_IndustrialTalks_MattWelsh.pdf, 2013.

[108] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks. I-SPAN 2000.*, pages 232–237. IEEE, 2000.

[109] Yang Xiao. IEEE 802.11 n: enhancements for higher throughput in wireless LANs. *IEEE Wireless Communications*, 12(6):82–91, 2005.

[110] Yang Xiao. Efficient MAC strategies for the IEEE 802.11 n wireless LANs. *Wireless Communications and Mobile Computing*, 6(4):453–466, 2006.

[111] Yang Xiao and Jon Rosdahl. Performance Analysis and Enhancement for the Current and Future IEEE 802.11 MAC Protocols. In *SIGMOBILE Mobile Computing and Communications Review*, volume 7, pages 6–19. ACM, 2003.

[112] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM Con-*

ference on Internet Measurement Conference, IMC '11, pages 329–344. ACM, 2011.

[113] L. Lily Yang. 60GHz: Opportunity for Gigabit WPAN and WLAN Convergence. *SIGCOMM Computing and Communications Review*, 39(1):56–61, December 2008.

[114] L. Lily Yang and Minyoung Park. Applications and challenges of multiband gigabit mesh networks. In *Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications*, SENSORCOMM '08, pages 813–818. IEEE Computer Society, 2008.

[115] Lei Yang, Lili Cao, Heather Zheng, and Elizabeth Belding. Traffic-aware Dynamic Spectrum Access. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, WICON '08, pages 10:1–10:9, 2008.

[116] Lei Yang, Wei Hou, Lili Cao, Ben Y. Zhao, and Haitao Zheng. Supporting Demanding Wireless Applications with Frequency-agile Radios. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 5–5. USENIX Association, 2010.

[117] Xue Yang and Nitin H Vaidya. A wireless MAC protocol using implicit pipelining. *IEEE Transactions on Mobile Computing*, 5(3):258–273, 2006.

[118] Xue Yang, Nitin H Vaidya, and Priya Ravichandran. Split-Channel Pipelined Packet Scheduling for Wireless Networks. *IEEE Transactions on Mobile Computing (TMC)*, 5(3):240–257, 2006.

[119] Hujun Yin and Siavash Alamouti. OFDMA: A broadband wireless access technology. In *Proceedings of the IEEE Sarnoff Symposium*, pages 1–4. IEEE, 2006.

[120] Jun Yin, Xiaodong Wang, and Dharma P Agrawal. Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function. In *Proceedings of Wireless Communications and Networking Conference. WCNC 2004*, volume 3, pages 1654–1659. IEEE, 2004.

[121] Yuan Yuan, Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, and Yunnan Wu. Allocating Dynamic Time-spectrum Blocks in Cognitive Radio Networks. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 130–139. ACM, 2007.

[122] Yuan Yuan, Daqing Gu, William Arbaugh, and Jinyun Zhang. High-Performance MAC for High-Capacity Wireless LANs. In *Proceedings of 13th International Conference on Computer Communications and Networks. ICCCN 2004.*, pages 167–172. IEEE, 2004.

[123] Kyriakos Zarifis, Tobias Flach, Srikanth Nori, David Choffnes, Ramesh Govindan, Ethan Katz-Bassett, Z Morley Mao, and Matt Welsh. Di-

agnosing path inflation of mobile client traffic. In *Passive and Active Measurement*, PAM '14, pages 23–33, 2014.

[124] Fatima Zarinni, Ayon Chakraborty, Vyas Sekar, Samir R Das, and Phillipa Gill. A First Look at Performance in Mobile Virtual Network Operators. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC 2014, pages 165–172. ACM, 2014.

[125] Fatima Zarinni and Samir R Das. Understanding the FICA MAC Protocol in High Data Rate WLANs. In *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2012.